The theory and application of bootstrap control charts for statistical process control

by

Evert Johan Coetzee (27179631)

Submitted in partial fulfillment of the requirements for the degree

Magister Scientiae

In the Department of Statistics

In the Faculty of Natural and Agricultural Sciences

University of Pretoria

May 2017

Declaration

I, *Evert Johan Coetzee,* declare that this mini-dissertation, which I hereby submit for the degree Magister Scientiae in Mathematical Statistics at the University of Pretoria, is my own work and has not previously been submitted by me for a degree at this or any other tertiary institution.

Signature:_____

Date: 29 May 2017

Member, South African Translators' Institute

28 18th Street
Menlo Park
Pretoria 0081
06 February 2017

TO WHOM IT MAY CONCERN

This is to confirm that the thesis titled "The theory and application of bootstrap control charts for statistical process control" by Evert Johan Coetzee, submitted in partial fulfillment of the requirements for the degree Magister Scientiae, in the Department of Statistics, Faculty of Natural and Agricultural Sciences,University of Pretoria, was proof read and edited by me in respect of Language Content.

I verify that it is ready for publication or public viewing in respect of language and style.

Please note that no view is expressed in respect of the specific technical contents of the document or changes made after the date of this letter.

Kind regards

Anna M de Wet

BA Hons (Cum Laude), University of Pretoria

Summary

Chapter 1 of this mini-dissertation gives an introduction to Statistical Process Control (SPC) and provides some background on the Shewhart, CUSUM and the EWMA control charts. The bootstrap by [9] Efron (1979) is discussed and a brief overview of Phase I and Phase II analysis is given. The chapter concludes with the research objectives of this dissertation.

Chapter 2 of this dissertation provides a literature review of bootstrap Shewhart, cumulative sum (CUSUM), exponentially weighted moving average (EWMA) and multivariate control charts. The Shewhart-type control charts mostly focus on the bootstrap procedures proposed by [2] Bajgier (1992), [34] Seppala, Moskowitz, Plante and Tang (1995) and [23] Liu and Tang (1996). An overview of the bootstrap CUSUM charts proposed by [7] Chatterjee and Qiu (2009) and [1] Ambartsoumian and Jeske (2015) is given. A review of the parametric bootstrap control chart used by [33] Saleh, Mahmoud, Jones-Farmer, Zwetsloot and Woodal (2015) to construct EWMA control charts is given. The chapter concludes with a review of the bootstrap T^2 control chart proposed by [32] Phaladiganon, Kim, Chen, Baek and Park (2011).

In **Chapter 3** the design of a potential nonparametric bootstrap EWMA control is given. The chapter concludes with two examples of how the control limits for such a chart can be constructed for two different statistics.

Chapter 4 of this mini-dissertation examines conditional in-control (IC) and out-of-control (OOC) average run-length, for the chart proposed in Chapter 3, taking different underlying process distributions into consideration.

In **Chapter 5** the the mini-dissertation is concluded by summarising the research that has been done and providing recommendations for further research.

Acknowledgements

In Memory of Johan Uys (1942-2016)

I would like to thank my parents Johan and Sonet who never lost faith in me and encouraged me to pursue my dreams regardless of what obstacles might lie in my path.
I would like thank my supervisor Dr M.A. Graham for her guidance, support, insight and motivation - without her support and belief in me, I would not have made it.
I would like to gratefully acknowledge the financial assistance I received from the National Research Foundation (NRF) for a SASA-NRF Statistics in Crisis grant number 96628.
Special thanks to thank Dr F.H.J. Kanfer for his insightful comments and valuable discussions.
I would like to thank Professor A. Bekker and Dr I. Fabris-Rotelli, from the bottom of my heart, for all their support and motivation.
A special thank you to Mrs Corbett for her support.

Finally, I would like to thank my girlfriend, Roelien du Toit, for her love, support and the sacrifices she made.

Contents

Chapter 1

Introduction

1.1 What is a control chart?

A control chart is a two-dimensional graphical representation of a plotting (charting) statistic against time or subgroup number. The charting statistic and the control limits are calculated from data collected sequentially over time. The data can either be individual observations or subgroups (samples) of observations. The statistical control chart concept was developed by [36] Walter A. Shewhart in 1924 when he was working at Bell Laboratories. An example of a typical two-sided Shewhart-type control chart is shown in Figure 1.1.

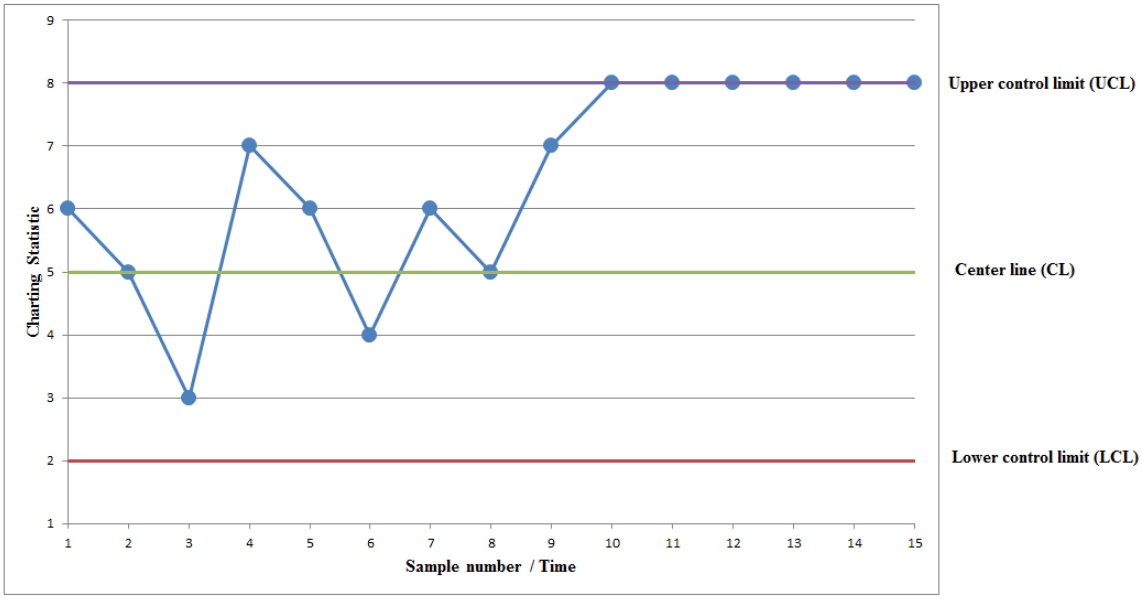


Figure 1.1: A two-sided Shewhart-type control chart

From Figure 1.1 it can be seen that a control chart consists of three horizontal lines: a lower control limit (LCL) at the bottom, a center line (CL) and an upper control limit (UCL) at the top. These lines are placed on the control chart to aid the user in making informed and objective decisions, with respect to whether a process is in-control (IC) or out-of-control (OOC). When a charting statistic plots *on* or *outside* one of the control limits the process is declared to be OOC. This can also be referred to as a signaling event. On the contrary, if the charting statistic plots anywhere between the two control

limits, then the process is declared to be IC; this is referred to as a non-signaling event. From Figure 1.1 it can be seen that samples 1 to 9 are non-signaling events and the process is subsequently IC. However, from sample 10 onwards we can see that the process is OOC, since the sample's charting statistic plotted on the UCL.

1.1.1 Run-length distribution

[14] Human and Graham (2007) stated in their control charting overview paper that the the run-length of a control chart is defined as: "The number of rational subgroups to be collected or the number of charting statistics to be plotted on a control chart before the first OOC signal is observed..." . The run-length is a random variable, usually denoted by N, usually with finite mean and variance. The most widely used statistic to measure the performance of a control chart is the mean run-length, usually referred to as the average run-length (ARL).

It is known that the run-length distribution is skewed to the right and it has been suggested by other researchers such as [12] Gan (1994) and [19] Khoo, Wong, Wu and Castagliola (2011) to use the median run-length (MRL), the standard deviation of the run-length ($SDRL$) and other percentiles to measure the performance of the control chart. In this mini-dissertation the ARL will primarily be used to measure the performance of control charts, since it is still the most well-known and the most widely used performance measure in statistical process control (SPC) literature. The run-length distribution and its corresponding characteristics (such as the average, median, standard deviation, etc.) can be obtained by four methods which are the exact approach, the Markov chain approach, computer (or Monte Carlo) simulations and the exact / integral approach. In this dissertation the computer simulation approach will be used and more detail on it will be given later.

1.2 Types of control charts

There are three main classes of control charts: the Shewhart chart, the cumulative sum (CUSUM) chart and the exponentially weighted moving average (EWMA) chart and other variations and/or refinements based on these charts. Relative advantages and disadvantages of these charts are documented in the literature (see e.g. [27] Montgomery (2009)). These charts are described in more detail in each of the three sections that follow.

1.2.1 Shewhart

For the discussion on the Shewhart control chart that follows in this section, assume that $X_{i1}, X_{i2}, \ldots, X_{in}$ denote a random sample of size $n \geq 1$ from the process at time $i = 1, 2, 3, \ldots$. Let T denote some sample statistic that measures the characteristic of interest to the researcher/practitioner, and suppose that the mean of T is μ_T and the standard deviation of T is σ_T. Then the UCL, CL and LCL are given by

$$UCL = \mu_T + k\sigma_T$$

$$CL = \mu_T$$

$$LCL = \mu_T - k\sigma_T$$

respectively, where $k \geq 0$ is the 'distance' of the control limits from the CL, expressed in standard deviation units. If the charting statistic plots on or outside either of the control limits, the process is declared to be OOC.

1.2.2 CUSUM

Assume that $X_1, X_2, X_3, \ldots$ denote independent and identically distributed (i.i.d.) measurements from a process with mean, μ_0, and standard deviation, σ_0, where the subscript zero indicates that the mean and standard deviation are known. The CUSUM chart is formed by plotting C_i where

$$C_i = \sum_{k=1}^{i}(X_k - \mu_0) = (X_i - \mu_0) + \sum_{k=1}^{i-1}(X_k - \mu_0) = (X_k - \mu_0) + C_{i-1}. \tag{1.1}$$

The upper one-sided CUSUM works by accumulating deviations from $(\mu_0 + k)$ that are above target. For the upper one-sided CUSUM chart we use C_i^+, with

$$C_i^+ = max[0, X_i - (\mu_0 + k) + C_{i-1}] \text{ for } i = 1, 2, 3, \ldots \tag{1.2}$$

where $k > 0$ is referred to as the reference or slack value and the starting value is $C_0^+ = 0$. A signaling event occurs for the first i such that $C_i^+ \geq H$, where $H > 0$ is referred to as the decision interval, which can be seen as an upper control limit.

The lower one-sided CUSUM considers deviations from $(\mu_0 - k)$. For the lower one-sided CUSUM chart we use C_i^-, with

$$C_i^- = max[0, (\mu_0 - k) - X_i + C_{i-1}^-] \tag{1.3}$$

or

$$C_i^{-^*} = min[0, X_i - \mu_0 + k + C_{i-1}^-] \text{ for } i = 1, 2, 3, \ldots, \tag{1.4}$$

where the starting values are $C_0^+ = C_0^- = 0$ and $k \geq 0$.

Here a signaling event occurs for the first i such that $C_i^- \leq -H$ or $C_i^{-^*} \geq H$. The control chart design parameters, k and H, are selected such that a desired nominal ARL_0 is attained.

1.2.3 EWMA

Assume that $X_1, X_2, X_3, \ldots$ denote i.i.d. observations from a process with a known process mean, μ_0, and a known process standard deviation, σ_0. The exponentially weighted moving average (EWMA) charting statistic is defined as

$$Z_i = \lambda X_i + (1-\lambda)Z_{i-1} \text{ for } i = 1, 2, 3, \ldots, \tag{1.5}$$

where $0 < \lambda \leq 1$ is a smoothing constant and the starting value, Z_0, is the process target value, i.e. $Z_0 = \mu_0$. The expected value and the variance of Z_i are given by

$$E[Z_i] = \mu_0 \tag{1.6}$$

and

$$Var[Z_i] = \sigma_0^2 \left(\frac{\lambda}{2-\lambda}\right)(1-(1-\lambda)^{2i}), \tag{1.7}$$

respectively. The exact control limits and the center line of the EWMA control chart are given by

$$UCL_i = \mu_0 + L\sigma_0\sqrt{\tfrac{\lambda}{2-\lambda}(1-(1-\lambda)^{2i})}$$

$$CL_i = \mu_0 \tag{1.8}$$

$$LCL_i = \mu_0 - L\sigma_0\sqrt{\tfrac{\lambda}{2-\lambda}(1-(1-\lambda)^{2i})},$$

where $L > 0$ is the width of the control limits, that is chosen together with λ to obtain a desired nominal ARL_0.

The steady-state control limits are implemented when the chart has been running for some period of time. As $i \to \infty$, then $Var[Z_i] = \sigma_0^2 \frac{\lambda}{2-\lambda}$, since $(1-(1-\lambda)^{2i}) \to 1$ as $i \to \infty$. The steady-state UCL and LCL are given by

$$UCL = \mu_0 + L\sigma_0\sqrt{\tfrac{\lambda}{2-\lambda}}$$

and (1.9)

$$LCL = \mu_0 - L\sigma_0\sqrt{\tfrac{\lambda}{2-\lambda}},$$

respectively. If the charting statistic Z_i plots between the two control limits the process is IC, otherwise it is considered to be OOC. [27] Montgomery (2009) found that values of λ in the interval $[0.05, 0.25]$ work well in practice, with $\lambda = 0.05$, $\lambda = 0.10$ and $\lambda = 0.20$ being popular choices. Recommendations for λ by [27] Montgomery (2009) are summarised in Table 1.1.

Table 1.1: Choice of smoothing parameter λ

Magnitude of the shift of interest	Choice of λ
Small	0.05
Moderate	0.10
Large	0.20

[27] Montgomery (2009) found that $L = 3$ works reasonably well, particularly for larger values λ, although Montgomery (2009) advised using an L between about 2.6 and 2.8 when $\lambda \leq 0.1$.

1.2.4 Overview of the Monte Carlo simulation approach

The main advantage of using this method is that the run-length distribution characteristics can easily be calculated, regardless of how complicated the run-length distribution is. [28] Mundform, Schaffer, Kim, Shaw, Thongteeraparp, Preecha and Supawan (2011) showed that 7 500 to 8 000 simulations are sufficient for most SPC applications. In this mini-dissertation 10 000 simulations were used, since it should be a sufficient number of simulations to keep the error of a run-length characteristic within reasonable bounds. [6] Chakraborti and Van de Wiel (2008) stated the 10% error band (i.e. run-length characteristic + 0.1(run-length characteristic)) might be too wide to detect practical departures of the simulated results from the target value, they used a 2% error band. SAS v9.4 was used to perform simulations in this mini-dissertation. A typical computer simulation procedure to calculate the run-length distribution for a two-sided control chart, where the charting statistic is calculated from a random sample, is given as follows:

1. After specifying the necessary parameters, such as the subgroup size, the control limits are calculated (i.e. LCL and UCL).
2. Random subgroups are generated from some process distribution, say, the normal or exponential distribution.
3. The charting statistic for each subgroup is calculated and compared to the control limits calculated in step 1.
4. The number of samples (or subgroups) needed until the charting statistic plots on or outside the control limits is recorded and is then used as an observation for the run-length distribution.
5. Repeat steps 1 to 4 a total of 10 000 times.
6. Once a dataset with 10 000 observations from the run-length distribution has been obtained, summary statistics like the ARL, $SDRL$ and percentiles of the run-length distribution can be obtained.

1.3 Phase I and Phase II analysis of process data

[5] Chakraborti, Human and Graham (2009) stated that SPC is "implemented in two phases or stages in practice: Phase I, the so-called retrospective phase, and Phase II, the prospective or monitoring

phase."

Typically, the planning, administration, design of the study, data collection, data management and exploratory work (graphical and numerical analysis, goodness-of-fit and so on) occurs during Phase I ([5] Chakraborti et al. (2009), p53). It was stated by [5] Chakraborti et al. (2009) that the goal of Phase I is to make sure the process is operating at or near acceptable target(s) under natural (or common) causes of variation. The data obtained during the Phase I analysis is used to construct control limits, that could potentially be used to monitor an attribute or parameter of the process. [5] Chakraborti et al. (2009) showed that the control limits obtained during this phase are merely trial limits, and are often refined and revised to ensure the process is IC. Phase I is iterative by nature, since data points that plot on or outside the preliminary control limits are discarded and the limits are subsequently recalculated using the remaining data.

When no assignable causes for variation are found, the final set of control limits is constructed using the supposed IC data. These control limits are then used to monitor the process. Very often, the process parameters are unknown and need to be estimated using the Phase I data, this is known as the s*tandard(s) unknown case* (denoted *Case U*).

1.4 The bootstrap

The bootstrap is a statistical technique, proposed by [9] Efron (1979), which deals with the problem of being given a random sample, say $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$, from an unknown probability distribution with cumulative density function (CDF) $F(x)$. One then proceeds to use the random sample, $\boldsymbol{X}$, to estimate the *sampling* distribution of a statistic, say $T(\boldsymbol{X}) = T(X_1, X_2, \ldots, X_n)$. Efron (1979) argued that the unknown CDF $F(x)$ can be approximated by the empircal density function (EDF) $\hat{F}(x)$, that is

$$\hat{F}(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(X_i \leq x), \tag{1.10}$$

where $\mathbf{1}(A) = 1$, when A is true and $\mathbf{1}(A) = 0$, when A is false. [9] Efron (1979) suggested drawing random samples, each of size n, from the population with distribution function equal to the EDF. This random sample is denoted by $X_1^*, X_2^*, \ldots, X_n^*$ and is referred to as a bootstrap sample or replication. It can be shown that drawing random samples from the EDF, is logically equivalent to sampling with replacement from the original sample, $\boldsymbol{X}$. [9] Efron (1979) suggested calculating T, based on the bootstrap sample $X_1^*, X_2^*, \ldots, X_n^*$; this is denoted by T^*. The resampling procedure is typically repeated a large number of times, say $B \geq 1\ 000$, and the statistic T is calculated for each of these bootstrap samples, i.e. T is calculated for every re-sample so that we have $T_1^*, T_2^*, \ldots, T_B^*$.

That bootstrap statistics $T_1^*, T_2^*, \ldots, T_B^*$ can be used to emulate (or approximate) the sampling distribution for T. [9] Efron (1979) showed that $T_1^*, T_2^*, \ldots, T_B^*$ can be used to construct reasonably accurate confidence intervals, provided that T is not an extreme statistic (like the maximum value in the sample, for example).

1.5 Research objectives

The primary focus of this mini-dissertation is to investigate how the bootstrap can be used to construct univariate control charts, more specifically, to detect changes in the location of a process distribution (i.e. an upward/downward shift in the process mean). In **Chapter 2** what has been done with regard to the Shewhart, CUSUM and EWMA control charts are reviewed. In **Chapter 3** a naive approach for a bootstrap EWMA control chart is proposed. In **Chapter 4** the performance of the proposed EWMA chart is reviewed. In **Chapter 5** we discuss potential problems with the chart and make recommendations for further study.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, an overview of the literature is given starting with the well-known Shewhart control chart (Section 2.2), followed by the CUSUM, EWMA and some other control charts in Sections 2.2, 2.3 and 2.4, respectively.

In Section 2.2 the bootstrap approaches proposed by [2] Bajgier (1992), [34] Seppala, Moskowitz, Plante and Tang (1995) and [23] Liu and Tang (1996) are discussed and compared to the findings of [18] Jones and Woodall (1998) (see Table 2.1, 2.2, 2.3 and 2.4, respectively and Figure 2.1, 2.2, 2.3, 2.4, 2.5 , 2.6 and 2.7, respectively).

In Section 2.3 the bootstrap variants of the CUSUM control charts proposed by [39] Yaschin (1992), [7] Chatterjee and Qiu (2009) and [1] Ambartsoumian and Jeske (2015) are discussed. A summary of the performance of the bootstrap CUSUM chart proposed by [7] Chatterjee and Qiu (2009) is given in Section 2.3.2. The performance of the bootstrap CUSUM chart proposed by [1] Ambartsoumian and Jeske (2015) (more specifically the probability integral transformation CUSUM) is summarised in Table 2.5 and compared to the transformed CUSUM proposed by [16] Jeske (2009).

In Section 2.4 an overview of the bootstrap EWMA chart used by [13] Gandy and Kvaløy (2013) and [33] Saleh et al. (2015) is given.

Section 2.5 gives an overview of the bootstrap-based multivariate T^2 control chart proposed by [32] Phaladiganon et al. (2011).

2.2 Shewhart control chart

2.2.1 Overview of bootstrap control chart designs

[2] Bajgier (1992) proposed a control chart to monitor the mean ($\bar{X}$) of a process with the intention of providing an alternative for the Shewhart $\bar{X}$ control chart. [18] Jones and Woodall (1998) suggested constructing [2] Bajgier's (1992) bootstrap control chart as follows:

1. Observe m subgroups of size n for a total of $N = mn$ observations.

2. Draw a random sample with replacement from the mn pooled observations. This sample, $x_1^*, x_2^*, \ldots, x_n^*$, is a bootstrap sample.

3. Compute the mean ($\bar{x}^*$) from the bootstrap sample drawn in step 2.

4. Repeat steps 2 to 3 a large number of times, say, B times.

5. Sort the B bootstrap estimates, $\bar{x}_1^*, \bar{x}_2^*, \ldots, \bar{x}_B^*$, in ascending order.

6. Find the smallest ordered $\bar{x}^*$ such that $(\alpha/2)B$ values are below it. This is the bootstrap control chart's lower control limit, LCL.

7. Find the smallest ordered $\bar{x}^*$ such that $(1-\alpha/2)B$ values are below it. This is the bootstrap control chart's upper control limit, UCL.

Here, α is the desired false alarm rate (FAR). For a typical 3σ Shewhart control chart with the mean and the variance assumed known, we have that $\alpha = 0.0027$.

Since each set of mn observations will produce different bootstrap and standard control limits, the ARL of each chart is a random variable. [2] Bajgier (1992) simulated 1 000 sets of control limits for $m = 20$ samples, each of size $n = 5$, from a normal distribution with mean 0 and variance 1, and a χ^2 distribution with five degrees of freedom. The author compared the bootstrap control limits to standard Shewhart control limits based on

$$\begin{gathered}(LCL, UCL) = \bar{\bar{X}} \pm A_2 \bar{R} \\ \text{and} \\ (LCL, UCL) = \bar{\bar{X}} \pm A_3 \bar{S}\end{gathered} \tag{2.1}$$

respectively, where

$$\bar{\bar{X}} = \frac{1}{m}\sum_{i=1}^{m} \bar{X}_i = \frac{1}{m}\sum_{i=1}^{m}\left[\frac{1}{n}\sum_{j=1}^{n} X_{ij}\right] = \frac{1}{mn}\sum_{j=1}^{m}\sum_{i=1}^{n} X_{ij},$$

$$\bar{R} = \frac{1}{m}\sum_{i=1}^{m} R_i,$$

$$R_j = max\{X_{i1}, X_{i2}, \ldots, X_{in}\} - min\{X_{i1}, X_{i2}, \ldots, X_{in}\} \text{ for } i = 1, 2, 3, \ldots, m,$$

$$\bar{S} = \frac{1}{m}\sum_{i=1}^{m} S_j = \sum_{j=1}^{n}\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(X_{ij} - \bar{X}_i\right)^2}$$

and A_2 and A_3 are standard charting constants (see [27] Montgomery (2009), p702). [2] Bajgier (1992) concluded that his bootstrap control chart performs comparably to the standard methods in terms of the distribution of the ARL's.

[34] Seppala et al. (1995) pointed out a flaw with [2] Bajgier's (1992) bootstrap control chart, namely that [2] Bajgier's (1992) approach assumed that the process was stable and in-control when the control limits were computed. [18] Jones et al. (1998) argued that violating this assumption would lead to control limits that were too wide. [34] Seppala et al. (1995) attempted to reduce the necessity of this assumption with their subgroup bootstrap. [34] Seppala et al. (1995) assumed that the observations were described by the model

$$X_{ij} = \mu_i + \epsilon_{ij} \text{ for } i = 1, 2, \ldots, m \text{ and } j = 1, 2, \ldots, n,$$

where μ_i is the process mean for the i-th subgroup and ϵ_{ij} is a random error term with mean 0 and variance σ_ϵ^2. The subgroup bootstrap control chart by [34] Seppala et al. (1995) is constructed as follows:

1. Observe m subgroups of size n for a total of $N = mn$ observations.
2. Compute $\hat{e}_{ij} = x_{ij} - \bar{x}_i$ for $i = 1, 2, 3, \ldots, m$ and $j = 1, 2, 3, \ldots, n$, where $\bar{x}_i$ is the sample mean for the i-th subgroup.
3. Draw a random sample with replacement from the mn pooled sample residuals. This sample, $\hat{e}_1^*, \hat{e}_2^*, \ldots, \hat{e}_n^*$, is a bootstrap sample.
4. Compute $x_i^* = \bar{\bar{x}} + a\hat{e}_i^*$ for $i = 1, 2, 3, \ldots, n$. Here, $a = \sqrt{n/(n-1)}$ is a correction factor used to adjust the variance of the resampled subgroups.
5. Compute the sample mean $(\bar{x}^*)$ from $x_1^*, x_2^*, \ldots, x_n^*$.
6. Repeat steps 3 to 5 a large number of times, say, B times.
7. Sort the B bootstrap estimates, $\bar{x}_1^*, \bar{x}_2^*, \ldots, \bar{x}_B^*$, in ascending order.
8. Find the smallest ordered $\bar{x}^*$ such that $(\alpha/2)B$ values are below it. This is the bootstrap control chart's lower control limit, LCL.
9. Find the smallest ordered $\bar{x}^*$ such that $(1 - \alpha/2)B$ values are below it. This is the bootstrap control chart's upper control limit, UCL.

Here, α is the desired FAR for the control chart. [34] Seppala et al. (1995) suggested replacing the control limits found in steps 8 and 9 with interpolated percentiles. [34] Seppala et al. (1995) also suggested a modification to the subgroup bootstrap algorithm referred to as a balanced bootstrap. [11] Efron and Tibshirani (1994) described the balanced bootstrap as a process where the data is resampled in such a way so that each bootstrap sample appears exactly once in the B resamples. This is done by concatenating B copies of $x_1.x_2, \ldots, x_n$ into a string L of the length nB, then taking a random permutation of L, say $\tilde{L}$. Finally, the first bootstrap sample should be elements $1, 2, \ldots, n$ of $\tilde{L}$, the second bootstrap sample to be elements $n+1, n+2, \ldots, 2n$ of $\tilde{L}$, and so on. [34] Seppala et al.

(1995) used a measure called 'simulated coverage probabilities' to measure the performance of their subgroup control chart. They simulated samples of various sizes from either a $\mathcal{N}(0,1)$ distribution or $Exp(1)$ distribution. The exact coverage probability of each set of control limits can be calculated by

$$CVG = P\left(LCL < \bar{X}_n < UCL\right),$$

where UCL and LCL are the given upper and lower control limits of the subgroup bootstrap chart and where $\bar{X}_n$ represents a future sample mean based on a sample of size n from the IC process distribution. A control chart is considered to perform well if $CVG \approx 1-\alpha$. [34] Seppala et al. (1995) calculated simulated coverage for the upper and lower control limits seperately and did not specify how they computed their measure of simulated coverage. [18] Jones et al. (1998) attempted to emulate the metric used by [34] Seppala et al. (1995) and found results consistent with theirs. Like [2] Bajgier (1992), [23] Liu et al. (1996) assumed that the Phase I data was obtained from a stable and IC process. They also assumed that the observations were independent. [23] Liu et al. (1996) constructed their bootstrap control chart as follows:

1. Observe m subgroups of size n for a total of $N = mn$ observations.
2. Compute $\bar{\bar{X}}_N = \frac{1}{N}\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}$.
3. Draw a random sample with replacement from the mn pooled observations. This sample, $x_1^*, x_2^*, \ldots, x_n^*$, is a bootstrap sample.
4. Compute the sample mean $\bar{x}^* = \frac{1}{n}\sum_{i=1}^{n} x_i^*$ and $t^* = \sqrt{n}\left(\bar{x}^* - \bar{\bar{X}}\right)$ from the bootstrap sample in step 3.
5. Repeat steps 3 and 4 a large number of times, say B times.
6. Sort the B bootstrap values $t_1^*, t_2^*, \ldots, t_B^*$ in ascending order.
7. Find the smallest ordered t^* such that $(\alpha/2)B$ values are below it. This value is denoted by $t^*_{\alpha/2}$.
8. Find the smallest ordered t^* such that $(1-\alpha/2)B$ values are below it. This value is denoted by $t^*_{1-\alpha/2}$.

Compute the lower and upper control limits using

$$LCL = \bar{\bar{X}}_N + \frac{t^*_{\alpha/2}}{\sqrt{n}} \tag{2.2}$$

and

$$UCL = \bar{\bar{X}}_N + \frac{t^*_{1-\alpha/2}}{\sqrt{n}}. \tag{2.3}$$

[18] Jones et al. (1998) argued that

$$t^*_{\alpha/2} = \sqrt{n}\left(\bar{x}^*_{\alpha/2} - \bar{\bar{X}}_N\right),$$

where $\bar{x}^*_{\alpha/2}$ was the smallest value of $\bar{x}^*$such that $(\alpha/2)B$ values were below it. Similarly,

$$t^*_{1-\alpha/2} = \sqrt{n}\left(\bar{x}^*_{1-\alpha/2} - \bar{\bar{X}}_N\right),$$

where $\bar{x}^*_{1-\alpha/2}$ is the smallest value of $\bar{x}^*$such that $(1-\alpha/2)B$ values are below it. Substituting the above results into the control limits for the control chart proposed by [23] Liu et al. (1996), i.e. LCL and UCL yields

$$LCL = \bar{x}^*_{\alpha/2}$$

and

$$UCL = \bar{x}^*_{1-\alpha/2}.$$

[23] Liu et al. (1996) discussed the asymptotic properties of their bootstrap control chart and used a simulation study to evaluate the performance of their chart. They only used one simulated set of control limits for each type of distribution and reached the conclusion that the bootstrap control limits are superior to the standard limits. This contradicts [2] Bajgier's (1992) findings, since he concluded that one needs more than a single sample of size $N = mn$ to measure the chart's performance, since the LCL and UCL are random variables.

[30] Nichols and Padgett (2006) used a parametric bootstrap to construct control charts for monitoring a specified percentile of the distribution of the process characteristic of interest. The chart proposed by [30] Nichols and Padgett (2006) was designed to be applied to small percentiles of the Weibull distribution. The probability density function (PDF) of the Weibull distribution is given by

$$f(w) = \frac{\delta}{\beta}\left(\frac{w}{\beta}\right)^{\delta-1}\exp\left[-\left(\frac{w}{\beta}\right)^{\delta}\right],\ w > 0\ (\delta,\ \beta > 0) \tag{2.4}$$

where δ and β are the shape and scale parameters, respectively. [30] Nichols and Padgett (2006) noted that the Weibull distribution could take a variety of shapes which could make it useful for a variety of applications. The $100p$th percentile for the Weibull distribution is given by $W_p = \beta\left[-\ln(1-p)\right]^{1/\delta}$ for $p \in (0,1)$. [30] Nichols and Padgett (2006) proposed that the following steps be followed to construct the bootstrap Weibull percentile control chart.

1. From an in-control, stable process, observe $n \times m$ observations from an assumed Weibull dis-

tribution with unknown shape parameter δ and unknown scale parameter β. The observations are denoted by x_{ij} for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$ and are assumed to come from m independent subgroups of size n.

2. Use maximum likelihood estimators (MLEs) for the unknown values of δ and β using the equations

$$\hat{\delta} = \left[\frac{\sum_{i=1}^{m} \sum_{i=1}^{n} x_{ij}^{\delta} \log x_{ij}}{\sum_{j=1}^{k} \sum_{i=1}^{n} x_{ij}^{\delta}} - \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \log x_{ij}}{nm} \right]^{-1} \quad \text{and} \quad \hat{\beta} = \left(\frac{\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}^{\delta}}{nm} \right)^{1/\delta}. \tag{2.5}$$

Here the Newton-Raphson method is used to find $\hat{\delta}$ from Equation (2.5) numerically.

3. Generate a parametric bootstrap of size n, $x_1^*, x_2^*, \ldots, x_n^*$, from the Weibull distribution using the MLEs obtained in step 2 as the Weibull parameters.

4. Find the parameter MLEs from the bootstrap subgroup and denote these as $\hat{\beta}^*$ and $\hat{\delta}^*$.

5. For the bootstrap subgroup, find $W_p^* = \hat{\beta}^* \left[-\ln (1-p) \right]^{1/\delta^*}$, the bootstrap estimate of the $100p$th percentile of interest, W_p.

6. Repeat steps 3 to 5 a larger number of times, say B, obtaining B bootstrap estimates of W_p, denoted by $W_{p1}^*, W_{p2}^*, \ldots, W_{pB}^*$.

7. Using the B bootstrap estimates obtained in step 6, find the $(\alpha/2) \times 100$th and $(1-\alpha/2) \times 100$th percentiles. Here $\alpha \in (0, 1)$ is the probability that an observation is considered to be OOC when the process is actually IC, i.e. α is the FAR.

8. The lower- (LCL) and upper control limits (UCL) are set equal to the $(\alpha/2) \times 100$th and $(1-\alpha/2) \times 100$th percentiles of the B bootstrap estimates respectively.

9. Once the control limits have been computed, future subgroups of size n are taken from the process at regular time intervals and W_p is estimated for each new subgroup by the MLEs indicated in step 5. If the estimate, $\hat{W}_p$, falls on or outside LCL or UCL then the process is declared to be out-of-control.

[21] Lio and Park (2010) proposed using a parametric bootstrap control chart to monitor percentiles by fitting a two-parameter inverse Gaussian distribution to the in-control reference sample and sampling from the fitted distribution, instead of the conventional sampling with replacement approach. The PDF for a two-parameter inverse Guassian distribution is given by

$$f(t; \nu, \lambda) = \sqrt{\frac{\lambda}{2\pi t^3}} \exp \left[-\lambda \frac{(t-\nu)^2}{2\nu^2 t} \right], \; t > 0, \tag{2.6}$$

where $\nu > 0$ is a location parameter and $\lambda > 0$ is a scale parameter. The cumulative distribution

function (CDF) is given by

$$F(t;\nu,\lambda) = \Phi\left[\sqrt{\frac{\lambda}{t}}\left(\frac{t}{\nu}-1\right)\right] + \exp\left(\frac{2\lambda}{\nu}\right)\Phi\left[-\sqrt{\frac{\lambda}{t}}\left(1+\frac{1}{\nu}\right)\right], \tag{2.7}$$

where $\Phi(.)$ is the standard normal CDF. Let $x_1, x_2, \ldots, x_n$ be a random sample of size $n \geq 1$ from the inverse Gaussian distribution with its PDF in Equation (2.6). The MLEs of the location parameter ν and scale parameter λ are given by

$$\hat{\nu} = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{2.8}$$

and

$$\hat{\lambda} = \frac{1}{(1/n)\sum_{i=1}^{n} x_i^{-1} - 1/\bar{x}}, \tag{2.9}$$

respectively. Let $F^{-1}(p;\nu,\lambda)$ denote the inverse function of the inverse Gaussian CDF, i.e. the $p \times 100$th percentile is obtained by solving the equation $F(t;\nu,\lambda) = p$ for t. Therefore, the MLE of $p \times 100$th percentile, denoted by $F^{-1}(p;\hat{\nu},\hat{\lambda})$, is also obtained by solving $F(t;\hat{\nu},\hat{\lambda}) = p$ for t. [21] Lio and Park (2010) constructed their bootstrap chart as follows:

1. Observe $m \geq 1$ random samples each of size n_j (for $j = 1, 2, \ldots, m$) from an in-control and stable process independently.

2. Using the MLEs given in Equations (2.8) and (2.9), calculate the MLEs of ν and λ with the pooled sample of size $N = \sum_{j=1}^{m} n_j$.

3. Generate a parametric bootstrap of size $n \geq 1$, $x_1^*, x_2^*, \ldots, x_n^*$, from the inverse Gaussian distribution using the MLEs obtained in step 2 as the inverse Gaussian parameters. Here n is the sample size that will be used for future subgroups.

4. Find the MLEs using the bootstrap sample in step 3 and denote these as $\hat{\nu}^*$ and $\hat{\lambda}^*$.

5. For the bootstrap subgroup sample obtained in step 3 and $\hat{\nu}^*$ and $\hat{\lambda}^*$ in step 4, find the bootstrap estimate, $\hat{W}_p^* = F^{-1}\left(p;\hat{\nu}^*,\hat{\lambda}^*\right)$, of the $100p$th percentile $\hat{W}_p = F^{-1}(p;\hat{\nu},\hat{\lambda})$.

6. Repeat steps 3 to 5 a large number of times, say B, obtaining B bootstrap estimates of $\hat{W}_p$, denoted by $\hat{W}_{p1}^*, \hat{W}_{p2}^*, \ldots, \hat{W}_{pB}^*$.

7. Using the B bootstrap estimates obtained in step 6, find the $(\alpha/2)\times 100$th and $(1-\alpha/2)\times 100$th percentiles. Here $\alpha \in (0,1)$ is the probability that an observation is considered to be OOC when the process is actually IC, i.e. α FAR. The lower- (LCL) and upper control limits (UCL) are set equal to the $(\alpha/2) \times 100$th and $(1-\alpha/2) \times 100$th percentiles of the B bootstrap estimates respectively.

8. Once the control limits have been computed, future subgroups of size n are taken from the process at regular time intervals and W_p is estimated for each new subgroup by the MLEs indicated in step 5. If the estimate, $\hat{W}_p$, falls on or outside LCL or UCL then the process is declared to be out-of-control.

2.2.2 Performance review and application of Shewart-type bootstrap control charts

[2] Bajgier (1992) noted that one cannot compare the performance of the bootstrap control chart using a single sample of mn in-control observations, since each set of mn observations will produce different bootstrap and standard control limits. [2] Bajgier (1992) simulated 1 000 sets of control limits for $m = 20$ samples of size $n = 5$ from a $\mathcal{N}(0, 1)$ and $\chi^2(5)$ distribution, respectively. Despite not providing any summary statistics from the simulation study, [2] Bajgier (1992) concluded that his bootstrap chart performs *'comparably'* to the standard methods in terms of the distribution of the ARL's.

In this mini-dissertation, the researcher investigates the implementation of [2] Bajgier's (1992) control chart by constructing two control charts: one being [2] Bajgier's (1992) bootstrap control chart and the other being the implementation of a Shewhart control chart with estimated parameters. [2] Bajgier (1992) constructed conventional control charts with estimated control limits using Equation (2.1). The researcher deviates from this and opt to use the following estimators

$$\hat{\mu}_0 = \bar{\bar{X}} = \frac{1}{m}\sum_{i=1}^{m}\bar{X}_i = \frac{1}{m}\sum_{i=1}^{m}\left[\frac{1}{n}\sum_{j=1}^{n}X_{ij}\right] = \frac{1}{mn}\sum_{j=1}^{m}\sum_{i=1}^{n}X_{ij} \tag{2.10}$$

and

$$\hat{\sigma}_0 = \frac{S_p}{c_{4,m}}, \tag{2.11}$$

where

$$S_p = \sqrt{\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(X_{ij} - \bar{X}_i)^2}{m(n-1)}} \tag{2.12}$$

and

$$c_{4,m} = \frac{\sqrt{2}\Gamma\left(\frac{m(n-1)+1}{2}\right)}{\sqrt{m(n-1)}\Gamma\left(\frac{m(n-1)}{2}\right)}. \tag{2.13}$$

The constant, $c_{4,m}$, in Equation 2.13 was used by [25] Mahmoud, Henderson, Epprecht and Woodall (2010) so that $\mathbb{E}\left[\frac{S_p}{c_{4,m}}\right] = \sigma_0$. [25] Mahmoud et al. (2010) showed that Equation (2.11) provides the lowest variance of all the commonly used estimators for σ_0.

The researcher generated a single set of $m = 20$ samples of size $n = 5$ from a $\chi^2(5)$ distribution.

Since we want to construct a standard 3σ Shewhart control chart, we set $\alpha = 0.0027$. Using Bajgier's (1992) algorithm, we drew 1 000 re-samples of size $n = 5$ from the set of $mn = 20(5) = 100$ in-control X_{ij}'s and calculated the sample mean for each of the 1 000 re-samples, i.e. $\bar{X}_1^{(1)}, \bar{X}_2^{(1)}, \ldots, \bar{X}_{1000}^{(1)}$.

We then find the ordered $\bar{X}_i^{(1)}$ value so that $(\alpha/2)B$ values are below it and we denote this value as $\bar{X}_{\alpha/2}^{(1)}$. We also find the ordered $\bar{X}_i^{(1)}$ value so that $(1-\alpha/2)B$ values are below it and we denote this value as $\bar{X}_{1-\alpha/2}^{(1)}$. We set $\hat{LCL}_1^* = \bar{X}_{\alpha/2}^{(1)}$ and $\hat{UCL}_1^* = \bar{X}_{1-\alpha/2}^{(1)}$. We repeat this process 1 000 times so that we have different sets of lower- and upper control limits. We obtain our final control limits by calculating

$$\hat{LCL}_{avg}^* = \frac{1}{1000}\sum_{i=1}^{1000} \hat{LCL}_i^* \tag{2.14}$$

and

$$\hat{UCL}_{avg}^* = \frac{1}{1000}\sum_{i=1}^{1000} \hat{UCL}_i^*. \tag{2.15}$$

The process used to calculate Equations (2.14) and (2.15) is summarised below.

Algorithm 2.1 Determining the control limits for [2] Bajgier's (1992) bootstrap control chart

---

1. Let $\alpha \in (0, 1)$.
2. Let $X_1^*, \ldots, X_n^*$ be a set of $n \geq 1$ values randomly drawn from $\{X_{11}, X_{12,} \ldots, X_{1n}, X_{21}, X_{22}, \ldots, X_{2n}, \ldots, X_{m1}, X_{m2}, \ldots X_{mn}\}$ with replacement.
3. Calculate $\bar{X}^* = \frac{1}{n}\sum_{i=1}^{n} X_i^*$.
4. Repeat steps 2 and 3 a large number of times, say $B_1 \geq 1\ 000$, to obtain B_1 $\bar{X}^*$ values, i.e. $\bar{X}_1^*, \bar{X}_2^*, \ldots, \bar{X}_{B_1}^*$.
5. Let $\hat{LCL}^*$ be the $(\alpha/2) \times 100$th percentile of $\bar{X}_1^*, \bar{X}_2^*, \ldots, \bar{X}_{B_1}^*$ and let $\hat{UCL}^*$ be the $(1-\alpha/2) \times 100$th percentile of $\bar{X}_1^*, \bar{X}_2^*, \ldots, \bar{X}_{B_1}^*$.
6. Repeat steps 2 to 5 a large number of times, say $B \geq 1\ 000$, to obtain B values of $\hat{LCL}^*$ and $\hat{UCL}^*$, i.e. $\hat{LCL}_1^*, \ldots, \hat{LCL}_B^*$ and $\hat{UCL}_1^*, \ldots, \hat{UCL}_B^*$.
7. Calculate

$$\hat{LCL}_{avg}^* = \frac{1}{B}\sum_{i=1}^{B} \hat{LCL}_i^*$$

and

$$\hat{UCL}_{avg}^* = \frac{1}{B}\sum_{i=1}^{B} \hat{UCL}_i^*.$$

The symmetric estimated control limits for the conventional two-sided Shewhart $\bar{X}$-chart are given by

$$\left(L\hat{C}L, U\hat{C}L\right) = \hat{\mu}_0 \pm k\frac{\hat{\sigma}_0}{\sqrt{n}}, \tag{2.16}$$

where $\hat{\mu}_0$ and $\hat{\sigma}_0$ are given by Equation (2.10) and Equation (2.11) respectively. The value $k > 0$ is chosen so that both the bootstrap and conventional control charts have the same in-control average run-length ARL_0. This is done so that a fair comparison can be made with respect to the performance of the charts in terms of detecting a shift in magnitude of $|\delta|$ standard error units $(\sigma_0/\sqrt{n})$ from the process mean.

A SAS v9.4 program was written to implement [2] Bajgier's (1992) control chart (see Section A.1.1 for the source code). We generated the aforementioned i.i.d. $\chi^2(5)$ random variables in $m = 20$ samples each of size $n = 5$. For the sake of illustration and without loss of generality, it can be assumed that these samples were generated from a *stable* and *in-control* process. Algorithm 2.1 was implemented and 1 000 sets of lower- and upper control limits were generated. After averaging over the 1 000 values we obtained

$$L\hat{C}L^*_{avg} = 1.9535515$$

and

$$U\hat{C}L^*_{avg} = 10.218125,$$

respectively. The standard deviations for the lower and upper control limits were estimated as

$$s_{L\hat{C}L^*_i} = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}\left(L\hat{C}L^*_i - L\hat{C}L^*_{avg}\right)^2} \approx 0.1607388$$

and

$$s_{U\hat{C}L^*_i} = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}\left(U\hat{C}L^*_i - U\hat{C}L^*_{avg}\right)^2} \approx 0.5151147,$$

respectively. We can further estimate the standard errors for $L\hat{C}L^*_{avg}$ and $U\hat{C}L^*_{avg}$ to be

$$SE\left(LCL^*_{avg}\right) \approx \frac{s_{L\hat{C}L^*_i}}{\sqrt{B}} = \frac{0.1607388}{\sqrt{1000}} \approx 0.0016$$

and

$$SE\left(UCL^*_{avg}\right) \approx \frac{s_{U\hat{C}L^*_i}}{\sqrt{B}} = \frac{0.5151147}{\sqrt{1000}} \approx 0.01629,$$

respectively. We will always get a different set of control limits for the same in-control reference sample (as noted by [2] Bajgier (1992)), so averaging over the 1 000 different lower- and upper control limits allows us to greatly reduce the variability of our results.

Since we know what the underlying process distribution is, the exact control limits can be determined using the fact that $n\bar{X}$ follows a $\chi^2(5n)$ distribution. So the *desired* lower- and upper control limits are given by

$$LCL_{desired} = \frac{\chi^2_{\alpha/2}(5n)}{n} = 1.791711742$$

and

$$UCL_{desired} = \frac{\chi^2_{1-\alpha/2}(5n)}{n} = 10.3182539,$$

respectively, where $\chi^2_{\alpha/2}(5n)$ and $\chi^2_{1-\alpha/2}(5n)$ are the $(\alpha/2)\times 100$th and $(1-(\alpha/2))\times 100$th percentiles of the $\chi^2(5n)$ distribution, respectively. The difference between the bootstrap control limits and the *desired* control limits are summarised in Table 2.1.

Table 2.1: Comparison of the bootstrap and exact control limits for the Shewhart $\bar{X}$ chart when the in-control process is $\chi^2(5)$ distributed

Control limit	Desired	Bootstrap	Absolute percentage difference (2 decimal places)
Lower	1.791711742	1.9535515	9.03%
Upper	10.3182539	10.218125	0.97%

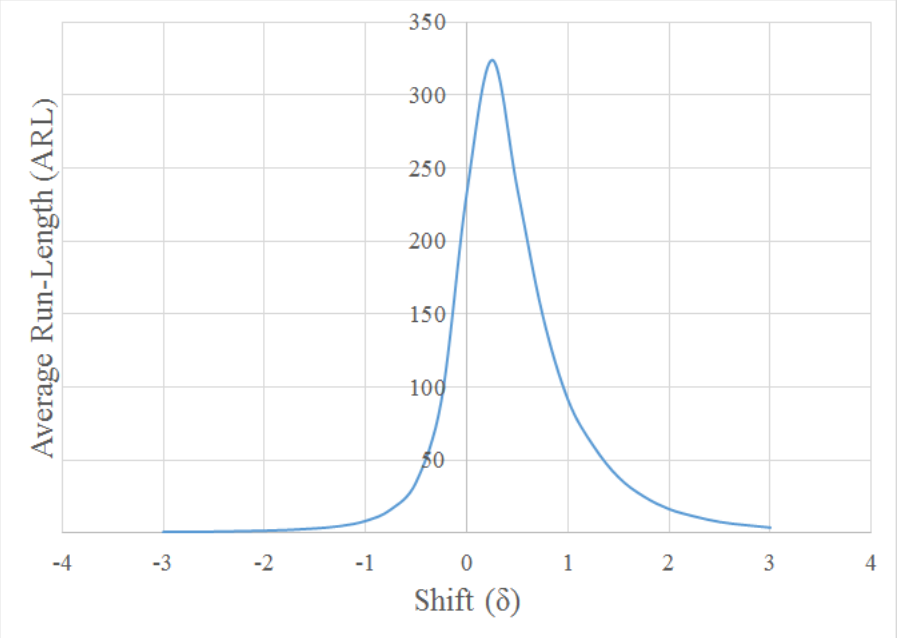
For this instance, the upper bootstrap control limit $U\hat{C}L_{avg}$ is very close to the desired upper control limit $UCL_{desired}$ (within 1% of its value), whilst the lower bootstrap control limit $L\hat{C}L^*_{avg}$deviates significantly (more than 5%) from the desired lower control limit $LCL_{desired}$. Undoubtedly this will have an impact on the false alarm rate and subsequently, the average run-length.

The distance between the bootstrap control charts is a little shorter than the distance between the *desired* control limits. One would expect a lower in-control average run-length for the bootstrap control chart (which we will denote by ARL^*_0). Since we know that $n\bar{X} \sim \chi^2(25)$ in this instance, we can calculate the false alarm rate for the bootstrap control chart, FAR^* as follows:

$$\begin{aligned}
FAR^* &= P\left(\text{Signal}|\text{IC}\right) \\
&= 1 - P\left(\text{No Signal}|\text{IC}\right) \\
&= 1 - P\left(L\hat{C}L^*_{avg} < \bar{X} < U\hat{C}L_{avg}|\text{IC}\right) \\
&= 1 - P\left(nL\hat{C}L^*_{avg} < n\bar{X} < nU\hat{C}L_{avg}|\text{IC}\right) \\
&= 1 - P\left(nL\hat{C}L^*_{avg} < \chi^2(5n) < nU\hat{C}L_{avg}\right) \\
&= 1 - P\left(9.7677575 < \chi^2(25) < 51.090625\right) \\
&\approx 0.0043.
\end{aligned}$$

From the above calculation, it follows that $ARL_0^* = 1/FAR^* \approx 231$, whereas the nominal in-control average run-length (by design) ARL_0 is supposed to be $1/0.0027 \approx 370$. We ran a simulation in SAS v9.4 (see Section A.1.1) to determine what the average run-length will be when a shift in magnitude of $|\delta| \leq 3$ standard deviations from the process mean is introduced. The results of this simulation are shown in Figure 2.1 .

Figure 2.1: Out-of-control values for the bootstrap Shewhart $\bar{X}$ control chart



From Figure 2.1 it may be noted that there is some bias in the ARL (the out-of-control ARL_δ is larger than ARL_0 for $0 < \delta < 0.5$). This can be attributed to the fact that the $\chi^2(5)$ (which is the underlying in-control process distribution) is a right-skewed distribution. Our simulation consisting of 10 000 different run-length simulations yielded an IC average run-length of 233.1591 which is lower than the desired average run-length of 370.4. This means that one could roughly expect 1 in every 233 samples to produce a false alarm (signaling that the process is out-of-control when it is in fact in-control). For a low volume process this could still be reasonable, but it can lead to serious and

unnecessary downtime in higher volume processes.

The researcher proceeded to compare the bootstrap chart to the more conventional two-sided symmetric control limit chart given by Equation (2.16). Using Equations (2.10) and (2.11) the following estimates for μ_0 and σ_0 were obtained

$$\hat{\mu}_0 = 5.0962183$$

and

$$\hat{\sigma}_0 = 3.1939624,$$

respectively. Using Equation (2.16) the lower- and upper control limits are obtained as

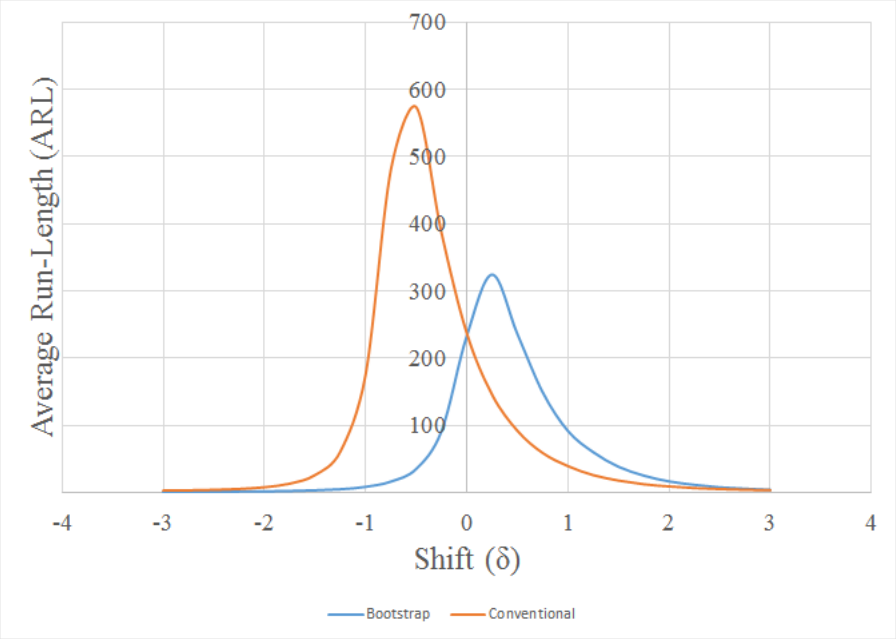
$$\begin{aligned} \hat{LCL} &= \hat{\mu}_0 - k\frac{\hat{\sigma}_0}{\sqrt{n}} \\ &= 0.6941761 \end{aligned}$$

and

$$\begin{aligned} \hat{UCL} &= \hat{\mu}_0 + k\frac{\hat{\sigma}_0}{\sqrt{n}} \\ &= 9.4982605, \end{aligned}$$

respectively. The researcher set $k = 3.0818352$ to produce an ARL_0 similar to that of the bootstrap chart, in order to compare the two opposing charts' abilities with respect to detecting shifts in the process mean. Another simulation study of 10 000 run-length simulations was conducted and the results were summarised in Figure 2.2.

Figure 2.2: Out-of-control values for the bootstrap and conventional Shewhart $\bar{X}$ control charts with estimated control limits



It may be noted that both the conventional and the bootstrap control charts have ARL bias. The conventional Shewhart chart with symmetrically placed control limits has a much higher degree of ARL bias than the bootstrap control chart. The bootstrap control chart also detects downward shifts ($\delta < 0$) in the process mean much faster than the conventional chart, whereas the conventional chart detects upward shifts ($\delta > 0$) much faster than the bootstrap control chart. The question is whether there is some truth to [2] Bajgier's (1992) statement that the bootstrap control chart performs similarly to the conventional chart. For a practitioner who is more interested in detecting drops in the process mean it most certainly seems like a good alternative in this instance.

Similarly, when the process followed a standard normal $\mathcal{N}(0,1)$ distribution, we implemented Algorithm (2.1) again and obtained the following values for the bootstrap lower- and upper control limits

$$\hat{LCL}^*_{avg} = -1.293245$$

and

$$\hat{UCL}^*_{avg} = 1.4620733,$$

respectively. The standard errors for $\hat{LCL}^*_{avg}$ and $\hat{UCL}^*_{avg}$ were

$$SE\left(LCL^*_{avg}\right) \approx \frac{s_{\hat{LCL}^*_i}}{\sqrt{B}} = \frac{0.0945022}{\sqrt{1000}} \approx 0.0030$$

and

$$SE\left(UCL^*_{avg}\right) \approx \frac{s_{U\hat{C}L^*_i}}{\sqrt{B}} = \frac{0.1255819}{\sqrt{1000}} \approx 0.0040,$$

respectively. Since we know that $\bar{X} \sim \mathcal{N}(0, \frac{1}{n})$ in this instance, we can calculate the exact (or desired) control limits as follows:

$$LCL_{desired} = \bar{x}_{\alpha/2} = -1.341630497$$

and

$$UCL_{desired} = \bar{x}_{1-a/2} = 1.341630497,$$

respectively, where $\bar{x}_p$ is the $p \times 100$th percentile of the $\mathcal{N}(0, \frac{1}{n})$ distribution. The difference between the bootstrap control limits and the *desired* control limits are summarised in Table 2.2.

Table 2.2: Comparison of the bootstrap and exact control limits for the Shewhart $\bar{X}$ chart when the in-control process is $\mathcal{N}(0, 1)$ distributed

Control limit	Desired	Bootstrap	Absolute percentage difference (2 decimal places)
Lower	-1.341630497	-1.293245	3.61%
Upper	1.341630497	1.4620733	8.98%

In this case the bootstrap overestimated the lower and upper control limits by 4% and 9%, respectively. It may also be noted that bootstrap control limits are further apart than the desired (or ideal) control limits. Naturally, this will lead to a higher in-control average run-length. In the simulation study an IC average run-length of approximately 405 was obtained using the bootstrap control limits. A two-sided Shewhart control chart, with symmetrically placed control limits and estimated parameters, was constructed using Equation (2.16) and k was chosen so that the chart has an in-control average run-length close to 405. The estimated lower- and upper control limits were obtained as

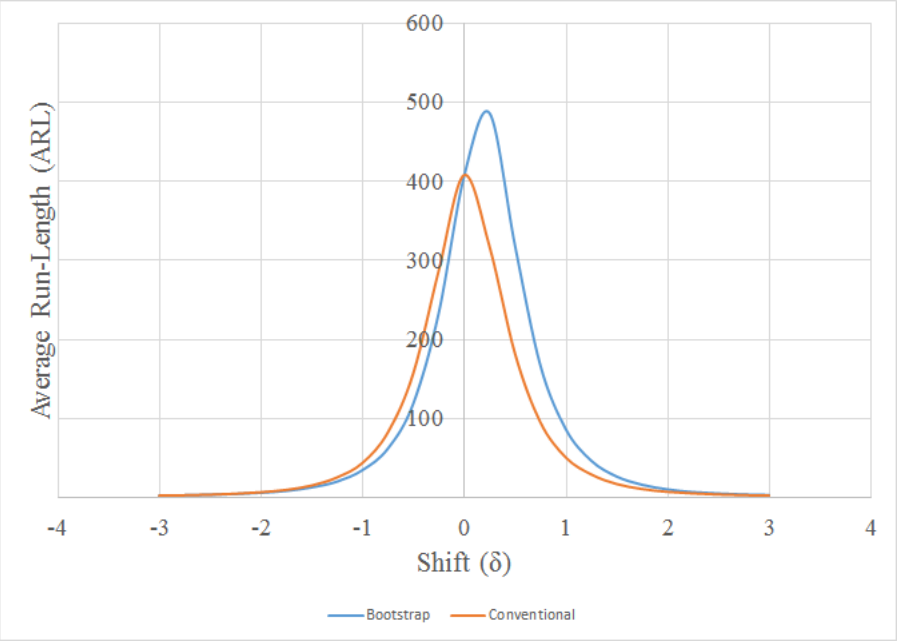
$$\hat{LCL} = -1.342541$$

and

$$\hat{UCL} = 1.3663521$$

respectively. A simulation study of 10 000 run-length simulations was conducted to test the performance of the chart when a shift in magnitude of $|\delta| < 3$ was introduced. The results are summarised in Figure 2.3.

Figure 2.3: Out-of-control values for the bootstrap and conventional Shewhart $\bar{X}$ control charts with estimated control limits



From Figure 2.3 it can be noted that the bootstrap control chart shows some slight ARL bias when $0 < \delta < 0.5$. This can be attributed to simulation error or the asymmetrically placed control limits (since the normal distribution is symmetrically distributed about its mean) of the bootstrap control chart. When $\delta > 0$, the conventional control limits outperform the bootstrap control limits, since the conventional chart responds to a shift in the process mean much faster than the bootstrap chart. However, when $\delta < 0$, the bootstrap control chart performs better than the conventional control chart.

The greatest strength of [2] Bajgier's (1992) bootstrap control chart is the relative ease with which it can be implemented and it is complemented by a set of less rigid and relaxed assumptions about the underlying process distribution. Unfortunately those same assumptions come with inherent weaknesses since one has to naively assume that the reference sample contains very few extreme observations that might have slipped past the initial Phase I analysis. [34] Seppala et al. (1995) pointed out that this could potentially lead to intervals that are too wide, resulting in an increased Type I error rate.

The method used by [34] Seppala et al. (1995) to construct the control limits of their bootstrap control chart (henceforth known as the subgroup bootstrap) was discussed in Section 2.1 of this mini-dissertation. [34] Seppala et al. (1995) attempted to account for OOC observations by subtracting the subgroup mean

$$\bar{X}_i = \frac{1}{n}\sum_{j=1}^{n} X_{ij} \text{ for } i = 1, 2, \ldots, m$$

from each observation and subsequently adding it to the grand mean ($\bar{\bar{X}}$). [34] Seppala et al. (1995)

measured the performance of their control chart by calculating the coverage probability for a large number of different control limits obtained through the subgroup bootstrap. The subgroup bootstrap control charts were constructed for various sized subgroups ($m = 5$ and $m = 20$) of different sample sizes ($n = 5$ and $n = 10$) from $\mathcal{N}(0, 1)$ and $Exp(1)$ populations, respectively. [34] Seppala et al. (1995) considered the cases where $\alpha = 0.0026$, 0.01, 0.05 and 0.1 respectively, and concluded that the subgroup bootstrap control chart outperforms the standard control chart (a Shewhart control chart with symmetrically placed control limits and estimated parameters) when the process is non-normally distributed.

[18] Jones et al. (1998) argued that the problem with the method proposed by [34] Seppala et al. (1995) was that the sample mean was sensitive to extreme values and simply adding the residual values to the grand mean did not lessen the impact of extreme points. [18] Jones et al. (1998) also criticised [34] Seppala et al. (1995) for calculating the coverage probability of the lower and upper control limits and not specifying how these values were computed.

To compare the subgroup bootstrap control chart of [34] Seppala et al. (1995) to the bootstrap control chart of [2] Bajgier (1992), the case where $\alpha = 0.0027$, $m = 20$, $n = 5$ and $X_{ij} \sim$ i.i.d. $Exp(1)$ for $i = 1, 2, \ldots, m$, and $j = 1, 2, \ldots, n$ is considered. Since it is known that

$$n\bar{X}_i = \sum_{j=1}^{n} X_{ij}, \text{ for } i = 1, 2, \ldots, m$$

follows a $Gamma(n, 1)$ distribution when $X_{ij} \sim$ i.i.d. $Exp(1)$ for $j = 1, 2, \ldots, n$, the coverage probability for the ith set of subgroup bootstrap control limits, given by $L\hat{C}L^*_{\text{sb},i}$ and $U\hat{C}L^*_{\text{sb},i}$, respectively, can be calculated as

$$\begin{aligned} CVG_i &= P\left(L\hat{C}L^*_{\text{sb},i} < \bar{X}_i \leq U\hat{C}L^*_{\text{sb},i} \mid \text{IC}\right) \\ &= P\left(nL\hat{C}L^*_{\text{sb},i} < n\bar{X}_i \leq nU\hat{C}L^*_{\text{sb},i} \mid \text{IC}\right) \\ &= P\left(2nL\hat{C}L^*_{\text{sb},i} < 2n\bar{X}_i \leq 2nU\hat{C}L^*_{\text{sb},i} \mid \text{IC}\right) \\ &= P\left(2nL\hat{C}L^*_{\text{sb},i} < \chi^2(2n) \leq 2nU\hat{C}L^*_{\text{sb},i} \mid \text{IC}\right). \end{aligned} \tag{2.17}$$

The result in Equation (2.17) follows from the fact that if $n\bar{X} \sim Gamma(n, 1)$ (where $n \in \mathbb{N}$), then $2n\bar{X} \sim \chi^2(2n)$. The FAR for the ith subgroup bootstrap control limits is given by

$$FAR^*_{\text{sb},i} = 1 - CVG_i$$

and subsequently, the IC ARL for the ith subgroup bootstrap control chart is given by

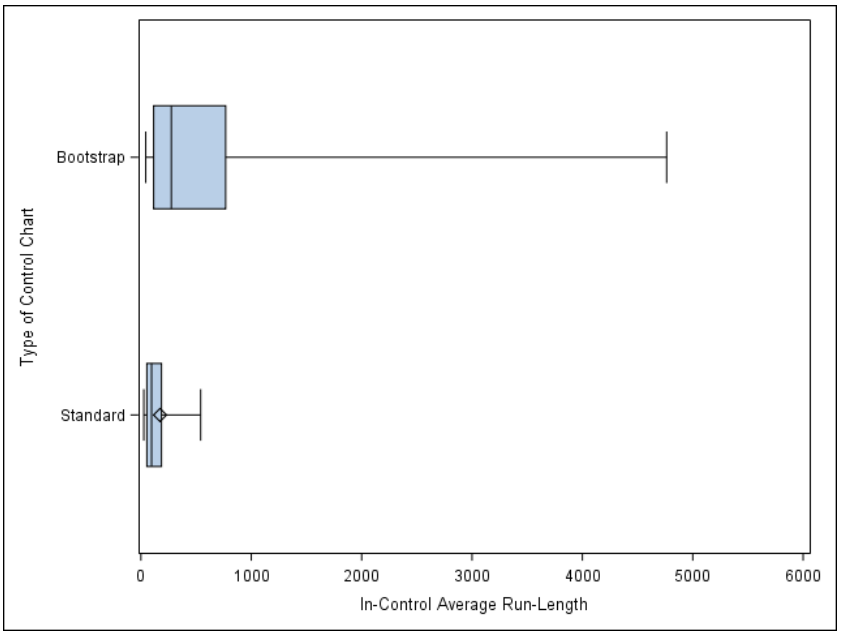
$$ARL^*_{\text{sb},i} = \frac{1}{FAR^*_{\text{sb},i}}. \tag{2.18}$$

To measure the performance of the subgroup bootstrap proposed by [34] Seppala et al. (1995), 10 000 sets of subgroup bootstrap control charts were constructed, by using 2 000 bootstrap replications, each from a different set of IC reference samples consisting of $m = 20$ subgroups of size $n = 5$, from an $Exp(1)$ population. Equations (2.17) and (2.18), respectively, were used to determine the attained conditional IC ARL for every set of control limits. The attained conditional IC ARL's of the subgroup bootstrap were compared to the *standard* control limits, which [34] Seppal et al. (1995) defined as

$$\left(\hat{LCL}_{\text{std}}, \hat{UCL}_{\text{std}}\right) = \bar{\bar{X}} + z_{1-\alpha/2}\left(\frac{S}{\sqrt{n}}\right), \tag{2.19}$$

where $\bar{\bar{X}}$ is the grand mean of the mn pooled IC observations, $z_{1-\alpha/2}$ is the $(1-\alpha/2)\times 100$th percentile of the standard normal distribution and S^2 is the sample variance of the mn pooled IC observations. The *standard* control limits were calculated, for each of the 10 000 sets of mn IC reference samples, and their coverage probabilities calculated by replacing $\left(\hat{LCL}_{\text{sb},i}, \hat{UCL}_{\text{sb},i}\right)$ with $\left(\hat{LCL}_{\text{std},i}, \hat{UCL}_{\text{std},i}\right)$ in Equation (2.17). The subgroup percentiles were constructed using 2 000 bootstrap replications. The attained conditional IC ARL for the *standard* control limits were then calculated, by using the coverage probabilities in Equation (2.18). A SAS v9.4 program was written (see Section A.1.2 for source code) and the results were summarised by the boxplot-like graphs in Figure 2.4.

Figure 2.4: Boxplot-like graphs of the attained conditional IC ARL for the subgroup bootstrap chart (first boxplot on top) and the *standard* chart (second boxplot at the bottom) when the IC process distribution is $Exp(1)$



Each boxplot shows the median as a middle band inside the box. The left and right sides of the box are the 25th and 75th percentiles, respectively. The "whiskers" are extended to the 5th and the 95th percentiles instead of the usual minimum and maximum. From Figure 2.4 it can be noted that the results for the subgroup bootstrap are much more variable than the results for the *standard* chart,

since the rectangle representing the middle 50% of the distribution of the subgroup bootstrap chart attained conditional IC *ARL*'s is wider than the rectangle of the bottom placed boxplot representing the middle 50% of the distribution of the *standard* chart's attained conditional IC *ARL*'s. It can also be seen in Figure 2.4 that the *standard* chart produces much more consistent results than the subgroup bootstrap chart. The summary statistics for the simulation study are provided in Table 2.3.

Table 2.3: Summary statistics for the attained IC conditional *ARL*'s of 10 000 different sets of subgroup bootstrap and *standard* control charts, when the IC process distribution is $Exp(1)$

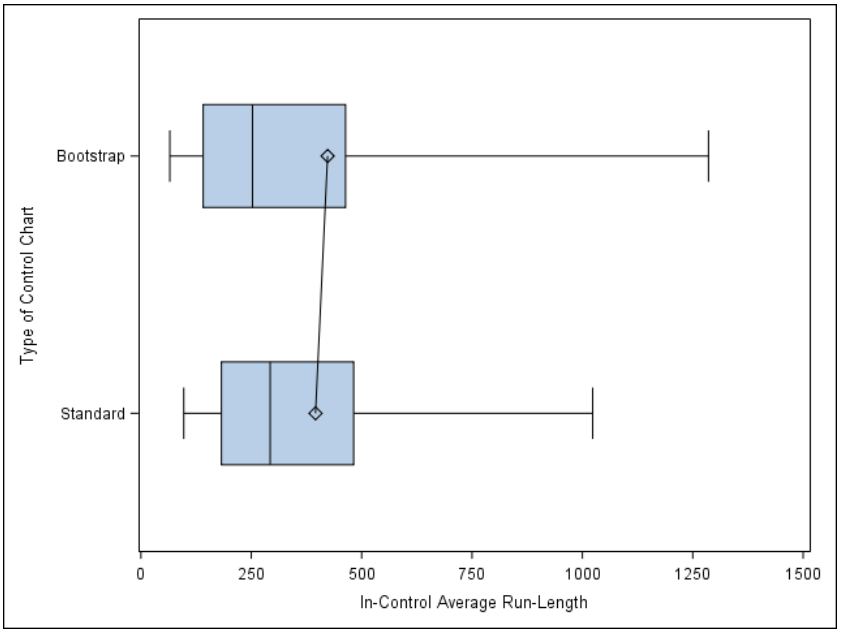
	ARL_{avg}	Standard error
Subgroup bootstrap	6978.1203	3320.0758
Standard	172.6726	2.8844

	5th Percentile	25th Percentile	50th Percentile	75th Percentile	95th Percentile
Subgroup bootstrap	43.6180	116.0971	275.3626	768.1300	4772.7674
Standard	25.9971	53.8048	96.6048	186.4389	541.6446

From Table 2.3 it can be seen that control charts constructed using the subgroup bootstrap proposed by [34] Seppala et al. (1995) yield much more variable results with respect to the IC *ARL* of the chart versus the *standard* chart. The standard error for IC *ARL* of the subgroup bootstrap is more than a thousand times that of the standard error for the *standard* control chart. It is extremely difficult to recommend the subgroup bootstrap of [34] Seppala et al. (1995) to a practitioner for this case.

The same procedure was used to compare the results of the subgroup bootstrap chart with the *standard* chart, when the IC process distribution is $\mathcal{N}(0, 1)$ (see Section A.1.2 for the SAS v9.4 source code). The results are summarised in the form of boxplot-like graphs in Figure 2.5.

Figure 2.5: Boxplot-like graphs of the attained in-control *ARL* of the subgroup bootstrap chart (first boxplot on top) and the *standard* chart (second boxplot at the bottom) when the process distribution is $\mathcal{N}(0,1)$



The boxplot-like graphs in Figure 2.5 are constructed similar to those in Figure 2.4, with the addition of a line segment connecting the average values of the attained IC conditional *ARL*'s for the respective control charts. In Figure 2.5 it may be noticed that the average attained IC conditional *ARL* for the subgroup bootstrap is a little bit larger than the average attained IC conditional *ARL* for the standard control chart. The results for the subgroup bootstrap control chart vary slightly more than the results of the *standard* control chart. The summary statistics for the simulation study are provided in Table 2.4.

Table 2.4: Summary statistics for the attained IC conditional *ARL*'s of 10 000 different sets of subgroup bootstrap and *standard* control charts, when the IC process distribution is $\mathcal{N}(0,1)$

	ARL_{avg}	Standard error
Subgroup bootstrap	422.8321	6.2070
Standard	395.4553	3.5418

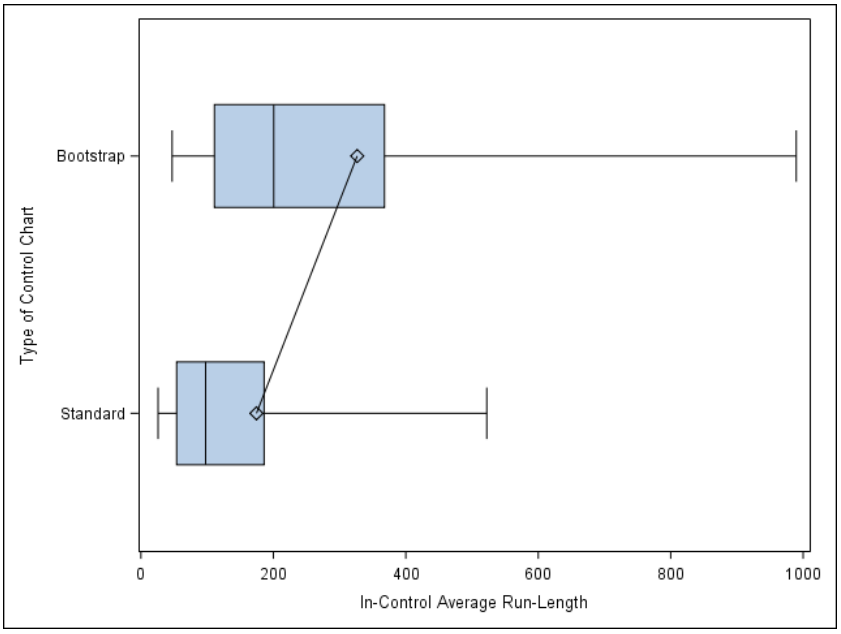
	5th Percentile	25th Percentile	50th Percentile	75th Percentile	95th Percentile
Subgroup bootstrap	65.3755	141.0838	252.7240	463.6236	1288.1164
Standard	97.0419	182.4611	292.7548	482.0677	1023.3255

The standard error for the average IC conditional *ARL* for the subgroup bootstrap is much lower for the $\mathcal{N}(0,1)$ case, compared to the $Exp(1)$ case in Table 2.3. The subgroup bootstrap is fairly consistent with the *standard* control chart in this case. It can be a viable alternative for the practitioner, but it should be done with caution since the results are easily influenced by extreme values and may vary significantly from one set of reference observations to another.

In Section 2.2.1 it was shown that the bootstrap control chart proposed by [23] Liu and Tang

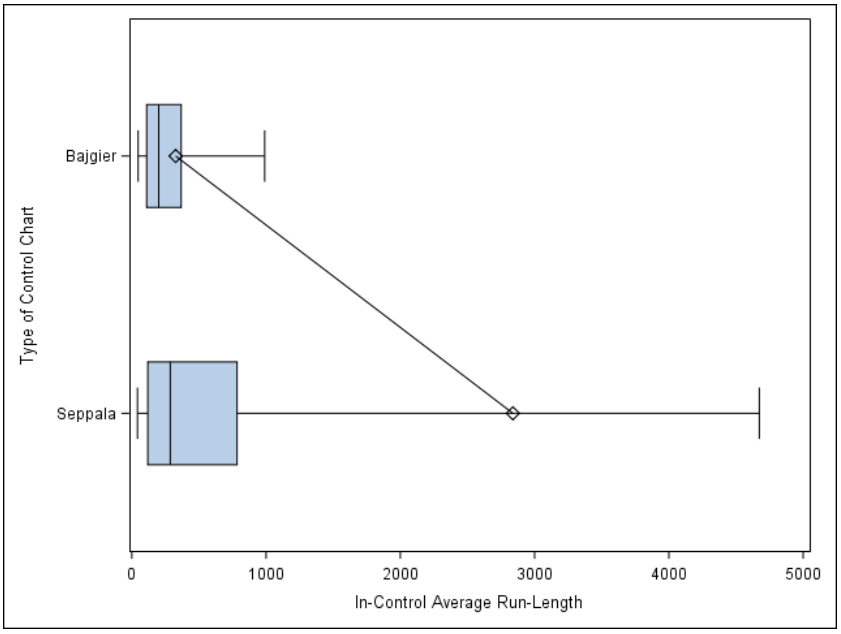
(1996) is equivalent to [2] Bajgier's (1992). The variability of the IC *ARL* of [2] Bajgier's (1992) and [23] Liu and Tang's (1996) bootstrap control chart will be investigated for the $Exp(1)$ distribution. A simulation study was conducted where 10 000 different sets of bootstrap and *standard* control limits were constructed from individual IC pooled observations of size mn (see Section A.1.1 for the source code). [2] Bajgier's (1992) control chart was compared to the *standard* control chart proposed by [34]Seppala et al. (1995). The results are summarised in the form of boxplot-like graphs in Figure 2.6.

Figure 2.6: Boxplot-like graphs of the attained IC conditional *ARL* for [2] Bajgier's (1992) bootstrap chart (first boxplot on top) and the *standard* chart (second boxplot at the bottom) when the IC process distribution is $Exp(1)$



[2] Bajgier (1992), as well as [23] Liu and Tang's (1996) control chart has more variable results than the *standard* control chart, but its results are less varied than those of [34] Seppala et al. (1995). The distribution of the attained IC conditional *ARL*'s for [2] Bajgier's (1992) bootstrap control chart is compared to those of [34] Seppala et al. (1995) in Figure 2.7.

Figure 2.7: Boxplot-like graphs of the attained IC conditional *ARL*'s for the [2] Bajgier's (1992) bootstrap chart (first boxplot on top) and the subgroup bootstrap chart (second boxplot at the bottom) by [34] Seppala et al. (1995) when the process distribution is $Exp(1)$



In Figure 2.7 it can be seen that the results for [2] Bajgier's (1992) control chart are much less variable than the subgroup bootstrap's. [2] Bajgier's (1992) bootstrap control chart has a much lower average IC *ARL* than the subgroup bootstrap control chart proposed by [34] Seppala et al. (1995).

2.2.3 Conclusion

[18] Jones et al. (1998) reviewed the performance of the control charts proposed by [2] Bajgier (1992), [34] Seppala et al. (1995) and [23] Liu and Tang (1996). [18] Jones et al. (1998) concluded that "bootstrap control charts do not perform substantially better than the standard method when performance of the charts is evaluated in terms of the resulting in-control *ARL*."

However, [18] Jones et al. (1998) also stated that:

"When estimating the tails of an extremely skewed distribution, the bootstrap techniques discussed here seem to produce estimates that are closer on average to the true quantile values than the standard Shewhart method."

[18] Jones et al. (1998) cautions against using bootstrap control limits, on the basis that they do not offer superior performance in terms of a predictable IC *ARL*. This is consistent with the findings of this mini-dissertation, as seen in Section 2.1.2. Attempts have been made to improve bootstrap confidence intervals, more specifically the bootstrap-t, the variance-stabilised bootstrap-t, the BCa (bias corrected and accelerated) and ABC (approximate bootstrap confidence) intervals.

Authors like[15] Iranpanah and Moghadam (2010) have applied these methods to the conventional stable, independent and IC setting assumed by [2] Bajgier (1992). A brief discussion on the application of the BCa bootstrap and the average attained IC *ARL*'s for a control chart constructed using the BCa follows.

An implementation of the BCa proposed by [10] Efron (1987) for a Shewhart control chart is as follows:

Let $\bar{X}_1^*, \bar{X}_2^*, \ldots, \bar{X}_B^*$ be B bootstrap replications of the sample mean for a bootstrap sample, i.e. $X_1^*, X_2^*, \ldots, X_n^*$, where

$$P\left(X_i^* = X_{jk}|\chi\right) = \frac{1}{mn} \text{ for } i,\ k = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, m$$

and χ is the set of IC observations obtained during a Phase I analysis, i.e. $\chi = \{X_{i1}, \ldots, X_{in}\}_{i=1}^{m}$. Using [2] Bajgier's (1992) approach, we calculate

$$\left(\hat{LCL}, \hat{UCL}\right) = \left(\bar{X}_{\alpha/2}^*, \bar{X}_{1-\alpha/2}^*\right),$$

where $\bar{X}_{\alpha/2}^*$ and $\bar{X}_{1-\alpha/2}^*$ are the $(\alpha/2)\times 100$th and $(1-\alpha/2)\times 100$th sample percentiles of $\bar{X}_1^*, \bar{X}_2^*, \ldots, \bar{X}_B^*$, respectively. Using the approach proposed by [10] Efron (1987), the control limits that produce a FAR of $\alpha \in (0, 1)$, are given by

$$\left(\hat{LCL}_{\text{BCa}}, \hat{UCL}_{\text{BCa}}\right) = \left(\bar{X}_{\alpha_1}^*, \bar{X}_{\alpha_2}^*\right), \tag{2.20}$$

where

$$\begin{aligned} \alpha_1 &= \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z_{\alpha/2}}{1 - \hat{a}\left(\hat{z}_0 + z_{\alpha/2}\right)}\right) \\ \alpha_2 &= \Phi\left(\hat{z}_0 + \frac{\hat{z}_0 + z_{1-\alpha/2}}{1 - \hat{a}\left(\hat{z}_0 + z_{1-\alpha/2}\right)}\right). \end{aligned} \tag{2.21}$$

Here $\Phi(.)$ is the standard normal CDF, z_p is the $p \times 100$th percentile of a standard normal distribution, $\hat{z}_0$ is an estimator for the bias-correction factor and [10] Efron (1987) referred to $\hat{a}$ as the acceleration. We can compute $\hat{z}_0$ in Equation (2.21) as follows

$$\hat{z}_0 = \Phi^{-1}\left(\frac{\sum_{i=1}^{B} \mathbf{1}\left(\bar{X}_i \leq \bar{\bar{X}}\right)}{B}\right), \tag{2.22}$$

where $\mathbf{1}(A) = 1$ when A is true; 0 when A is false, $\bar{\bar{X}}$ is the average of all the values in the IC reference sample and $\Phi^{-1}(.)$ is the inverse of the CDF for a standard normal distribution. Let $\chi_{(i,j)}$ be the original sample, but with the (i, j)th element removed, i.e.

$$\chi_{(i,j)} = \{X_{11}, X_{12}, \ldots, X_{1n}, \ldots, X_{i1}, X_{i2}, \ldots, X_{i,j-1}, X_{i,j+1}, \ldots, X_{in}, \ldots, X_{m1}, \ldots, X_{mn}\}.$$

Let $\bar{X}_{(i,j)}$ be the mean of the sample with the (i,j)th element removed, i.e. $\chi_{(i,j)}$. We can compute $\hat{a}$ in Equation (2.21) as follows

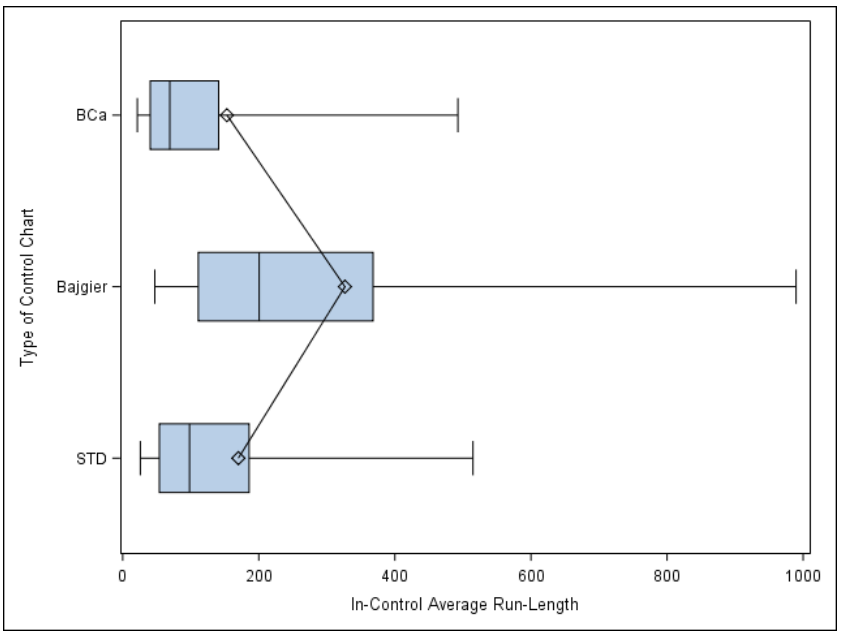
$$\hat{a} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(\bar{X}_{(.)} - \bar{X}_{(i,j)}\right)^3}{6\left\{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(\bar{X}_{(.)} - \bar{X}_{(i,j)}\right)^2\right\}^{3/2}}, \tag{2.23}$$

where

$$\bar{X}_{(.)} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\bar{X}_{(i,j)}.$$

For illustrative purposes, 10 000 IC reference samples from an $Exp(1)$ distribution, each consisting of $m = 20$ subgroups of size $n = 5$ were generated. For each of these samples, 10 000 sets of [2] Bajgier's (1992) bootstrap control limits were constructed, together with their BCa counterparts. The bootstrap control limits were obtained by using 2 000 bootstrap replications. The control charts were designed to have an ARL_0 of $1/0.0027 \approx 370.37$. The conditional ICARL's were obtained for each set of limits, and the results were summarised in the form of boxplot-like graphs in Figure 2.8. The SAS v9.4 source code used to obtain the results can be found in Section A.1.3.

Figure 2.8: Boxplot-like graphs of the attained in-control ARL's for the BCa bootstrap chart (first boxplot on top), [2] Bajgier's (1992) bootstrap control chart (second boxplot) and the standard Shewhart control chart (last boxplot at the bottom with estimated parameters), when the process distribution is $Exp(1)$



It can be seen in Figure 2.8 that the IC conditional ARL for the control chart using the BCa bootstrap control limits is much less variable than the control chart constructed using [2] Bajgier's (1992) bootstrap approach. The BCa bootstrap control chart has approximately the same consistency, with respect to the IC conditional ARL, as the standard Shewhart chart with estimated parameters. The BCa bootstrap control chart produces, on average, a much lower IC conditional ARL than [2] Bajgier's (1992) bootstrap control chart.

[27] Montgomery (2009) stated that the Shewhart control chart is optimal with respect to detecting larger (approximately 3 standard deviations above the mean) shifts in the process mean. The bootstrap control charts are quite variable with respect to their attained IC conditional ARL's and one should exercise caution when implementing them, which is supported by authors such as [18] Jones et al. (1998). Using the BCa, one could implement a bootstrap chart in a Phase I setting, where it could prove useful in detecting extreme values.

2.3 CUSUM control chart

2.3.1 Overview of Bootstrap Control Chart Designs

[39] Yashchin (1992) used a Markov-chain approach to estimate the run-length characteristics (the *FAR* and the *coefficient of variation* for the run-length). [39] Yashchin (1992) used a technique called the jackknife to obtain confidence intervals for the run-length characteristics. Suppose we have a random sample $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ and an estimator $\hat{\theta} = T(X_1, X_2, \ldots, X_n)$. To estimate the bias and standard error of $\hat{\theta}$, [11] Efron and Tibshirani (1994) argued that we could focus on samples that *leave out one observation at a time*:

$$\boldsymbol{X}_{(i)} = (X_1, X_2, \ldots, X_{i-1}, X_{i+1}, \ldots X_n) \text{ for } i = 1, 2, 3, \ldots, n.$$

These samples are referred to as jackknife samples by [11] Efron and Tibshirani (1994). Let

$$\hat{\theta}_{(i)} = T(\boldsymbol{X}_{(i)}) \tag{2.24}$$

be the ith jackknife replication of $\hat{\theta}$. [11] Efron and Tibshirani (1994) defined the jackknife estimate of bias by

$$\hat{\text{bias}}_{\text{jack}} = (n-1)(\hat{\theta}_{(.)} - \hat{\theta}) \tag{2.25}$$

where

$$\hat{\theta}_{(.)} = \frac{1}{n}\sum_{i=1}^{n} \hat{\theta_{(i)}}.$$

[11] Efron and Tibshirani (1994) defined the jackknife estimate of standard error by

$$\hat{\text{se}}_{\text{jack}} = \left[\frac{n-1}{n}\sum_{i=1}^{n}\left(\hat{\theta}_{(i)} - \hat{\theta}_{(.)}\right)^2\right]^{1/2}. \tag{2.26}$$

Assume that $X_1, X_2, X_3, \ldots$ form a sequence of i.i.d. random variables with CDF (x). [31] Page (1954) defined the upper CUSUM scheme in terms of three parameters, namely, $h \geq 0$ (control limit), $k \in \mathbb{R}$ (the *reference* or *target* value) and $0 \leq s_0 \leq h$ (with s_0 the head start). The charting statistic is defined as:

$$S_i = \max\{0, S_{i-1} + (X_i - k)\}, \text{ for } i = 1, 2, 3, \ldots, \tag{2.27}$$

where

$$S_0 = s_0.$$

When $S_i \geq h$ for some $i \in \mathbb{N}$, the process is declared to be OOC. [39] Yaschin (1992) argued that for a given CDF, $F(x)$, the run-length characteristics could be computed by using the approach suggested by [3] Brook and Evans (1972). [3] Brook and Evans (1972) proposed to subdivide the interval $(0, h)$ into d (d is called the *level of discretization*) subintervals of length $\Delta = h/(d - 0.5)$. The transition matrix $\boldsymbol{P}$ of the Markov-chain corresponding to the upper [31] Page's (1954) scheme with discretized states is expressed in terms of $F(x)$ as follows

$$\boldsymbol{P}_{d+1} = \begin{pmatrix} \boldsymbol{R} & (\boldsymbol{I} - \boldsymbol{R})\mathbf{1} \\ \mathbf{0}^T & 1 \end{pmatrix}, \tag{2.28}$$

where the elements r_{ij} $(i\ ,\ j = 0, 1, 2, \ldots, d-1)$ of $\boldsymbol{R}$ are given by

$$r_{ij} = \begin{cases} F\left(k + (-i + 0.5)\Delta\right) & ,\ \text{for}\ \ j = 0 \\ F\left(k + (j - i + 0.5)\Delta\right) - F\left(k + (j - i - 0.5)\Delta\right) & ,\ \text{for}\ \ j > 0, \end{cases} \tag{2.29}$$

where $\mathbf{1}$ is a vector of ones, $\boldsymbol{I}$ is a $(d \times d)$ identity matrix and $\mathbf{0}^T$ is a vector consisting of zeroes. If $\boldsymbol{K} = (\boldsymbol{I} - \boldsymbol{R})^{-1}$, then the vector containing ARL's corresponding to the head starts $0, \Delta, 2\Delta, \ldots, (d-1)\Delta$ and the similarly defined vector of second moments around 0 are defined by

$$\boldsymbol{\mu} = \boldsymbol{K}\mathbf{1} \tag{2.30}$$

and

$$\boldsymbol{\mu}_2 = 2\boldsymbol{K}\boldsymbol{\mu} - \boldsymbol{\mu}, \tag{2.31}$$

respectively. The CDF for the run-length (RL) for all values of the head start can be calculated from the formula

$$P\left(RL \leq n\right) = 1 - \boldsymbol{R}^n\mathbf{1}. \tag{2.32}$$

To obtain point estimates of the run-length, [39] Yashchin (1992) simply substituted $F(x)$ with the EDF $\hat{F}(x)$ in Equation (2.28). Suppose that $\boldsymbol{\mu} = (\mu_{11}, \mu_{12}, \ldots, \mu_{1d})^T$ (Equation (2.30)) and $\boldsymbol{\mu}_2 = (\mu_{21}, \mu_{22}, \ldots, \mu_{2d})^T$. [39] Yashchin (1992) used two performance characteristics $\boldsymbol{\theta}$ and ρ, where

$$\boldsymbol{\theta} = \left(\frac{1}{\mu_{11}}, \frac{1}{\mu_{12}}, \ldots, \frac{1}{\mu_{1d}}\right)^T \tag{2.33}$$

and

$$\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_d)^T,$$

respectively, where K_{ij} is the (i,j)th element of the matrix $\boldsymbol{K}$ and

$$\rho_i = 2 - \frac{2\sum_{j=1}^{d} K_{ij}\mu_{1j}}{\mu_{1i}^2}, \text{ for } i = 1, 2, \ldots, d. \tag{2.34}$$

[39] Yashchin (1992) used $(\hat{\theta}, \hat{\rho}) = \left(\frac{1}{\hat{\mu}_{11}}, \hat{\rho}_1\right)$ (this was done by replacing the CDF with the EDF) and calculated confidence intervals for these parameters, by removing a sample element and then proceeding to re-calculate the jackknife EDF. This was done n times and confidence intervals were obtained for the the differentARL values. [11] Efron and Tibshirani (1994, p142) argued that "the jackknife makes a *linear approximation* to the bootstrap."

However, we are only concerned with the application of the nonparametric bootstrap using Monte Carlo simulations and [39] Yaschin's (1992) approach falls outside the scope of this mini-dissertation.

[7] Chatterjee and Qiu (2009) proposed using a sequence of control limits for the CUSUM control chart, where the control limits were determined by the conditional distribution of the CUSUM charting statistic given the last time it was zero (henceforth known as the *sprint length*) and the control limits were determined by bootstrap. Suppose a sequence of i.i.d. random variables $\{X_n\}_{n=1}^{\infty}$ on the real line is observed, such that $X_1, X_2, \ldots, X_{t_0}$ follow a given distribution F, called the IC distribution, and $X_{t_0+1}, X_{t_0+2}, \ldots$ follow another distribution G, called the OOC distribution, where $F \neq G$. SPC techniques are often implemented to detect these distribution shifts as soon as possible. To detect an upward shift, the CUSUM charting statistic C_n is defined by

$$\begin{aligned} C_0 &= 0, \\ C_n &= \max\left(C_{n-1} + X_n - k,\ 0\right) \text{ for } n \geq 0, \end{aligned} \tag{2.35}$$

where $k \geq 0$ is a pre-specified *allowance constant.* The process is declared OOC if $C_n \geq H$, where H is a control limit determined by setting the IC ARL to a certain nominal level ARL_0, and the in-control ARL is defined to be the expected time to signal under F, that is

$$ARL = \mathbb{E}_F\left[\inf\left\{n > 0 \mid C_n > h\right\}\right]. \tag{2.36}$$

[7] Chatterjee and Qiu (2009) defined the *sprint length* statistic as

$$T_n = \begin{cases} 0, & \text{if } C_n = 0 \\ j, & \text{if } C_n \neq 0, \ldots, C_{n-j+1} \neq 0, C_{n-j} = 0, \text{ for } j = 1, 2, \ldots, n. \end{cases} \tag{2.37}$$

The procedure proposed by [7] Chatterjee and Qiu (2009) assumes that a set of IC process data was collected during a Phase I analysis. Let Y_j be a random variable having the distribution of $[C_n|T_n = j]$. For any positive integer $j_{max} \leq n$, the distribution of C_n equals that of

$$\sum_{j=1}^{j_{max}} Y_j \mathbf{I}(T_n = j) + Y^* \mathbf{I}(T_n > j_{max}),$$

where Y^* is a random variable with the distribution of $[C_n|T_n > j_{max}]$. It can be argued that it is reasonable to choose the control limit h_j (h^*) based on the distribution of Y_j (Y^*). [7] Chatterjee and Qiu (2009) declared the process to be OOC, at time point n, if $T_n = j$ and $C_n > h_j$, for $1 \leq j \leq j_{max}$, or if $T_n > j_{max}$ and $C_n > h^*$. The constants j_{max} and k are the two tuning parameters of this procedure whose choice is up to the practitioner. The control limits $\{h_j, 1 \leq j \leq j_{max}; h^*\}$ are obtained using the bootstrap, the choice of j_{max} is limited by the IC data, the allowance constant k and the computational power available to the practitioner. [7] Chatterjee and Qiu (2009) showed that a high j_{max} value does not result in the most efficient bootstrap based SPC. It was found that results seem to be fairly stable for $20 \leq j_{max} \leq 50$. In conventional CUSUMs, the selection of k is related to the desired magnitude of the shift to be detected by the process, i.e. δ. Because of the desired robustness against non-normality, [7] Chatterjee and Qiu (2009) argued that the choice for k should be related to the average sprint length, i.e. $\mathbb{E}[T_n]$.

From Equations (2.35) and (2.37) it can be seen that if k is chosen larger, then C_n will have a larger chance to bounce back to 0. Consequently, $\mathbb{E}[T_n]$ will be smaller. Conversely, if k is chosen smaller then $\mathbb{E}[T_n]$ will be larger. [7] Chatterjee and Qiu (2009) considered three choices for $\mathbb{E}[T_n]$, namely $\mathbb{E}[T_n] = 0.5j_{max}$, $\mathbb{E}[T_n] = 0.75j_{max}$ and $\mathbb{E}[T_n] = j_{max}$, respectively. The value of k was obtained from $\mathbb{E}[T_n]$ using an iterative procedure, explained by Algorithm 2.2.

Algorithm 2.2 Obtaining a value for k using $\mathbb{E}[T_n]$

1. Let k_L, k_U and k_0 be lower-bound, upper-bound and an initial value for k.
2. Draw B bootstrap samples from the normalised IC data (i.e. having zero sample mean and unit sample variance).
3. The CUSUM procedure uses allowance constant k_0.
4. Record the first sprint length of the CUSUM for each of the B bootstrap samples.
5. Estimate $\mathbb{E}[T_n]$ with the sample mean of the B sprint length values.
6. If the estimated $\mathbb{E}[T_n]$ value is larger than the target $\mathbb{E}[T_n]$ value, then we update k to be $k_1 = (k_U + k_0)/2$, and use k_0 and k_U as the new lower and upper bounds. Otherwise, update k to be $k_1 = (k_L + k_0)/2$ and use k_L and k_0 as the new lower and upper bounds. Set $k_0 = k_1$.
7. Repeat steps 2 to 6 until the estimated $\mathbb{E}[T_n]$ value in an iteration is sufficiently close to the target $\mathbb{E}[T_n]$ value.

[7] Chatterjee and Qiu (2009) set k_L, k_U and k_0 equal to the first, third and second quartiles of the IC data and $B = 5\,000$. The control limits $\{h_j, 1 \leq j \leq j_{max}; h^*\}$ can be determined from the IC

data using the bootstrap, once j_{max} and k are fixed. [7] Chatterjee and Qiu (2009) used a smoothed bootstrap as discussed by [37] Silverman (1986). The algorithm consists of two steps. In the first step the bootstrap is used for obtaining preliminary values $\{M_j, 1 \leq j \leq j_{max}; M^*\}$, such that $M_j \approx h_j$ and $M^* \approx h^*$. Then, in the second step these values are calibrated using some more bootstrap steps to ensure that the resulting IC ARL equals the nominal ARL_0 up to a certain level of accuracy.

Algorithm 2.3 Obtaining initial values for $\{M_j, 1 \leq j \leq j_{max}; M^*\}$

1. Let B be the bootstrap Monte Carlo sample size, $C^*_{\text{old}} = 0$, $T^*_{\text{old}} = 0$ and $b = 0$. For all $j \in \{1, 2, \ldots j_{max} + 1\}$ we implement steps 2 to 5.
2. Set $b = b + 1$.
3. Draw an observation X^* from $\hat{F}_0$.
4. Update C^*_{old} to $C^*_{new} = \max(C^*_{old} + X^* - k, 0)$. If $C^*_{\text{old}} > 0$, then compute T^*_{new} by $T^*_{\text{new}} = T^*_{\text{old}} + 1$. If $C^*_{\text{new}} = 0$, then set $T^*_{\text{new}} = 0$.
5. Check if $T^*_{\text{new}} = j$. If so, then record $Y_{j:b} = C^*_{\text{new}}$. If not, then set $C^*_{\text{old}} = C^*_{\text{new}}$ and $T^*_{\text{old}} = T^*_{\text{new}}$, and go to step 1. If $b < B$, go to step 2.
6. Define
$$\hat{\alpha} = \left(\hat{p}^2 ARL_0\right)^{-1},$$
where $\hat{p}$ denotes the proportion of observations in the IC data that are larger than k.
7. The $B(1-\hat{\alpha})$th ordered value from $Y_{j:1}, Y_{j:2}, \ldots, Y_{j:B}$ is taken as M_j for $j = 1, 2, \ldots, j_{max}$. The $B(1-\hat{\alpha})$th ordered value from $Y_{(j_{max}+1):1}, \ldots, Y_{(j_{max}+1):B}$ is taken as M^*.

To *fine tune* $M_1, M_2, \ldots, M_{j_{max}}$ and M^* to obtain $h_1, h_2, \ldots, h_{j_{max}}$ and h^* so that the nominal ARL_0 is reached we need to use an iterative approach as recommended by [7] Chatterjee and Qiu (2009). This approached is summarised by Algorithm 2.4.

Algorithm 2.4 Fine tuning M_j's and M^* to $h_1, h_2, \ldots, h_{j_{max}}$ and h^*

1. Define $h_j^{(0)} = M_j$ for $j = 1, 2, \ldots, j_{max}$ and $h^{*(0)} = M^*$ (as obtained in Algorithm (2.3)).
2. Let $n = 0$, $k = 0$, $C_0 = 0$ and $T_0 = 0$.
3. Set $n = n + 1$.
4. For $n \geq 1$ generate X_n^* from $\hat{F}$ (where $\hat{F}$ is the AKDE for the IC reference sample). Construct $C_n = \max\{C_{n-1} + X_n^* - k, 0\}$ and keep track of the corresponding *sprint length* T_n (as defined by Equation (2.37)). If $T_n = j$ and $C_n > h_j^{(0)}$ then process is declared OOC. If $T_n > j_{max}$ and $C_n > h^{*(0)}$ then the process is also declared OOC and take the run-length as n.
5. Repeat steps 2 to 4 N_1 times.
6. Set $RL^{(0)}$ equal to the average of the N_1 run-lengths obtained in steps 2 to 5.
7. Set $k = k + 1$.
8. If $RL^{(k-1)} < ARL_0$, then repeat steps 2 to 6 after $h_1^{(k-1)}, h_2^{(k-1)}, \ldots, h_{j_{max}}^{(k-1)}$ and $h^{*(k-1)}$ are replaced by $h_{1U}^{(k-1)} = (1+\varepsilon)M_1, h_{2U}^{(k-1)} = (1+\varepsilon)M_2, \ldots, h_{j_{max}U}^{(k-1)} = (1+\varepsilon)M_{j_{max}}$ and $h_U^{*(k-1)} = (1+\varepsilon)M^*$, where $\varepsilon > 0$ is a parameter. The corresponding average run-length is denoted by $RL_U^{(k-1)}$. Define

$$\begin{aligned} h_j^{(k)} &= \frac{RL_U^{(k-1)} - ARL_0}{RL_U^{(k-1)} - RL^{(k-1)}} h_j^{(k-1)} + \frac{ARL_0 - RL^{(k-1)}}{RL_U^{(k-1)} - RL^{(k-1)}} h_{jU}^{(k-1)}, \text{ for } j = 1, 2, \ldots, j_{max}, \\ h^{*(k)} &= \frac{RL_U^{(k-1)} - ARL_0}{RL_U^{(k-1)} - RL^{(0)}} h^{*(k-1)} + \frac{ARL_0 - RL^{(k-1)}}{RL_U^{(k-1)} - RL^{(k-1)}} h_U^{*(k-1)}. \end{aligned} \tag{2.38}$$

   If $RL^{(0)} < ARL_0$, then run the CUSUM procedure in steps 2 to 6 using control limits $h_{1L}^{(k-1)} = (1-\varepsilon)h_1^{(k-1)}, h_{2L}^{(k-1)} = (1-\varepsilon)h_2^{(k-1)}, \ldots, h_{j_{max}L}^{(k-1)} = (1-\varepsilon)h_{j_{max}}^{(k-1)}$ and $h_L^{*(k-1)} = (1-\varepsilon)h^{*(k-1)}$, where $\varepsilon > 0$ is a parameter. The corresponding average run-length is denoted by $RL_L^{(0)}$. Define

$$\begin{aligned} h_j^{(k)} &= \frac{ARL_0 - RL_L^{(k-1)}}{RL^{(k-1)} - RL_L^{(k-1)}} h_j^{(k-1)} + \frac{RL^{(k-1)} - ARL_0}{RL^{(k-1)} - RL_L^{(0)}} h_{jL}^{(k-1)}, \text{ for } j = 1, 2, \ldots, j_{max}, \\ h^{*(k)} &= \frac{ARL_0 - RL_L^{(k-1)}}{RL^{(k-1)} - RL_L^{(k-1)}} h^{*(k-1)} + \frac{RL^{(k-1)} - ARL_0}{RL^{(k-1)} - RL_L^{(k-1)}} h_L^{*(k-1)}. \end{aligned} \tag{2.39}$$

9. Repeat steps 2 to 8 until

$$\frac{\left|RL^{(k)} - ARL_0\right|}{ARL_0} < \tilde{\varepsilon}$$

   where $\tilde{\varepsilon} > 0$.

[1] Ambartsoumian and Jeske (2015) proposed a two-stage approach where a set of in-control observations is first accumulated through McDonald's (1990) sequential rank CUSUM (SRC). After a sufficient number of in-control observations have been accumulated a switch was made to one of the authors' proposed nonparametric CUSUM procedures. The procedures discussed in the paper are the nonparametric density estimation based CUSUM (NDEC) and probability integral transformation based CUSUM (PITC).

[1] Ambartsoumian and Jeske (2015) were concerned with detecting shifts in a process mean. To illustrate this, consider an in-control set of historical observations, $Y_1, Y_2, Y_3 \ldots, Y_N$. This set is

traditionally obtained during a Phase I (or retrospective) analysis, during which OOC observations (or subgroups) are removed. The process by nature is iterative and the IC reference sample gets recalibrated until there are only IC observations remaining. [26] McDonald's (1990) SRC is used to obtain $Y_1, Y_2, \ldots, Y_N$. To illustrate this procedure, consider a sequence of observations $X_1, X_2, \ldots$, the sequential rank of X_i is defined as

$$R_i = 1 + \sum_{j=1}^{i-1} (X_i - X_j)_+ \text{ for } i = 2, 3, 4, \ldots, \tag{2.40}$$

where

$$x_+ = \begin{cases} 1 & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

and

$$R_1 = 1.$$

The SRC is of the form

$$\begin{aligned} T_0 &= 0 \\ T_i &= max\left\{0, T_i + R_i/(i+1) - k\right\}, \; i = 1, 2, \ldots \end{aligned} \tag{2.41}$$

where k is a reference parameter fixed in advance. The CUSUM signals as soon as $T_i \geq H$, where H is the upper one-sided control limit. When the process is IC, the process is assumed to follow a distribution with probability density function $f_0(x)$ and distribution function $F_0(x)$. When the sequence is generated by the IC distribution, $F_0(x)$, the quantities $R_i(i+1)$ are independent and discrete uniform on $\left\{\frac{1}{(i+1)}, \frac{2}{(i+1)}, \ldots, \frac{i}{(i+1)}\right\}$. We set $k = \frac{1}{2}$, since it has the intuitive appeal of being the expected value of R_i when the process is IC. H is chosen using the following algorithm:

Algorithm 2.5 Determining the threshold for H in the SRC

1. Select a candidate value for H.
2. Simulate discrete uniform variables U_i from $\{1/(i+1), 2/(i+1), \ldots, i/(i+1)\}$ and substitute them for $R_i/(i+1)$ in Equation (2.41). Record the run-length needed before the SRC exceeds H.
3. Repeat step 2 B times ([1] Ambartsoumian and Jeske (2015) used $B = 10\,000$) and compute the ARL.
4. If the ARL from step 3 is sufficiently close to the target ARL_0, then stop, otherwise repeat steps 2 to 4.

We use McDonald's (1990) SRC with $k = \frac{1}{2}$ and H obtained by using Algorithm (2.5), to obtain a sufficiently large reference sample, $Y_1, Y_2, \ldots, Y_N$. This sample is then used to construct an Adaptive Kernel Density Estimate (AKDE). Given $Y_1, Y_2, \ldots, Y_N \sim f_0(x)$, the adaptive kernel density estimate

of f_0 at x is

$$\hat{f}_0(x) = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{\hat{h}_j}\phi\left(\frac{x - Y_j}{\hat{h}_j}\right), \tag{2.42}$$

with $\hat{h}_j = \hat{h}\hat{\lambda}_j$, where $\hat{\lambda}_j$ are adaptive factors that change with each value Y_j, $\hat{h}$ is a fixed bandwidth calculated by one of the standard methods and $\phi(.)$ is the standard normal density function. The adaptive factors are defined as

$$\hat{\lambda}_j = \left(\frac{\hat{g}}{\hat{f}(Y_j)}\right)^{\alpha},$$

where $\hat{f}$ is an initial pilot estimate of f_0 found by any available estimation method,

$$\hat{g} = \prod_{\tau=1}^{N}\left(\hat{f}\left(Y_\tau\right)\right)^{\frac{1}{N}}$$

and $\alpha \in [0, 1]$ is a constant. [1] Ambartsoumian and Jeske (2015) used a kernel density estimate with kernel function $\phi(.)$ and a fixed bandwidth to derive the pilot estimate of f_0. We set $\hat{h} = \hat{A}\left(4/\left(3N\right)\right)^{1/5}$ with $\hat{A} = min\left\{\hat{\sigma}, I\hat{Q}R/1.34\right\}$ where both $\hat{\sigma}$ and $I\hat{Q}R$ are estimated from the data. [37] Silverman (1986) recommends setting $\alpha = 0.5$.

Nonparametric density estimate CUSUM

The starting point for the NDEC is the following recursive form of [31] Page's (1954) optimal CUSUM

$$S_0 = 0,\ S_i = max\left\{0, S_{i-1} + \ln\frac{f_1(X_i)}{f_0(X_i)}\right\}. \tag{2.43}$$

The CUSUM signals a change when S_i crosses a corresponding control limit H. When the process is IC, it is assumed that the monitoring observations $X_1, X_2, \ldots$ have PDF $f_0(x)$ and CDF $F_0(x)$, as mentioned earlier. When the process is OOC, however, we assume that the observations have PDF $f_1(x)$ and CDF $F_1(x)$. It is assumed that $f_1(x)$ is a shift transformation of $f_0(x)$, i.e. $f_1(x) = f_0(x-K)$. Without loss of generality, we can consider a one-sided CUSUM because the two-sided CUSUM can be implemented, by running two one-sided CUSUMs (see Chapter 1.2.2).

The idea with the NDEC is to employ a properly chosen nonparametric density estimation (NDE) method to estimate the true, unknown IC density, f_0 in Equation (2.43), based on the historical IC data, i.e. $Y_1, Y_2, \ldots, Y_N$. If we denote the nonparametric density estimates of f_0 and f_1 by $\hat{f}_0$ and $\hat{f}_1$ respectively, the proposed nonparametric extension of Equation (2.43) is given by

$$\hat{S}_0 = 0,\ \hat{S}_1 = max\left\{0, \hat{S}_{i=1} + \ln\frac{\hat{f}_1(X_i)}{\hat{f}_0(X_i)}\right\}. \tag{2.44}$$

Note that $\hat{f}_1(x) = \hat{f}_0(x - K)$, since we assumed that f_1 is a shift transformation of f_0. The challenge of implementing the NDEC is to find a suitable value for H. [1] Ambartsoumian and Jeske (2015) used the following algorithm:

Algorithm 2.6 Determining H in the NDEC

1. Use $\{Y_i\}_{i=1}^{N}$ to obtain the AKDE of the in-control density, $\hat{f}_0$, and correspondingly, $\hat{f}_1$.
2. Select a candidate value for H.
3. Simulate (using a smoothed bootstrap) observations from $\hat{f}_0$and input them into the NDEC in Equation (2.44). Record the run-length needed before the NDEC exceeds H.
4. Repeat step 3 B times and compute the ARL.
5. If the ARL from step 4 is sufficiently close to the target ARL_0, then stop, otherwise repeat steps 2 to 5.

The smoothed bootstrap mentioned in step 3 of Algorithm 2.6 is performed using the following algorithm:

Algorithm 2.7 Smoothed bootstrap using AKDE

1. Calculate $\hat{h}_j$, for $j = 1, 2, 3, \ldots, N$, in Equation (2.42) using $Y_1, Y_2, \ldots, Y_N$.
2. For $i = 1, 2, 3, \ldots$, let r_i to be a random integer from $\{1, 2, \ldots, N\}$. Set $X_i = Y_{r_i} + \left(\hat{h}\hat{\lambda}_{r_i}\right)\epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, 1)$.

Probability integral transformation CUSUM (PITC)

When the process is IC, the transformed incoming observations $U_i = F_0(X_i)$ are distributed uniformly on the unit interval, i.e. $U(0, 1)$. [1] Ambartsoumian and Jeske (2015) hypothesised that the OOC distribution of U_i could be approximated as a beta distribution, $B(a, b)$, for suitably chosen values of a and b. The values for a and b are obtained by using the first and second moments, given by m_1 and m_2 respectively, of the OOC process distribution, which is assumed to be $B(a, b)$. When the change is a shift by a constant K, we have

$$\begin{aligned} m_1 &= \int_0^1 \left[1 - F_0\left(F_0^{-1}(u) - K\right)\right] du, \\ m_2 &= \int_0^1 2u\left[1 - F_0\left(F_0^{-1}(u) - K\right)\right] du. \end{aligned} \tag{2.45}$$

The solutions for a and b, in terms of m_1 and m_2 are given by

$$a = \frac{m_1^2 - m_1 m_2}{m_2 - m_1^2} \tag{2.46}$$

and

$$b = \frac{(m_1 - m_2)(1 - m_1)}{m_2 - m_1^2}, \tag{2.47}$$

respectively. If we let g_1 denote the 'best fit' beta distribution describing the OOC behaviour of the transformed data and the corresponding IC uniform density by g_0, then the optimal CUSUM increment based on the transformed data is

$$\begin{aligned} \log\left[g_1\left(U_i\right)/g_0\left(U_i\right)\right] &= \log\left[U_i^{a-1}\left(1-U_i\right)^{b-1}/B(a,b)\right] \\ &= (a-1)\log\left(U_i\right) + (b-1)\log\left(1-U_i\right) - \log B(a,b) \\ &= (a-1)\log\left[F_0(X_1)\right] + (b-1)\log\left[1-F_0(X_i)\right] - \log B(a,b). \end{aligned} \tag{2.48}$$

A smoothed estimate of the increment in Equation (2.48) was derived by approximating the unknown CDF F_0 with a kernel estimate $\hat{F}_0$, obtained from the historical IC data. Assuming that $F_0(x)$ has a corresponding density $f_0(x)$, the kernel estimate of $F_0(x)$ is taken to be the following statistic

$$\hat{F}_0(x) = \int_{-\infty}^{x} \hat{f}_0(t)dt,$$

where $\hat{f}_0(x)$ is the AKDE of the density function. Values for m_1 and m_2 (hence a and b) can be calculated numerically by replacing F_0 in Equation (2.48) with $\hat{F}_0(x)$. The proposed PITC tracking statistic is then

$$\begin{aligned} &\hat{S}_0 = 0, \\ &\hat{S}_i = \max\left[0, \hat{S}_{i-1} + (\hat{a}-1)\log\left[\hat{F}_0\left(X_i\right)\right] + \left(\hat{b}-1\right)\log\left[1-\hat{F}_0(X_i)\right] - \log B(\hat{a},\hat{b})\right]. \end{aligned} \tag{2.49}$$

As in the NDEC, the control limit H is determined through Monte Carlo simulation using the result that $\hat{F}_0(X_i) \overset{a}{\sim} U(0,1)$. H clearly depends on the historical IC data, this time trough $\hat{a}$ and $\hat{b}$. [1] Ambartsoumian and Jeske (2015) used the following algorithm:

Algorithm 2.8 Determining threshold H in the PITC

1. Use $\{Y_i\}_{i=1}^{N}$ to estimate the IC CDF $\hat{F}_0(x)$ and calculate $\hat{a}$ and $\hat{b}$ using Equations (2.45), (2.46) and (2.47), respectively.
2. Select a candidate value for H.
3. Simulate a U_i from $U(0,1)$ distribution and substitute it for $\hat{F}_0(x)$ in PITC plotting statistic given in Equation (2.49). Record the number of observations (i.e. the run-length) needed before PITC exceeds H.
4. Repeat step 3 B_1 times ([1] Ambartsoumian and Jeske (2015) used $B_1 = 10\ 000$) and compute the ARL.
5. If the ARL in step 4 is sufficiently close to the target ARL_0, then stop. Otherwise, repeat steps 2 to 5.

[1] Ambartsoumian and Jeske (2015) never discussed how they calculated $\hat{F}_0(x)$, but based on Equations (2.45), (2.46) and (2.47) it is strongly implied that this is done using numerical methods. Since

$$\hat{F}_0(x) = \int_{-\infty}^{x} \hat{f}_0(t)dt,$$

where

$$\hat{f}_0(t) = \frac{1}{N}\sum_{j=1}^{N} \frac{1}{\hat{h}_j}\phi\left(\frac{t-Y_j}{\hat{h}_j}\right).$$

It follows that

$$\begin{aligned}\hat{F}_0(x) &= \int_{-\infty}^{x}\left[\frac{1}{N}\sum_{j=1}^{N}\frac{1}{\hat{h}_j}\phi\left(\frac{t-Y_j}{\hat{h}_j}\right)\right]dt \\ &= \frac{1}{N}\sum_{j=1}^{N}\left[\int_{-\infty}^{x}\left(\frac{1}{\hat{h}_j}\phi\left(\frac{t-Y_j}{\hat{h}_j}\right)\right)dt\right].\end{aligned}$$

If we let $u_j = \left[(t-Y_j)/\hat{h}_j\right]$, i.e. $t = u_j\hat{h}_j + Y_j$, then it follows that

$$\begin{aligned}\int_{-\infty}^{x}\left(\frac{1}{\hat{h}_j}\phi\left(\frac{t-Y_j}{\hat{h}_j}\right)\right)dt &= \int_{-\infty}^{[(x-Y_j)/\hat{h}_j]}\left(\frac{1}{\hat{h}_j}\phi\left(u_j\right)\hat{h}_j\right)du_j \\ &= \int_{-\infty}^{[(x-Y_j)/\hat{h}_j]}\phi(u_j)du_j \\ &= \Phi\left(\frac{x-Y_j}{\hat{h}_j}\right) - \Phi(-\infty) \\ &= \Phi\left(\frac{x-Y_j}{\hat{h}_j}\right).\end{aligned}$$

Using the above result, we can calculate $\hat{F}_0(x)$ using the formula

$$\hat{F}_0(x) = \frac{1}{N}\sum_{j=1}^{N} \Phi\left(\frac{x - Y_j}{\hat{h}_j}\right), \tag{2.50}$$

where $\hat{h}_j$ is the same adaptive smoothing parameter used in the AKDE and $\Phi(.)$ is the CDF for a normal distribution with mean equal to 0 and variance equal to 1. We can define the inverse of $\hat{F}_0$ as

$$\hat{F}_0^{-1}(u) = \inf\left\{x \in \mathbb{R} \mid \hat{F}_0(x) \geq u\right\}, \tag{2.51}$$

where $0 < u < 1$. We can also determine $\hat{F}_0^{-1}(u)$ numerically, using the fact that $\hat{F}_0^{-1}(u) = x$ is equivalent to $\hat{F}_0(x) = u$ for some $x \in \mathbb{R}$ and $0 < u < 1$. This can be done using Algorithm (2.9).

Algorithm 2.9 Secant method to solve $\hat{F}_0^{-1}(u) = x$

1. Specify a tolerance level TOL, where $TOL > 0$. Take two initial guesses for x, say x_1 and x_2, such that $\hat{F}_0(x_1) < u$ and $\hat{F}_0(x_2) > u$.
2. Calculate
$$x_{n+2} = x_{n+1} - \frac{x_{n+1} - x_n}{\hat{F}_0(x_{n+1}) - \hat{F}_0(x_n)}\left(\hat{F}_0(x_{n+1}) - u\right) \text{ for } n = 1, 2, \ldots.$$
3. Repeat step 2 until $\left|\hat{F}_0(x_{n+2}) - u\right| \leq TOL$ for some $n = 1, 2, \ldots$.

Using Equation (2.50) and Algorithm (2.9) one can obtain estimates for m_1 and m_2 by numerically integrating the functions

$$\begin{aligned} \hat{m}_1 &= \int_0^1 \left[1 - \hat{F}_0\left(\hat{F}_0^{-1}(u) - K\right)\right] du, \\ \hat{m}_2 &= \int_0^1 2u\left[1 - \hat{F}_0\left(\hat{F}_0^{-1}(u) - K\right)\right] du. \end{aligned}$$

It follows that the estimates for a and b can be calculated by substituting $\hat{m}_1$ and $\hat{m}_2$ in Equations (2.46) and (2.47), respectively.

2.3.2 Performance overview of bootstrap CUSUM charts

[7] Chatterjee and Qiu (2009) tested the performance of their bootstrap CUSUM chart by performing a simulation study. The following cases were considered:

1. A standard normal distribution, i.e. $\mathcal{N}(0, 1)$, when the process is IC and $\mathcal{N}(\delta, 1)$ distribution when the process is OOC, where $\delta > 0$.
2. The IC process distribution has the density function

$$f(x) = \begin{cases} \frac{1}{6}e^{-x/3} & , \text{ when } x \geq 0 \\ \frac{1}{2}e^{x} & , \text{ when } x < 0. \end{cases}$$

Note than any random variable X from the distribution with PDF $f(.)$ is standardised to have a mean of 0 and variance of 1, i.e. $Y = T(X)$, such that $\mathbb{E}(Y) = 0$ and $Var(Y) = 1$. A random variable from OOC process distribution is just a location shift of $f(.)$, i.e. $Z = Y - \delta$ for $\delta > 0$. In this case, the distribution is skewed to the right.

3. The IC process distribution has the density function

$$h(x) = \begin{cases} \frac{1}{2}e^{-x} & , \text{ when } x \geq 0 \\ \frac{1}{6}e^{x/3} & , \text{ when } x < 0. \end{cases}$$

Note than any random variable X from the distribution with PDF $h(.)$ is standardised to have a mean of 0 and variance of 1, i.e. $Y = T(X)$, such that $\mathbb{E}(Y) = 0$ and $Var(Y) = 1$. A random variable from OOC process distribution is just a location shift of $h(.)$, i.e. $Z = Y - \delta$ for $\delta > 0$.

[7] Chatterjee and Qiu (2009) found that their bootstrap control chart "performs reasonably well, provided that j_{max} is not too small." In general, it was found that when $j_{max} \geq 30$, then the bootstrap chart proposed by [7] Chatterjee and Qiu (2009) performs reasonably well. [7] Chatterjee and Qiu (2009) stated that:

"In this paper we suggest choosing j_{max} from computational considerations, and $\mathbb{E}[T_n]$ by linking it to j_{max}, as a matter of convenience. The distribution of (C_n, T_n) depends on (j_{max}, k) in a way that is poorly understood at present."

[7] Chatterjee and Qiu (2009) observed that the best choice for j_{max} may depend on both the IC and OOC process distributions. It was found that the general rule of $j_{max} \geq 30$ is sufficient for most circumstances. The method proposed by [7] Chatterjee and Qiu (2009) may prove cumbersome to implement for the novice practitioner, due to the computational complexity of determing the best value for j_{max} and k.

[1] Ambartsoumian and Jeske (2015) benchmarked the performance of their proposed NDEC and PITC charts by simulating from the following distributions:

1. The standard normal, i.e. $\mathcal{N}(0, 1)$,
2. the t-distribution with 3 degrees of freedom, i.e. $t(3)$ and
3. the Laplace distribution with location parameter 0 and scale parameter 1, i.e. $L(0, 1)$.

The NDEC and PITC control charts were constructed using Algorithms 2.6 and 2.8. [1] Ambartsoumian and Jeske (2015) also considered the Transformed CUSUM (TC) control chart by [16] Jeske, Montes De Oca, Bischoff and Marvasti (2009). The TC tracking statistic was defined as

$$\hat{S}_i = \max\left\{0, \hat{S}_{i-1} + \hat{F}_0(X_i) - \alpha\right\}, \text{ for } i = 1, 2, 3, \ldots, \tag{2.52}$$

where $\hat{S}_0 = 0$ and $\hat{F}_0(.)$ is the usual empirical density function (EDF). [1] Ambartsoumian and Jeske

(2015) set $\alpha = 0.5$ in Equation (2.52). The procedure used by [1] Ambartsoumian and Jeske (2015) to determine the control limit for the TC chart is listed in Algorithm 2.10.

Algorithm 2.10 Determing the threshold H in TC

1. Select a candidate value for H.
2. Simulate discrete uniform random variable $U_1, U_2, U_3, \ldots$ from $\{0, 1/N, 2/N, \ldots, (N-1)/N, 1\}$ and substitute them for $\hat{F}_0(x)$ in Equation (2.52). Record the run-length needed before the TC plotting statistic exceeds H.
3. Repeat step 2 a large number of times, say B_1 ([1] Ambartsoumian and Jeske (2015) used $B_1 = 10\ 000$) and compute the ARL.
4. If the ARL in step 3 is sufficiently close to the targe ARL_0, then stop. Otherwise, repeat steps 1 to 4.

[1] Ambartsoumian and Jeske (2015) showed that, at a minimum, a reference dataset of size $N \geq 13\ 000$ is required to consistently achieve an IC ARL close to the nominal ARL_0 value, for the NDEC chart. A change point model was used to evaluate the performance of the control charts. Typically, in a change point model environment, it is assumed that random observations follow the IC process distribution with CDF $F_0(x)$, before a certain point in time, say $\tau \in \mathbb{N}$, i.e. $X_i \sim F(x)$ for $i = 1, 2, \ldots, (\tau - 1)$. From the τth observation onwards, it is assumed that random observations follow the OOC process distribution with CDF $F_1(x)$, i.e. $X_i \sim F_1(x)$ for $i = \tau, \tau + 1, \tau + 2, \ldots$.

[1] Ambartsoumian and Jeske (2015) introduced a shift in the process mean at the 2 000th observation, this was done so that the CUSUM charts could reach their so-called steady-states, hence the run-lengths were recorded after the insertion of the shift at the 2 000th observation. It should also be noted that [1] Ambartsoumian and Jeske (2015) calibrated their control charts to be optimised for detecting shifts in magnitude of 0.5 standard deviations ($\Delta = 0.5$). To mitigate the effect of the computational load that the AKDE carries, [1] Ambartsoumian and Jeske (2015) used the *exact* (or known) PDF and CDF functions for the respective distribution, this referred to as the $N = \infty$ assumption. The rationale behind the $N = \infty$ assumption is that the AKDE, $\hat{f}_0(x)$, should approach the *real* (or true) PDF of the IC process distribution, $f_0(x)$, as $N \to \infty$.

Under the $N = \infty$ assumption, it was found that the NDEC and PITC perform at a similar level when there is a shift in the process mean in magnitude of $\delta \in \{0, 0.1, 0.25, 0.5, 0.75, 1\}$ (see [1] Ambartsoumian and Jeske (2015), p273). [1] Ambartsoumian and Jeske (2015) also showed that their PITC and NDEC chart out-performed the benchmark TC control chart, for the $\mathcal{N}(0, 1)$, $t(3)$ and $L(0, 1)$ cases, respectively.

2.3.3 Conclusion

The bootstrap control chart proposed by [7] Chatterjee and Qiu (2009) relies on using a sequence of control limits $\{h_j\}_{i=1}^{j_{max}}$, which depend on the conditional distribution of $\{C_n | T_n = j\}$, obtained via the bootstrap. For any given scenario, implementing Algorithms 2.2, 2.3 and 2.4, seems to yield decent results, provided that j_{max} is not too small or too large. The problem with the approach proposed by

[7] Chatterjee and Qiu (2009) is that no clear guidelines can be given on how some of the parameters should be chosen, since there is a lack of understanding of the distributional relationship between (C_n, T_n) and (j_{max}, k). This makes it difficult to recommend it to a practitioner.

The NDEC and PITC proposed by [1] Ambartsoumian and Jeske (2015) seems more robust and easier to implement. [1] Ambartsoumian and Jeske (2015) tested their chart(s) by simulating from symmetric distributions, whereas [7] Chatterjee and Qiu. (2009) used both symmetric- and skew distributions. A problem with the approach proposed by [1] Ambartsoumian and Jeske (2015) is the extreme computation time required to construct the NDEC. Since [1] Ambartsoumian and Jeske (2015) showed that the PITC is equivalent to the NDEC in terms of performance, a PITC chart was constructed for a process with an IC $Exp(1)$ process distribution. [1] Ambartsoumian and Jeske (2015) recommended using an IC reference dataset of size $N \geq 13\,000$. A reference dataset of size $N = 15\,000$ was used, the PITC and TC charts were calibrated to have nominal $ARL_0 = 2\,000$. A shift in magnitude of δ standard deviations was inserted at the 2 000th observation. The SAS v9.4 source code is given in Section A.1.4 of this mini-dissertation.

The results for the simulation study are given in Table 2.5.

Table 2.5: ARL when IC distribution is $Exp(1)$, $N = \infty$. Target shift inserted at the 2 000th observation, $\Delta = 0.5$

δ	PITC	$TC_{0.5}$
0.00	1997.2909	1702.4304
0.10	115.3836	180.6027
0.25	25.3909	78.5742
0.50	13.3509	44.1701
0.75	9.8974	32.9701
1.00	8.3392	27.3905

The TC chart has a lower IC ARL than the PITC chart, despite finding the control limit for the TC chart as suggested by Algorithm 2.10. Since the two charts have significantly different IC ARL's it would be unfair to compare them directly. It can be noted from Table 2.5 that, despite having a lower IC ARL than the PITC chart, the TC chart performs much worse when there is a shift in the process mean, than the PITC.

The NDEC and PITC charts proposed by [1] Ambartsoumian and Jeske (2015) are fairly easy to implement and provide consistent results, provided that the IC reference dataset is sufficiently large. The two-stage approach, where the Phase I analysis is completely skipped and replaced by the online process monitoring procedure known as the SRC, ensures that the necessary amount of data will be acquired at some stage to implement the NDEC and PITC. A disadvantage of the NDEC is the speed at which the process is monitored, since it is computationally intensive to obtain estimates for $\hat{f}_1(x)$ and $\hat{f}_0(x)$ using AKDE. The PITC has the complication of having to estimate a and b using numerical integration.

The NDEC and PITC can easily be modified for monitoring any process parameter using a statistic, say $T(.)$, i.e. the $Y_1, Y_2, \ldots, Y_N$ values can be replaced with $T_1, T_2, \ldots, T_N$ where

$$T_i = T(X_{i1}, X_{i2}, \ldots, X_{in}) \text{ for } i = 1, 2, \ldots, N,$$

$$X_{ij} \sim \text{i.i.d. } F_0(x) \text{ for } j = 1, 2, \ldots n,$$

and $n \geq 1$ is the subgroup size. The IC process sampling distribution for $T(.)$ can be approximated by $\hat{f}_{T,0}(t)$, where $\hat{f}_{T,0}(t)$ is the AKDE of $T(.)$.

2.4 EWMA control chart

[33] Saleh et al. (2015) showed that larger values for the EWMA smoothing constant (λ) resulted in higher levels of variability in the IC *ARL* distribution and concluded that more Phase I data is required for charts with larger smoothing constants. [33] Saleh et al. (2015) considered different estimators for the IC process standard deviation, to compensate for *practitioner-to-practitioner* variability. [33] Saleh et al. (2015) used a bootstrap approach proposed by [13] Gandy and Kvaløy (2013) to determine the control limits for the EWMA control chart. [33] Saleh et al. (2015) summarised the steps of the procedure proposed by [13] Gandy and Kvaløy (2013) as follows:

1. Without loss of generality, it can be assumed that the IC process distribution is $\mathcal{N}(0, \sqrt{n})$. A Phase I dataset of m samples each of size n is generated from a $\mathcal{N}(0, \sqrt{n})$ distribution.

2. The IC process mean and standard deviation are estimated from the Phase I data, generated in step 1, i.e. $\hat{\mu}_0$ and $\hat{\sigma}_0$.

3. Compute the quantity $q(\hat{P}, \hat{\theta})$, where $\hat{P} = \mathcal{N}(\hat{\mu}_0, \hat{\sigma}_0)$ and $\hat{\theta} = (\hat{\mu}_0, \hat{\sigma}_0)$, respectively. The quantity $q(\hat{P}, \hat{\theta})$ is the value of L that produces the desired IC *ARL* when the Phase II data are generated from a $\mathcal{N}(\hat{\mu}_0, \hat{\sigma}_0)$ distribution.

4. Generate a bootstrap dataset consisting m samples each size n from a $\mathcal{N}(\hat{\mu}_0, \hat{\sigma}_0)$ distribution. The mean and standard deviation are calculated for the bootstrap dataset as in step 2, i.e. $\hat{\mu}_0^*$ and $\hat{\sigma}_0^*$, respectively. We denote the set of bootstrap parameters as $\hat{\theta}^* = (\hat{\mu}_0^*, \hat{\sigma}_0^*)$.

5. Compute the quantities $q\left(\hat{P}^*, \hat{\theta}^*\right)$ and $q\left(\hat{P}, \hat{\theta}^*\right)$, respectively, where $q\left(\hat{P}^*, \hat{\theta}^*\right)$ is the value of L that produces the desired IC *ARL* when the Phase II data are generated from a $\mathcal{N}(\hat{\mu}_0^*, \hat{\sigma}_0^*)$ distribution and $q\left(\hat{P}, \hat{\theta}^*\right)$ is the value of L that produces the desired IC *ARL* when the Phase II data are generated from a $\mathcal{N}(\hat{\mu}_0, \hat{\sigma}_0)$ distribution.

6. Repeat steps 4 and 5 a large number of times, say $B \geq 1\,000$, and compute the quantities $q\left(\hat{P}_i^*, \hat{\theta}_i^*\right)$ and $q\left(\hat{P}, \hat{\theta}_i^*\right)$, respectively, for $i = 1, 2, \ldots, B$.

7. Calculate the value p_α^* as the $\alpha \times 100$th percentile of the bootstrap distribution of $\left(q\left(\hat{P}^*, \hat{\theta}^*\right) - q\left(\hat{P}, \hat{\theta}^*\right)\right)$.

8. The final (adjusted) control limit for the chart is then taken as $q\left(\hat{P}, \hat{\theta}\right) - p_\alpha^*$.

[33] Saleh et al. (2015) warned that this procedure can take a long time to produce results, since the value of L has to determined B different sets of data using a trial and error approach. In this mini-dissertation a nonparametric bootstrap procedure is proposed for the EWMA control chart.

2.5 Other control charts

[32] Phaladiganon et al. (2011) proposed a bootstrap-based multivariate T^2 control chart. Suppose that we have an IC reference sample that contains n observations and each observation is characterised by $p \geq 1$ process variables. If we assume that the data follows a multivariate normal distribution with an unknown mean vector $\boldsymbol{\mu} : (p \times 1)$ and a covariance matrix $\boldsymbol{\Sigma} : (p \times p)$, then Hotelling's T^2 statistic calculated by

$$T^2 = (\boldsymbol{x} - \bar{\boldsymbol{x}})^T \boldsymbol{S}^{-1} (\boldsymbol{x} - \bar{\boldsymbol{x}}), \tag{2.53}$$

where

$$\bar{\boldsymbol{x}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i,$$

and

$$\boldsymbol{S}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$$

follows an F distribution with p and $(n-p)$ degrees of freedom under the normality assumption. Under this assumption, the control limit for T^2 can be determined by using the $100(1-\alpha)$-th percentile of the F distribution. Since this control limit is only accurate, when assuming T^2 follows an F distribution, [8] Chou and Mason (2001) proposed a nonparametric approach that uses KDE to estimate the distribution of the T^2 statistics. The control limit can be determined by the $100(1-\alpha)$-th percentile of the estimated kernel distribution. One major drawback of this approach is that it is computationally intensive, since it requires numerical integration of the KDE. [32] Phaladiganon et al. (2011) suggested resampling from a set of $n \geq 1$ T^2 statistics computed from an in-control dataset using Equation (2.53). The approach followed by Phaladiganon et al. (2011) is summarised as follows.

Algorithm 2.11 The bootstrap procedure to calculate control limits for T^2 control chart

1. Compute the T^2 statistics with the $n \geq 1$ observations from an in-control dataset, i.e. $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, using
$$T_i^2 = (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \boldsymbol{S}^{-1} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}) \text{ for } i = 1, 2, \ldots n.$$
2. Let $T_1^{2(i)}, T_2^{2(i)}, \ldots, T_n^{2(i)}$ be a set of n T^2 from the i-th bootstrap sample ($i = 1, 2, \ldots, B$) randomly drawn from $\left\{T_i^2\right\}_{i=1}^n$ with replacement. In general, B is a large number, say $B \geq 1\ 000$.
3. For $i = 1, 2, \ldots, B$ determine the $(1 - \alpha) \times 100$th percentile value, $T_{(100(1-\alpha))}^{2(i)}$, for a given user-specified α.
4. Determine the control limit by taking an average of the B $100(1 - \alpha)$-th percentile values, i.e.
$$\bar{T}_{100(1-\alpha)}^2 = \frac{1}{B} \sum_{i=1}^{B} T_{(100(1-\alpha))}^{2(i)}.$$
5. Use the established control limit to monitor a new observation. If the monitoring statistic of a new observation exceeds $\bar{T}_{100(1-\alpha)}^2$ then the process is declared out-of-control.

[32] Phaladiganon et al. (2011) concluded that with normally distributed data, the conventional, the KDE and bootstrap approaches all performed similarly. It was also concluded that the bootstrap chart's performances in normal and nonnormal situations were similar to the KDE approach by [8] Chou and Mason (2001).

Chapter 3

Proposed design for EWMA bootstrap control chart

3.1 Introduction

In this chapter a different approach to constructing a bootstrap-based EWMA control chart is shown. Potential estimators for the process mean and variance, based on the statistic used to monitor the process, are shown. The estimates for the process mean and variance, respectively, are calculated using the bootstrap.

Two illustrative examples are provided, showing the implementation of the bootstrap EWMA chart for monitoring the process mean and median, respectively, of the $Exp(1)$ distribution. The width of the EWMA control limits is determined using a secant method and the bootstrap.

3.2 Design

Suppose that a sufficiently large in-control reference sample consisting of $m \geq 1$ samples (or subgroups) of size $n \geq 1$ i.i.d. random variables was accumulated, i.e.

$$\chi = (X_{11}, X_{12}, \ldots, X_{1n}, X_{21}, X_{22}, \ldots, X_{2n}, \ldots, X_{m1}, X_{m2}, \ldots, X_{mn}) .$$

Without loss of generality, it can be assumed that this in-control reference sample was obtained by performing either a Phase I analysis (where out of control subgroups were iteratively removed using the process described by [27] Montgomery (2009)) or a self-starting procedure like [26] McDonald's (1990) SRC.

The parameter of interest for the process is IC monitored by calculating a statistic

$$\psi_i = T(X_{i1}, X_{i2}, \ldots, X_{in})$$

based on a sample of size n collected at a specific sampling point $i = 1, 2, 3, \ldots$. The charting statistic for the classic (or Vanilla) EWMA control chart is defined as

$$Z_i = \lambda\psi_i + (1-\lambda)Z_{i-1} \text{ for } i = 1, 2, 3, \ldots \tag{3.1}$$

where $\lambda \in (0, 1]$ is a constant called the smoothing parameter and ψ_i is the statistic used to monitor the process. The starting value for Z_i is typically taken to be the target value, i.e. $Z_0 = \theta_0$. Theoretically,

$$E\left[\psi_i|\text{IC}\right] = \theta_0$$

and

$$Var\left[\psi_i|\text{IC}\right] = \sigma_0^2.$$

In the case where both θ_0 and σ_0 are unknown, estimates for these parameters need to be obtained from the Phase I data. The problem with this is that distributional assumptions need to be made about the process distribution. The theoretical framework for most control charts assumes that the statistic, used to monitor the process, is normally distributed with in-control mean θ_0 and variance σ_0^2. Within this framework good estimators for the unknown parameters can easily be found. Even when the process is not normally distributed, if one assumes some known distribution, closed-form expressions for the estimators can still be found which leads to a myriad of possible estimators. We propose a different, if somewhat naive, approach where one uses the bootstrap to estimate θ_0 and σ_0. Before we can do this, we need to assume that the X_{ij}'s in the in-control reference sample are i.i.d. with CDF given by $F_0(x)$ and the OOC process distribution has CDF $F_1(x)$.

The empirical distribution function (EDF) for the in-control distribution is given by

$$\hat{F}_0(x) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{I}\left(X_{ij} \leq x\right), \tag{3.2}$$

where $\mathbf{I}(A)$ equals 1 when A is true and 0 when A is false. It can be shown that

$$E\left[\hat{F}_0(x)\right] = F_0(x)$$

and

$$\lim_{N\to\infty} Var\left[\hat{F}_0(x)\right] = 0,$$

where $N = mn$, which implies that $\hat{F}_0(x)$ is an unbiased and consistent estimator for $F_0(x)$. One can

emulate the IC process distribution by sampling from the EDF, i.e. sample with replacement from the X_{ij}'s. If we draw a sample of size $n \geq 1$ from the reference sample (by sampling with replacement), say $X_1^*, X_2^*, \ldots, X_n^*$, we can calculate the statistic $\psi^* = T(X_1^*, \ldots, X_n^*)$ using the bootstrap sample. If we were to draw all possible permutations of bootstrap samples and calculate ψ^* for each of them, we would obtain the sampling distribution for ψ^*. If we had a reference sample of size mn and we drew bootstrap samples of size n from the available mn elements, there would be $(mn)^n$ possible samples, which becomes computationally intensive very quickly, e.g. if $mn = 20$ and $n = 5$ then we would have $20^5 = 3\ 200\ 000$ possible bootstrap samples of size 5.

[9] Efron (1979) showed that the bootstrap sampling distribution can be sufficiently replicated by performing a large number of Monte Carlo simulations (say B). We want to use these B bootstrap samples, which we generated by taking samples of size $n \geq 1$ from $\hat{F}_0$, to emulate the actual in-control sampling distribution function of ψ, given by F_ψ, by using the bootstrap sampling distribution of ψ^* given by F_ψ^*. It follows that

$$\lim_{B\to\infty} \hat{F}_\psi^*(x) = F_\psi^*(x),$$

where

$$\hat{F}_\psi^*(t) = \frac{1}{B}\sum_{i=1}^{B} \mathbf{1}\left(\psi_i^* \leq t\right).$$

Let

$$\mathbb{E}_{\hat{F}_0}\left[\psi^*\right] = \int_{-\infty}^{\infty} \psi^* d\hat{F}_0 = \theta_0^*$$

and

$$SE_{\hat{F}_0}\left[\psi^*\right] = \sigma_{\psi,0}^*,$$

where θ_0^* and σ_0^* are the mean and standard error, respectively, for the distribution of ψ^*. Let

$$\hat{\theta}_0^* = \frac{1}{B}\sum_{i=1}^{B} \psi_i^* \tag{3.3}$$

and

$$\hat{\sigma}_0^* = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}\left(\psi_i^* - \hat{\theta}_0^*\right)^2}. \tag{3.4}$$

Clearly $\mathbb{E}_{\hat{F}_0}\left[\hat{\theta}_0^*\right] = \theta_0^*$ and $\mathbb{E}_{\hat{F}_0}\left[\left(\hat{\theta}_0^* - \theta_0^*\right)^2\right] = \frac{\sigma_0^2}{B}$. From this it follows that

$$\lim_{B\to\infty} \frac{1}{B}\sum_{i=1}^{B} \psi_i^* = \theta_0^* \tag{3.5}$$

and

$$\lim_{B\to\infty} \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}\left(\psi_i^* - \hat{\theta}_0^*\right)^2} = \sigma_0^*. \tag{3.6}$$

The following algorithm can be used to obtain estimates for θ_0 and σ_0:

Algorithm 3.1 Bootstrap estimators for θ_0 and σ_0

1. Use $X_{11}, X_{12}, \ldots, X_{1n}, \ldots, X_{m1}, X_{m2}, \ldots, X_{mn}$ to construct the EDF $\hat{F}_0$ as defined by Equation (3.2).
2. Draw a random sample with replacement of size $n \geq 1$ from the $N = mn$ pooled in-control observations.
3. Compute $\psi_i^* = T\left(X_1^*, X_2^*, \ldots, X_n^*\right)$ from the bootstrap sample in step 2, for $i = 1, 2, \ldots, B$, where B is a large number, say $B = 10\ 000$.
4. Compute

$$\hat{\theta}_0^* = \frac{1}{B}\sum_{i=1}^{B}\psi_i^*$$

and

$$\hat{\sigma}_0^* = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}\left(\psi_i^* - \hat{\theta}_0^*\right)^2}.$$

From Equations 3.5 and 3.6, respectively, it was shown that $\hat{\theta}_0^* = \theta_0^*$ and $\hat{\sigma}_0^* = \sigma_0^*$ if $B \to \infty$, i.e. $\hat{\theta}_0^* \approx \theta_0^*$ and $\hat{\sigma}_0^* \approx \sigma_0^*$ when B is sufficiently large.

3.2.1 Naive EWMA bootstrap control chart

Consider the following version of the classic or vanilla EWMA charting statistic

$$Z_i^* = \lambda\psi_i^* + (1-\lambda)Z_{i-1}^*, \tag{3.7}$$

where $\psi_i^* = T\left(X_{i1}^*, X_{i2}^*, \ldots, X_{in}^*\right)$ for $i = 1, 2, 3, \ldots$, $X_{i1}^*, X_{i2}^*, \ldots, X_{in}^*$ is a resample from the in-control X_{ij}'s and $Z_0^* = \hat{\theta}_0^*$ (where $\hat{\theta}_0^*$ is defined in Algorithm 3.1). The control limits and CL for the Naive Bootstrap EWMA (NBEWMA) chart are determined by using the formulae

$$\begin{aligned} UCL_i^* &= \hat{\theta}_0^* + L\hat{\sigma}_0^* \sqrt{\left(\frac{\lambda}{2-\lambda}\right)(1-(1-\lambda)^{2i})} \\ CL_i^* &= \hat{\theta}_0 \\ LCL_i^* &= \hat{\theta}_0^* - L\hat{\sigma}_0^* \sqrt{\left(\frac{\lambda}{2-\lambda}\right)(1-(1-\lambda)^{2i})}, \end{aligned} \tag{3.8}$$

where L is chosen to achieve a nominal in-control average run-length ARL_0 and $\lambda \in (0, 1]$ is chosen to detect the magnitude of a shift in the process mean. The chart is set up using a two-stage approach, where the value for L is chosen using Algorithm 3.2 and the process is then subsequently monitored using the tuned value for L.

Algorithm 3.2 Choosing L for the NBEWMA chart

1. Calculate $\hat{\theta}_0^*$ and $\hat{\sigma}_0^*$ using Algorithm 3.1.
2. Select a value for L and calculate the control limits in Equation (3.8).
3. Simulate (by sampling with replacement) observations from $\hat{F}_0$ and input them in Equation (3.7). Record the run-length needed before $Z_i^* \geq UCL_i^*$ or $Z_i^* \leq LCL_i^*$.
4. Repeat step 3 B times (typically $B \geq 1\ 000$) and compute the ARL.
5. If the ARL from step 4 is sufficiently close to the target ARL_0 (i.e. within 5%), then stop, otherwise repeat steps 2 to 4.

3.2.2 Illustrative example 1

Suppose we have an in-control reference sample consisting of $m = 20$ subgroups of size $n = 5$ generated from an $Exp(1)$ distribution. The SAS v9.4 code used to produce the results in this example is provided in Section A.2.1 of the Appendix. Suppose that we want to monitor the process using the sample mean, i.e. $\psi = \bar{X}$, we want a nominal in-control average run-length of 500, i.e. $ARL_0 = 500$, and we want to detect small shifts ($\lambda = 0.05$). Using Algorithm 3.1 with $B = 100\ 000$ replications, we have

$$\hat{\mu}_0^* = 0.7965718$$

and

$$\hat{\sigma}_0^* = 0.4130238.$$

Since a large number of replications were used, i.e. $B = 100\ 000$, it can be inferred from Equations 3.5 and 3.6, respectively, that $\hat{\mu}_0^* \approx \mu_0^*$ and $\hat{\sigma}_0^* \approx \sigma_0^*$, respectively. As specified by Algorithm 3.2, it was initially guessed that $L = 2.5$. After running 1 000 iterations, an ARL of $340.376 < 500$ was obtained. Clearly, $L = 2.5$ produced intervals that were not wide enough, so it was decided to set $L = 2.7$ and 1 000 iterations were ran, again. This produced an ARL of $553.099 > 500$, which produced intervals that were too wide. Using a secant method, it was determined that $L = 2.6501$. The results used are

summarised in the Table 3.1.

Table 3.1: Searching for a suitable L-value using a secant method

L	ARL	Absolute % difference
2.5000	340.376	31.9248%
2.7000	553.099	10.6198%
2.6501	501.557	0.3114%

To employ the secant method used in Table 3.1, two initial guesses for L, given by L_1 and L_2, respectively and their corresponding ARL values, given by ARL_1 and ARL_2, respectively, are needed. Typically, $L_1 < L_2$ (and subsequently $ARL_1 < ARL_2$). The value for L was calculated iteratively by using

$$L_i = (ARL_0 - ARL_{i-1}) \frac{L_{i-1} - L_{i-2}}{ARL_{i-1} - ARL_{i-2}} + L_{i-1} \text{ for } i = 3, 4, 5, \ldots,$$

where ARL_0 is the specified in-control ARL (in our case $ARL_0 = 500$). The stopping criteria was determined by

$$\frac{|ARL_i - ARL_0|}{ARL_0} \leq TOL,$$

where TOL is a pre-specified tolerance (or desired level of precision). The TOL-value used to determine L was 2%.

3.2.3 Illustrative example 2

Again, we have an in-control reference sample consisting of $m = 20$ subgroups of size $n = 5$ generated from an $Exp(1)$ distribution. The SAS v9.4 code used to produce the results in this example is provided in Section A.2.2 of the Appendix. Suppose that the sample median is used to monitor the process, i.e. $\hat{\eta} = \tilde{X}$. To achieve nominal in-control average run-length of 500, i.e. $ARL_0 = 500$, and the detection small shifts ($\lambda = 0.05$) is of interest. Using Algorithm 3.1 with 100 000 replications, estimates for the process mean and standard error are given by

$$\hat{\eta}_0^* = 0.7976531$$

and

$$\hat{\sigma}_0^* = 0.5074285,$$

respectively. As specified by Algorithm 3.2, it was initially guessed that $L = 2.5$. After running 1 000

iterations, an ARL of $344.347 < 500$ was obtained. Clearly, $L = 2.5$ produced intervals that were not wide enough, it was decided to set $L = 2.7$ and run 1 000 iterations were ran again. This gave an ARL of $565.865 > 500$, which produced intervals that were too wide. Using a secant method, it was determined that $L = 2.656$. The results used are summarised in Table 3.2.

Table 3.2: Searching for a suitable L-value using a secant method

L	ARL	Absolute % Difference
2.5000	344.347	31.13%
2.7000	565.865	13.17%
2.641	491.117	1.78%
2.648	491.569	1.69%
2.779	646.085	29.22%
2.655	475.7	4.86%
2.673	550.117	10.02%
2.661	514.129	2.83%
2.656	493.808	1.24%

To employ the secant method used in Table 3.1, two initial guesses for L, given by L_1 and L_2, respectively and their corresponding ARL values, given by ARL_1 and ARL_2, respectively, are needed. Typically, $L_1 < L_2$ (and subsequently $ARL_1 < ARL_2$). The value for L was calculated iteratively by using

$$L_i = (ARL_0 - ARL_{i-1}) \frac{L_{i-1} - L_{i-2}}{ARL_{i-1} - ARL_{i-2}} + L_{i-1} \text{ for } i = 3, 4, 5, \ldots,$$

where ARL_0 is the specified in-control ARL (in our case $ARL_0 = 500$). The stopping criteria was determined by

$$\frac{|ARL_i - ARL_0|}{ARL_0} \leq TOL,$$

where TOL is a pre-specified tolerance (or desired level of precision). The TOL-value used to determine L was 2%.

Chapter 4

Performance of the proposed NBEWMA control chart

4.1 Introduction

In this chapter the IC and OOC performance of the proposed NBEWMA control chart (see Chapter 3) is determined for different underlying process distributions. The process mean is monitored using the sample mean, $\bar{x}$, in this chapter, since it is a commonly used location statistic in SPC literature.

The NBEWMA control chart is constructed using $m = 50$, 100 and 1000 samples, respectively, each of size $n = 1$, 5 and 10 respectively. The performance of the NBEWMA chart is compared to a conventional EWMA control chart (see [17] Jones, Champ and Rigdon (2001) and [33] Saleh, Mahmoud, Jones-Farmer, Zwetsloot and Woodall (2015)). This chapter is concluded with an overview of the performance of the NBEWMA chart, followed by recommendations for future research.

4.2 Simulation Study

[33] Saleh, Mahmoud, Jones-Farmer, Zwetsloot and Woodall (2015) made recommendations for the minimum number of subgroups required to construct Shewhart and EWMA control charts. It was shown by [33] Saleh et al. (2015) that the minimum number of subgroups required were typically more than recommended by [17] Jones, Champ and Rigdon (2001). [33] Saleh et al. (2015) used the standard deviation for the average run-length ($SDARL$) to measure whether the number of subgroups used are sufficient. It was argued by [33] Saleh et al. (2015) that the $SDARL$ is a measurement of the *practitioner-to-practitioner* variability. [33] Saleh et al. (2015) targeted $SDARL$ values that are approximately within 10% of the nominal ARL_0.

[33] Saleh et al. (2015) argued that about $m = 600$ samples of size $n = 5$ are needed in order to construct an EWMA-$\bar{X}$ control chart, if $\lambda = 0.1$, to obtain an $SDRL$ within 10% of the ARL_0. It is worth noting that [33] Saleh et al. (2015) assumed that the IC process distribution is normal, which is a common assumption in the literature of the EWMA control chart. It was shown by [33] Saleh et

al. (2015) that smaller values of λ ($\lambda < 0.25$) require fewer observations to construct reliable control charts; fewer than their Shewhart counterparts. Saleh et al. (2015) stated that: "In most applications, it would not be realistic to obtain this amount of stable Phase I data from the process."

In this mini-dissertation an alternative approach to constructing an EWMA control using a non-parametric bootstrap (sampling with replacement from the IC reference sample) is proposed (see Chapter 3 of this dissertation). A simulation study was conducted to test the conditional ARL performance of the NBEWMA control chart proposed in Chapter 3. The following distributions were used in the simulation study:

1. a standard normal distribution, i.e. $\mathcal{N}(0, 1)$,
2. t-distribution with 3 degrees of freedom, $t(3)$,
3. an exponentional distribution with mean equal to 1, $Exp(1)$ (or $Gamma(1, 1)$),
4. a Gamma distribution with scale parameter equal to 3 and shape parameter equal to 1, $Gamma(3, 1)$,
5. a Laplace distribution with location parameter 0 and scale parameter 1, $L(0, 1)$,
6. and Logistic distribution with location parameter 0 and scale parameter 1, $Logistic(0, 1)$.

Reference samples were generated, consisting of $m = 50, 100$ and 1 000 subgroups, respectively, each of different sizes $n = 1$, 5 and 1 000, respectively. This combination was chosen to be more reflective of a typical Phase I setting. The NBEWMA charts can be constructed usisng Equation (3.8), Algorithm 3.1 and Algorithm 3.2. The process mean was monitored using the sample mean, $\bar{x}$, since it is a commonly used location statistic (one could also use the sample median). In this section, we use closed-form expressions to calculate the bootstrap estimates for mean and standard error for the sample mean, when the process is IC. The derivation of the equations are shown in Section B.1 of this mini-dissertation.

[20] Knoth and Schmid (2015) showed that there have been many cases in the literature where EWMA charts, specifically for monitoring spread (typically the process variance), have been developed that are ARL-biased (see, e.g. [38] Wortham and Ringer (1971), [29] Ng and Case (1989) and [24] MacGregor and Harris (1993)). [20] Knoth and Schmid (2015) recommended using small values for λ, since this reduces the ARL-bias dramatically.

Any potential ARL-bias, of the NBEWMA chart, in the simulation study was minimised by setting $\lambda = 0.05$. The nominal IC ARL, i.e. ARL_0, was set equal to 500. Since the charts were constructed, using one data set generated from each distribution, the focus will be on the conditional attained ARL's. To measure the OOC performance of the chart, an upward (or downward) shift in magnitude of δ standard errors was added to the process mean, this was done for $-1.5 \leq \delta \leq 1.5$. The conditional IC ARL's for the $\mathcal{N}(0, 1)$, $t(3)$, $Exp(1)$, $Gamma(3, 1)$, $L(0, 1)$ and $Logistic(0, 1)$ cases, for $m = 50, 100, 1\,000$ and $n = 1, 5, 10$, are given in Tables 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 and 4.9, respectively. The simulation study was conducted using SAS v9.4 and the source code is available in Section A.3 of the Appendix.

Table 4.1: Summary statistics for the IC run-length distributions when using the NBEWMA chart versus a conventional EWMA control chart (given in parenthesis) with estimated IC process parameters, when $m = 50$ Phase I samples, each of size $n = 1$, were used to estimate the IC values of the process parameters.

Underlying Process Distribution	*ARL*	*SDRL*	Percentiles				
			5th	25th	50th	75th	95th
$\mathcal{N}(0,1)$	125.1487 (126.1438)	128.1563 (125.7447)	5 (4)	35 (36)	86 (89)	173 (175)	380 (377)
$t(3)$	129.9398 (126.6641)	140.4063 (134.0261)	2 (2)	31 (31)	86 (84)	181 (179)	409 (395)
$Exp(1)$	278.7150 (283.6591)	294.6640 (306.6476)	3 (2)	65 (66.5)	198 (186.5)	397 (394.5)	863 (889)
$Gamma(3,1)$	1284.4662 (1248.5240)	1321.2296 (1259.7625)	29.5 (18.5)	338.5 (320.5)	879.5 (857.5)	1805 (1732)	3872.5 (3837)
$L(0,1)$	561.8380 (563.9910)	555.6755 (570.8144)	30.5 (19)	169 (161)	385.5 (393)	795 (772.5)	1654.5 (1728)
$Logistic(0,1)$	693.5294 (697.0648)	692.6809 (706.1439)	28 (27)	191 (192)	478 (479)	974 (976.5)	2072.5 (2099)

From Table 4.1 it can be seen that the $\mathcal{N}(0,1)$, $t(3)$ and $Exp(1)$ distributions produce conditional IC *ARL*'s that are smaller than the nominal $ARL_0 = 500$, despite the control limits being adjusted to specifically achieve an IC *ARL* of 500 when using bootstrap samples to emulate the process distribution. From Table 4.1 it can be seen that the $Gamma(3,1)$, $L(0,1)$ and $Logistic(0,1)$ cases produced IC *ARL*'s that were much larger than the desired $ARL_0 = 500$. The percentiles in Table 4.1 give an idea of how the run-lengths were distributed when the NBEWMA was implemented in Phase II for different underlying process distributions.

Table 4.2: Summary statistics for the IC run-length distributions when using the NBEWMA chart versus a conventional EWMA control chart (given in parenthesis) with estimated IC process parameters, when $m = 50$ Phase I samples, each of size $n = 5$, were used to estimate the IC values of the process parameters.

Underlying Process Distribution	*ARL*	*SDRL*	Percentiles				
			5th	25th	50th	75th	95th
$\mathcal{N}(0,1)$	323.5578 (320.1552)	322.8327 (326.3702)	10 (10)	89 (89)	227 (219)	453 (447)	965 (964.5)
$t(3)$	155.8726 (156.7842)	165.1260 (162.7057)	3 (3)	39 (39)	105 (109)	217 (221)	481.5 (475)
$Exp(1)$	298.0540 (293.9975)	309.7303 (305.3862)	7.5 (8.5)	82 (79)	203 (197)	419.5 (406)	883 (894)
$Gamma(3,1)$	174.2340 (172.1644)	161.5419 (168.8018)	10.5 (7)	57.5 (53)	124.5 (122)	237.5 (237)	502 (507)
$L(0,1)$	654.5840 (674.0750)	684.8136 (687.7128)	17.5 (16)	180 (187)	444 (461)	891 (949)	1982 (2012.5)
$Logistic(0,1)$	758.9957 (755.5933)	769.0934 (783.6880)	24 (24)	207 (202)	521 (514)	1063 (1041.5)	2286 (2291.5)

From Table 4.2 it can be seen that for the $\mathcal{N}(0,1)$, $t(3)$, $Exp(1)$ and $Gamma(3,1)$ distributions, respectively, the NBEWMA chart produced IC *ARL*'s that were less than the nomal ARL_0 of 500, whilst the $L(0,1)$ and $Logistic(0,1)$ distributions produced IC *ARL*'s that were larger than 500. It can also be seen from Table 4.2 that there was no significant different in performance between the

NBEWMA chart and the conventional EWMA chart with estimated process mean and variance.

Table 4.3: Summary statistics for the IC run-length distributions when using the NBEWMA chart versus a conventional EWMA control chart (given in parenthesis) with estimated IC process parameters, when $m = 50$ Phase I samples, each of size $n = 10$, were used to estimate the IC values of the process parameters.

Underlying Process Distribution	*ARL*	*SDRL*	Percentiles				
			5th	25th	50th	75th	95th
$\mathcal{N}(0,1)$	140.9800 (142.3936)	135.1759 (137.7122)	8 (8)	46 (45)	102 (102)	192 (194)	411 (413)
$t(3)$	1401.4685 (1424.3249)	1281.4760 (1438.7274)	61 (59.5)	417 (382)	1007.5 (981)	2034 (1974)	4352 (4304)
$Exp(1)$	128.2590 (127.0175)	119.1728 (112.5368)	13 (15)	45 (47.5)	95 (95)	178 (172)	368.5 (349)
$Gamma(3,1)$	118.7170 (118.0140)	112.9180 (120.7201)	5 (4)	37 (33)	85 (80.5)	165.5 (163)	351.5 (354)
$L(0,1)$	214.0510 (212.4310)	209.0883 (207.7939)	15 (13)	70 (67)	153 (149)	292 (292)	595.5 (621.5)
$Logistic(0,1)$	195.0342 (194.3591)	190.1837 (189.4570)	12 (11)	62 (62)	138 (136)	268 (267)	570 (577)

2015

From Table 4.3 it can be seen that all six underlying process distributions produced IC *ARL*'s less than 500. It can also be seen in Table 4.3 that there was no significant different in performance between the NBEWMA chart and the conventional EWMA chart with estimated process mean and variance.

Table 4.4: Summary statistics for the IC run-length distributions when using the NBEWMA chart versus a conventional EWMA control chart (given in parenthesis) with estimated IC process parameters, when $m = 100$ Phase I samples, each of size $n = 1$, were used to estimate the IC values of the process parameters.

Underlying Process Distribution	*ARL*	*SDRL*	Percentiles				
			5th	25th	50th	75th	95th
$\mathcal{N}(0,1)$	243.7152 (242.2706)	249.8410 (231.0763)	8 (14)	67 (75)	170 (172)	339 (338)	731 (710)
$t(3)$	99.7746 (99.7920)	101.0281 (101.9523)	2 (2)	28 (27)	69 (69)	142 (140)	297 (306)
$Exp(1)$	272.4930 (277.1573)	285.9282 (304.9053)	2 (3)	65 (63)	181 (182)	390.5 (384)	828.5 (879)
$Gamma(3,1)$	189.3160 (192.6300)	189.5295 (192.6773)	5 (3)	58 (51.5)	134.5 (137.5)	259.5 (283)	559.5 (575.5)
$L(0,1)$	425.3750 (424.3427)	423.8685 (435.0104)	11.5 (10)	118.5 (113)	291.5 (293.5)	595.5 (593)	1274 (1276)
$Logistic(0,1)$	489.7498 (490.4518)	498.7364 (501.6566)	14 (14)	133.5 (135)	339 (334)	680 (680)	1496 (1488)

From Table 4.4 it can be seen that all six underlying process distributions produced IC *ARL*'s less than 500. It is interesting to note that the $L(0,1)$ and $Laplace(0,1)$ distributions produced IC *ARL*'s that are much closer to the nominal IC $ARL_0 = 500$. Again, It can seen in Table 4.4 that there was no significant different in performance between the NBEWMA chart and the conventional EWMA chart with estimated process mean and variance.

Table 4.5: Summary statistics for the IC run-length distributions when using the NBEWMA chart versus a conventional EWMA control chart (given in parenthesis) with estimated IC process parameters, when $m = 100$ Phase I samples, each of size $n = 5$, were used to estimate the IC values of the process parameters.

Underlying Process Distribution	*ARL*	*SDRL*	Percentiles				
			5th	25th	50th	75th	95th
$\mathcal{N}(0,1)$	210.8414 (207.7935)	210.6904 (209.8430)	8 (8)	62 (59)	146 (144)	292 (290.5)	626 (622)
$t(3)$	1517.9861 (1547.0236)	1373.5275 (1562.3598)	56 (52)	445 (414.5)	1087.5 (1059)	2214 (2174)	4786 (4694.5)
$Exp(1)$	275.0050 (277.0763)	260.2454 (263.7756)	21 (21)	82.5 (89)	193 (195)	390 (385)	771.5 (804)
$Gamma(3,1)$	172.0720 (169.4060)	182.0482 (175.3184)	5.5 (5)	52 (49)	114 (117)	233.5 (226)	523.5 (556)
$L(0,1)$	328.4980 (329.1805)	316.8112 (328.5283)	15.5 (14)	95.5 (94)	235 (229)	475 (458)	938 (997)
$Logistic(0,1)$	317.2831 (313.6287)	314.4513 (313.6838)	15 (13)	93 (92)	220 (219)	442 (433)	949 (932)

From Table 4.5 it can be seen that all six underlying process distributions produced IC *ARL*'s less than 500. No significant difference between the IC performance of the NBEWMA chart and the conventional EWMA with estimated values for the process mean and variance can be seen.

Table 4.6: Summary statistics for the IC run-length distributions when using the NBEWMA chart versus a conventional EWMA control chart (given in parenthesis) with estimated IC process parameters, when $m = 100$ Phase I samples, each of size $n = 10$, were used to estimate the IC values of the process parameters.

Underlying Process Distribution	*ARL*	*SDRL*	Percentiles				
			5th	25th	50th	75th	95th
$\mathcal{N}(0,1)$	347.3618 (345.2000)	349.1090 (351.0599)	11 (11)	97 (96)	243 (238)	487 (482)	1042 (1051)
$t(3)$	1206.8930 (1215.4754)	1152.5769 (1236.2749)	46 (41)	341 (337.5)	842 (832)	1711 (1688.5)	3687.5 (3692)
$Exp(1)$	374.4270 (368.9904)	373.2870 (375.2387)	16.5 (14)	104 (105)	255.5 (252)	525 (510.5)	1143.5 (1107)
$Gamma(3,1)$	276.3160 (276.6230)	279.2927 (281.2266)	8 (8)	84.5 (75.5)	193.5 (195)	381.5 (387.5)	833.5 (840)
$L(0,1)$	265.4100 (263.3815)	257.8081 (269.1061)	11 (10)	74 (74)	190.5 (178)	367.5 (365)	774 (795.5)
$Logistic(0,1)$	266.1720 (260.5684)	268.3931 (262.1348)	11 (10)	77 (75)	186 (180)	368 (360.5)	798.5 (784)

From Table 4.6 it can be seen that all six underlying process distributions produced IC *ARL*'s less than 500. No significant difference between the IC performance of the NBEWMA chart and the conventional EWMA with estimated values for the process mean and variance can be seen.

chapter a

Table 4.7: Summary statistics for the IC run-length distributions when using the NBEWMA chart versus a conventional EWMA control chart (given in parenthesis) with estimated IC process parameters, when $m = 1000$ Phase I samples, each of size $n = 1$, were used to estimate the IC values of the process parameters.

Underlying Process Distribution	*ARL*	*SDRL*	Percentiles				
			5th	25th	50th	75th	95th
$\mathcal{N}(0,1)$	478.9717 (480.2636)	487.1930 (493.4384)	12 (12)	132 (125)	330 (322)	668 (682)	1464 (1445)
$t(3)$	1463.6541 (1494.6339)	1347.9654 (1513.2246)	45 (41.5)	413.5 (397)	1036 (1028)	2153.5 (2087.5)	4667 (4562)
$Exp(1)$	687.3610 (686.9311)	684.3858 (719.0360)	9 (7)	163 (165)	456.5 (462)	998.5 (974)	2134.5 (2126)
$Gamma(3,1)$	392.2020 (386.9413)	421.4794 (403.6318)	8.5 (6)	101 (98)	258 (262)	540 (544)	1184 (1197.5)
$L(0,1)$	539.3120 (535.3770)	553.8567 (550.1273)	10.5 (8)	131.5 (135.5)	374.5 (365)	747.5 (754)	1672.5 (1643)
$Logistic(0,1)$	487.6441 (488.5199)	505.2175 (508.6631)	9 (9)	123 (125)	330 (327)	695.5 (679)	1500.5 (1513)

From Table 4.7 it can be seen that the $\mathcal{N}(0,1)$, $t(3)$, $Gamma(3,1)$ and $Logistic(0,1)$ distributions produced IC *ARL*'s less than 500. In Table 4.7 it can be seen that there is no significant difference between the IC performance of the NBEWMA chart and the conventional EWMA with estimated values for the process mean and variance.

Table 4.8: Summary statistics for the IC run-length distributions when using the NBEWMA chart versus a conventional EWMA control chart (given in parenthesis) with estimated IC process parameters, when $m = 1000$ Phase I samples, each of size $n = 5$, were used to estimate the IC values of the process parameters.

Underlying Process Distribution	*ARL*	*SDRL*	Percentiles				
			5th	25th	50th	75th	95th
$\mathcal{N}(0,1)$	531.9275 (537.0786)	553.6864 (557.2846)	14 (14)	140 (142)	363 (367)	734 (744)	1603 (1639)
$t(3)$	553.7659 (564.1165)	573.9586 (582.1677)	12 (11)	147 (148)	377 (384)	764 (790)	1704.5 (1729.5)
$Exp(1)$	518.1010 (520.3149)	536.8353 (523.1819)	14.5 (12)	138.5 (142)	329.5 (359)	728 (736)	1588 (1574)
$Gamma(3,1)$	501.7400 (498.6700)	500.1822 (518.4591)	10 (16)	136 (133.5)	365 (345.5)	698.5 (692)	1529.5 (1544.5)
$L(0,1)$	312.3030 (311.6744)	302.6070 (324.2712)	6.5 (7)	84.5 (81)	220 (213.5)	440.5 (431)	934.5 (974.5)
$Logistic(0,1)$	327.4753 (315.8558)	337.0085 (322.3036)	10 (8)	89 (85)	224.5 (216)	454 (439)	1006.5 (967.5)

In Table 4.8 it can be seen that the IC *ARL*'s for all the distributions are much closer to the target *ARL* of 500.

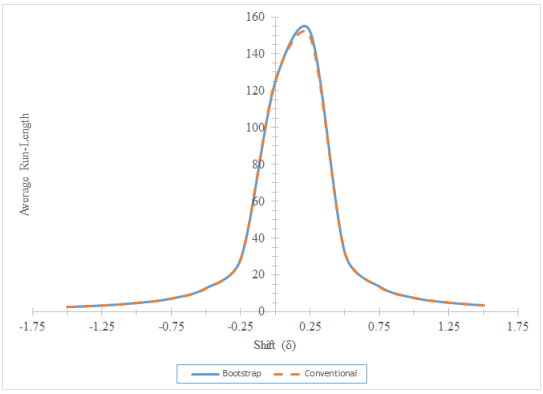
Table 4.9: Summary statistics for the IC run-length distributions when using the NBEWMA chart versus a conventional EWMA control chart (given in parenthesis) with estimated IC process parameters, when $m = 1000$ Phase I samples, each of size $n = 10$, were used to estimate the IC values of the process parameters.

Underlying Process Distribution	*ARL*	*SDRL*	Percentiles				
			5th	25th	50th	75th	95th
$\mathcal{N}(0,1)$	525.9098 (527.6114)	540.1841 (538.5985)	13 (15)	137 (142)	360 (359)	743 (737)	1617 (1606.5)
$t(3)$	481.9065 (482.0350)	505.0990 (496.6728)	9 (10)	121 (125.5)	332 (328)	664 (675)	1466 (1469.5)
$Exp(1)$	487.7240 (481.1230)	486.0607 (498.6627)	17.5 (14)	128 (130)	339 (324)	694.5 (668)	1468 (1472)
$Gamma(3,1)$	488.6020 (493.2570)	489.1681 (514.3483)	12.5 (9)	133.5 (124)	341 (343.5)	683.5 (692)	1446.5 (1513.5)
$L(0,1)$	376.3150 (359.8217)	370.6525 (368.6731)	7 (9)	100 (97.5)	261.5 (247.5)	532 (495)	1098.5 (1099.5)
$Logistic(0,1)$	376.6590 (377.2048)	384.6498 (380.8640)	11 (10)	103 (101)	256 (259)	519 (530)	1158.5 (1139)

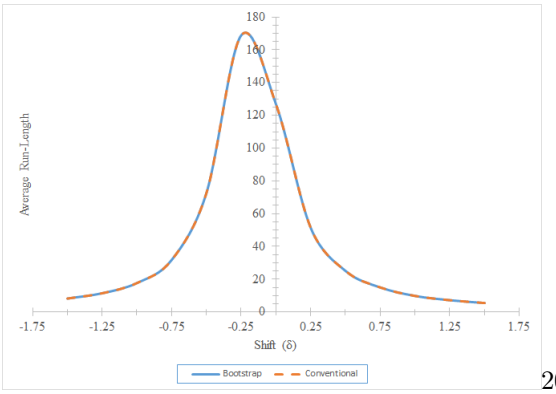
In Table 4.9 it can be seen that the IC *ARL*'s for the $\mathcal{N}(0,1)$, $t(3)$, $Exp(1)$ and $Gamma(3,1)$ are within 10% of the nominal $ARL_0 = 500$. The results in Tables 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 and 4.9, respectively, show that the IC performance of the NBEWMA chart is very similar to a conventional EWMA chart with estimated parameters. The results in Tables 4.7, 4.8 and 4.9, respectively, suggest that the bootstrap became better at emulating the process when a large enough number of subgroups (i.e. $m \geq 1\ 000$) was collected, in an IC reference sample. The results of the IC and OOC *ARL*'s for the distributions are summarised in Figure 4.1.

Figure 4.1: OOC *ARL* values for the NBEWMA-$\bar{X}$ control chart (blue graph) compared to the OOC *ARL* values for the conventional EWMA chart (orange graph) with estimated parameters, for $\lambda = 0.05$, $m = 50$ and $n = 1$.

(a) $\mathcal{N}(0,1)$

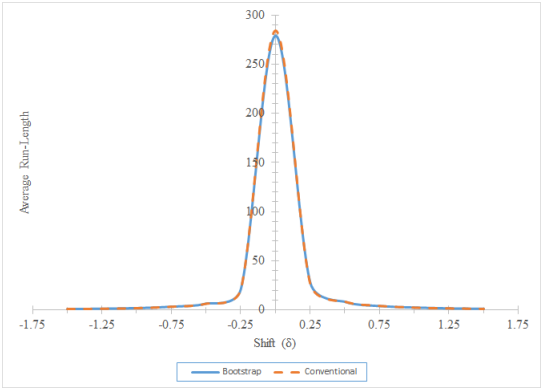


(b) $t(3)$

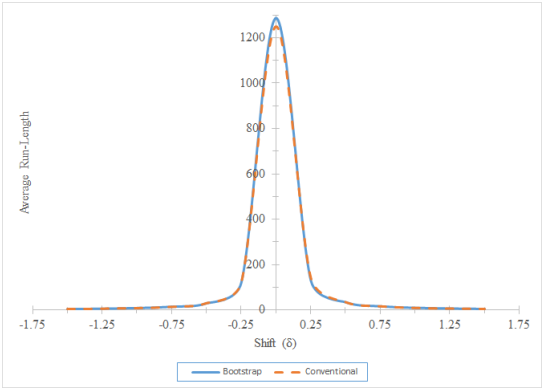


2015

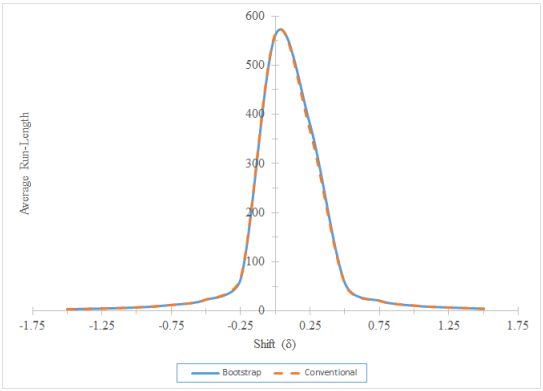
(c) $Exp(1)$



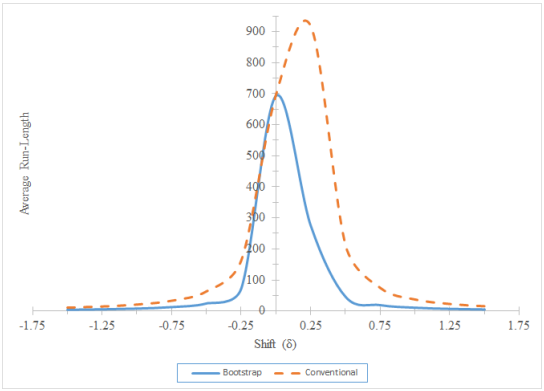
(d) $Gamma(3,1)$



(e) $L(0,1)$



(f) $Logistic(0,1)$

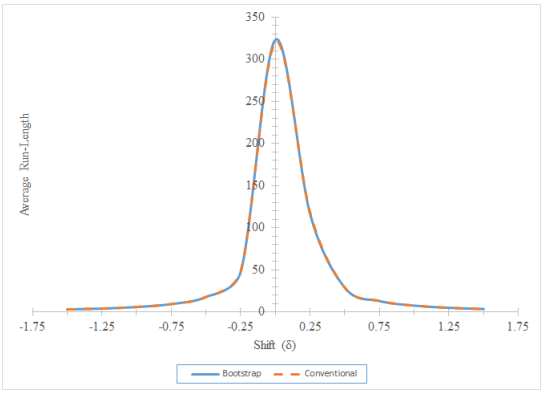


It can be seen in Figure 4.1a that both the bootstrap and conventional EWMA charts showed slight *ARL*-bias, this can be due to simulation error. The attained IC / OOC *ARL*'s for both the bootstrap and conventional EWMA control in Figure 4.1a seem identical and there appears to be no significant performance advantage (in terms of detecting shifts in magnitude of $|\delta| \leq 1.5$ standard deviations above/below the process mean). In Figure 4.1b it can be seen that both the bootstrap and conventional control limits produced a slight *ARL*-bias, when the underlying process distribution was $t(3)$. It can be seen in Figure 4.1b that the attained IC *ARL* is much lower than the desired value of

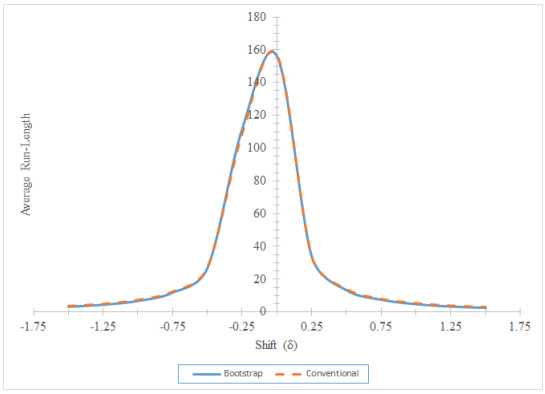
500. In Figure 4.1c it can be seen that the performance of the bootstrap control was equivalent to the performance of the conventional control chart, with the conventional EWMA control chart performing slightly better when the process is IC (i.e. $\delta = 0$), which can be attributed to simulation error, since the charts perform identically for the most part. Similarly, Figure 4.1d showed that the performance of the bootstrap chart was similar to that of the conventional control chart, with the latter performing slightly better when the process is IC. The bootstrap and conventional EWMA control charts for $L(0, 1)$ distribution shown in Figure 4.1e performed similarly and showed only slight ARL-bias. The $L(0, 1)$ distribution produced a higher IC ARL than specified. In Figure 4.1f it can be seen that the bootstrap control chart performed better than the conventional chart, since it had much lower ARL's when the process was OOC than its conventional counterpart.

Figure 4.2: OOC *ARL* values for the NBEWMA-$\bar{X}$ control chart (blue graph) compared to the OOC *ARL* values for the conventional EWMA chart (orange graph) with estimated parameters, for $\lambda = 0.05$, $m = 50$ and $n = 5$.

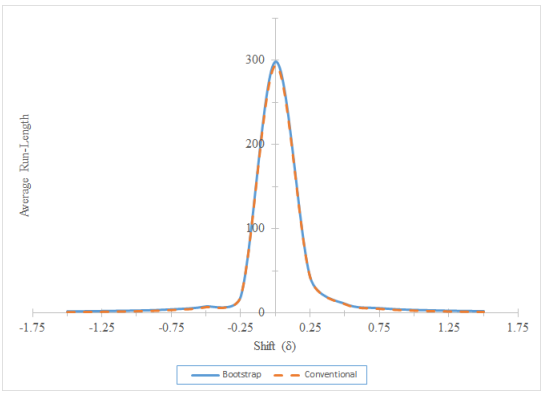
(a) $\mathcal{N}(0,1)$



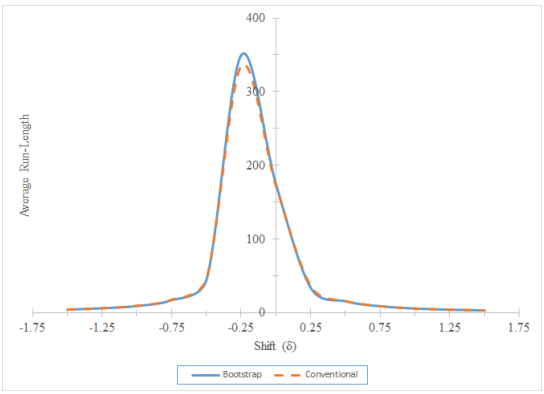
(b) $t(3)$



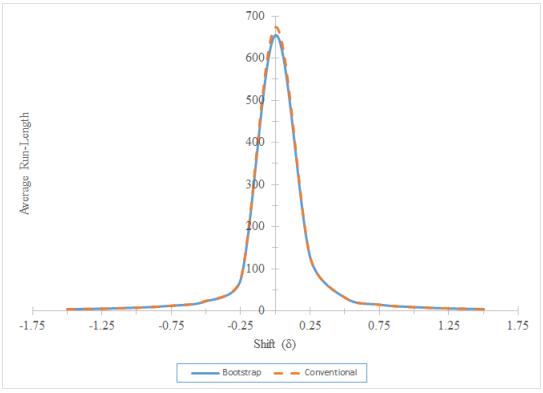
(c) $Exp(1)$



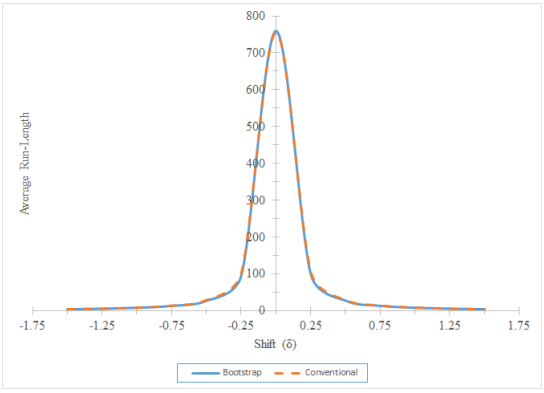
(d) $Gamma(3,1)$



(e) $L(0,1)$



(f) $Logistic(0,1)$

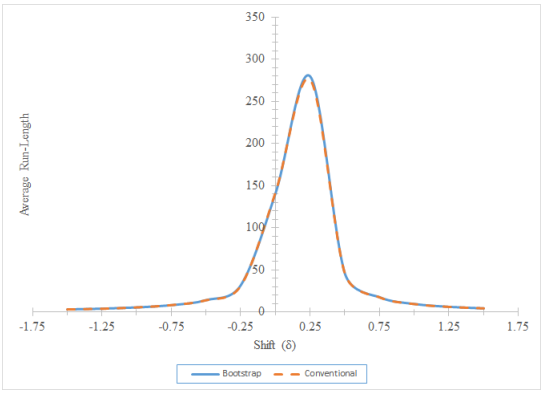


It can be seen in Figure 4.2 that the attained *ARL*'s for the $\mathcal{N}(0,1)$, $t(3)$, $Exp(1)$ and $Gamma(3,1)$ distributions were less than the nominal $ARL_0 = 500$. For the $L(0,1)$ and $Logistic(0,1)$ distributions, the attained *ARL*'s were more than 500. Both the bootstrap and conventional EWMA control charts showed *ARL*-bias for the $Gamma(3,1)$ case (see Figure 4.2d). The bootstrap and conventional EWMA charts performed similarly for the $\mathcal{N}(0,1)$, $t(3)$, $Exp(1)$ and $Logistic(0,1)$ cases. In Figure 4.2d it can be seen that the bootstrap control chart performs slightly better than the conventional control chart for $\delta \in [-0.25, 0)$. In Figure 4.2e it can be seen that the bootstrap control chart performs

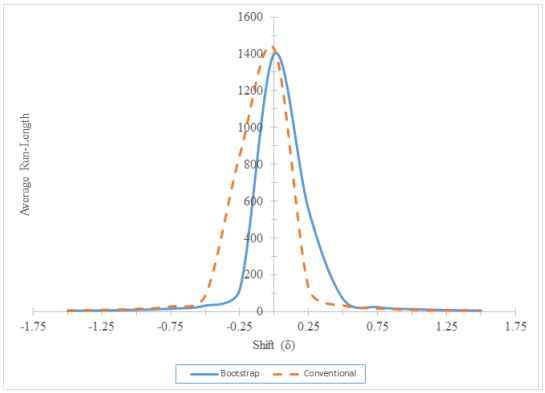
slightly better than the conventional control chart for $\delta \in [-0.25, 0)$. The better performance of the bootstrap control chart for the $Gamma(3,1)$ and $Logistic(0,1)$ cases, respectively, could be caused by simulation error.

Figure 4.3: OOC ARL values for the NBEWMA-$\bar{X}$ control chart (blue graph) compared to the OOC ARL values for the conventional EWMA (orange graph) chart with estimated parameters, for $\lambda = 0.05$, $m = 50$ and $n = 10$.

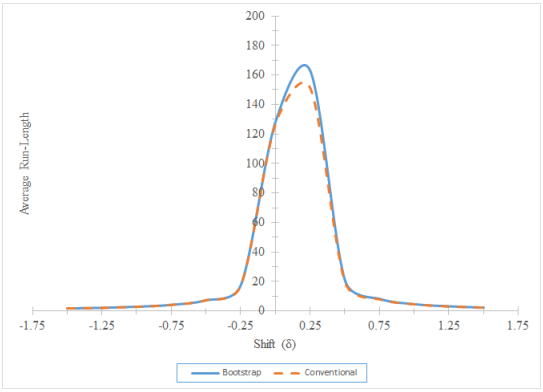
(a) $\mathcal{N}(0,1)$



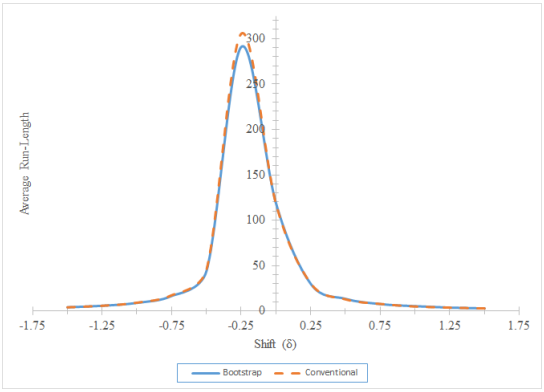
(b) $t(3)$



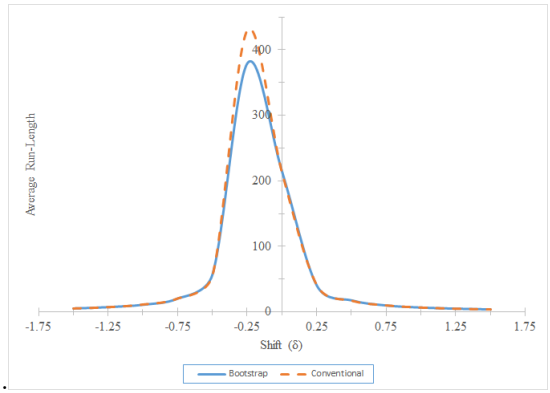
(c) $Exp(1)$



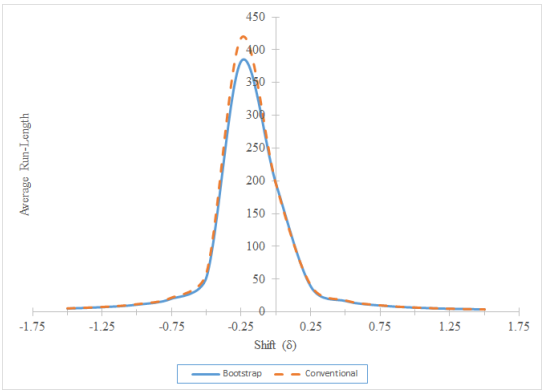
(d) $Gamma(3,1)$



(e) $L(0,1)$



(f) $Logistic(0,1)$

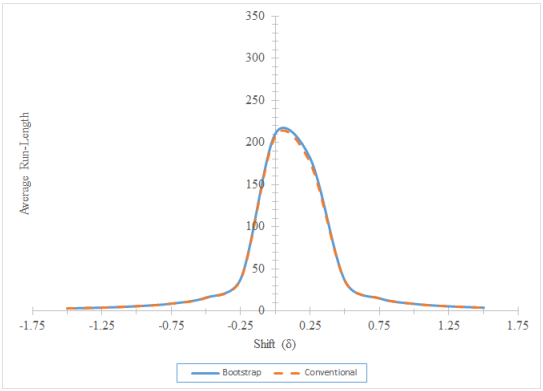


In Figure 4.3 it can be seen that the attained ARL's for the $\mathcal{N}(0,1)$, $Exp(1)$, $Gamma(3,1)$, $L(0,1)$ and $Logistic(0,1)$ cases, respectively, were less than the nominal value $ARL_0 = 500$. The attained ARL's for the $t(3)$ case were greater than the nomal value $ARL_0 = 500$. In 4.3b it can be seen that the bootstrap chart out-performed the conventional chart for $-0.5 \leq \delta \leq 0.5$. For the $t(3)$ case the

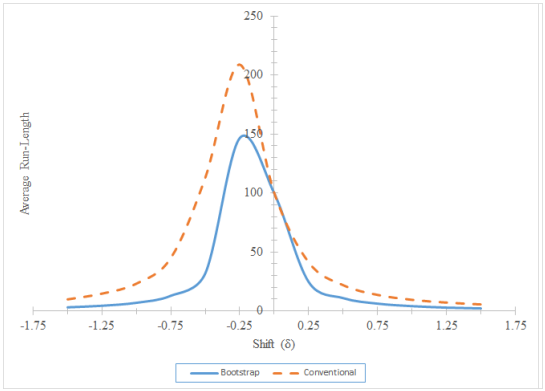
bootstrap chart out-performed the conventional chart for $-0.75 \leq \delta < 0$, whereas the conventional chart out-performed the bootstrap chart for $0 < \delta < 0.75$. In Figure 4.3e and 4.3f it can be seen that the bootstrap control chart out-performed the conventional chart for $-0.5 < \delta < 0$ for the $L(0,1)$ and $Logistic(0,1)$ cases, respectively. In the $Exp(1)$ case the conventional control chart out-performed the bootstrap control chart for $0 < \delta < 0.25$. The $\mathcal{N}(0,1)$, $Exp(1)$, $Gamma(3,1)$, $L(0,1)$ and $Logistic(0,1)$ cases were ARL-biased.

Figure 4.4: OOC ARL values for the NBEWMA-$\bar{X}$ control chart (blue graph) compared to the OOC ARL values for the conventional EWMA chart (orange graph) with estimated parameters, for $\lambda = 0.05$, $m = 100$ and $n = 1$.

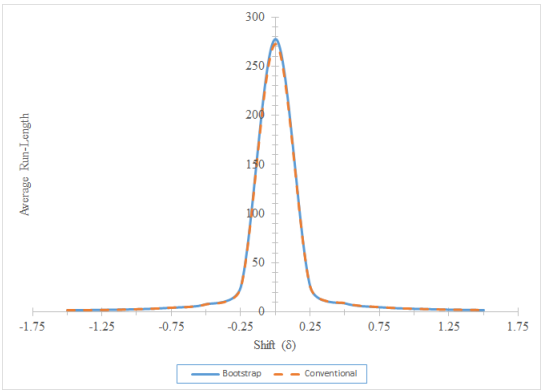
(a) $\mathcal{N}(0,1)$



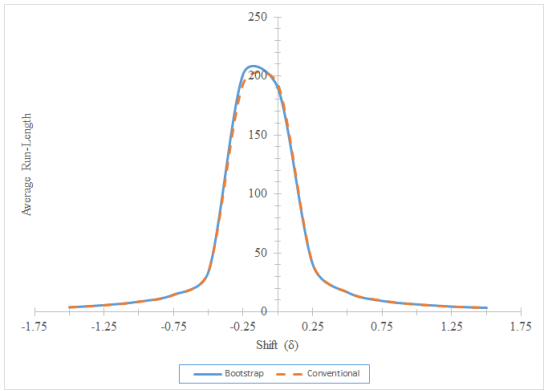
(b) $t(3)$



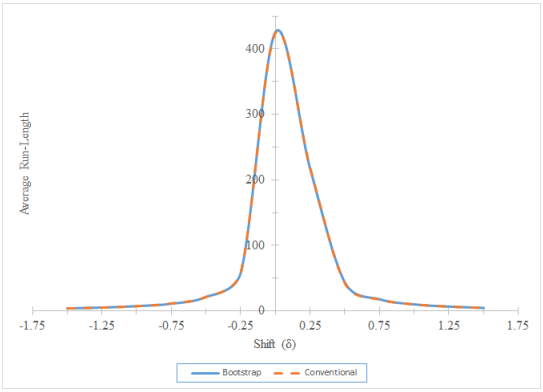
(c) $Exp(1)$



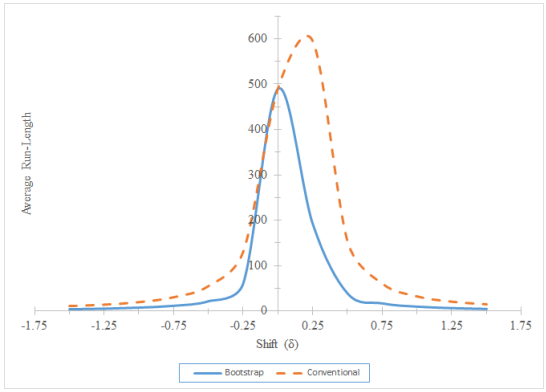
(d) $Gamma(3,1)$



(e) $L(0,1)$



(f) $Logistic(0,1)$

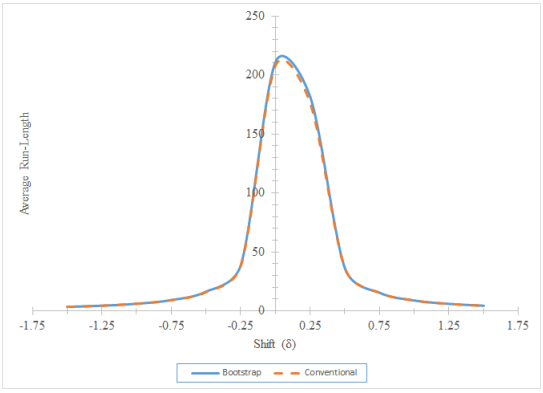


It can be seen in Figure 4.4 that for the $\mathcal{N}(0,1)$, $t(3)$, $Exp(1)$, $Gamma(3,1)$ and $L(0,1)$ distribu-

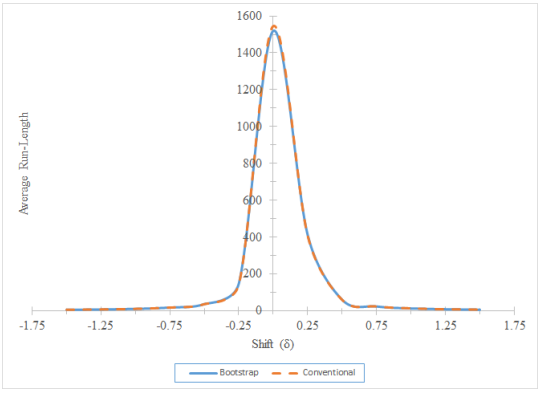
tions the attained ARL's are lower than the nominal value of $ARL_0 = 500$. The bootstrap control chart achieved the desired $ARL_0 = 500$ for the $Logistic(0, 1)$ distribution. In Figure 4.4f it can be seen that the bootstrap control chart out-performed the conventional control for $\delta > 0$. For the $t(3)$ case, it can be seen in Figure 4.4b that the conventional control chart out-performed the bootstrap chart when $\delta < 0$, however, the bootstrap control chart out-performed the conventional chart when $\delta > 0$. The manifestation of ARL-bias can clearly be seen in Figures 4.4a, 4.4b, 4.4d and 4.4f, respectively.

Figure 4.5: OOC ARL values for the NBEWMA-$\bar{X}$ control chart (blue graph) compared to the OOC ARL values for the conventional EWMA chart (orange graph) with estimated parameters, for $\lambda = 0.05$, $m = 100$ and $n = 5$.

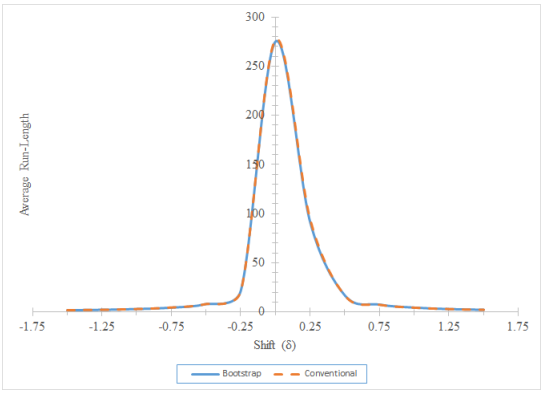
(a) $\mathcal{N}(0, 1)$



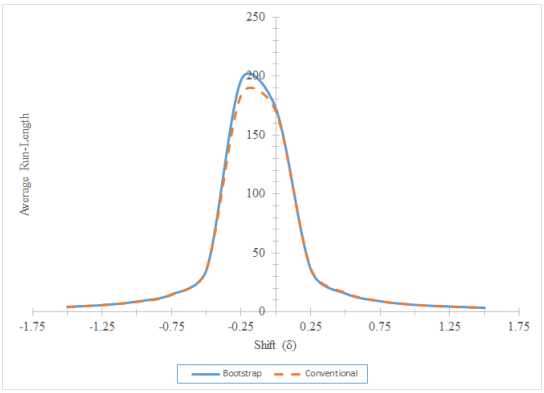
(b) $t(3)$



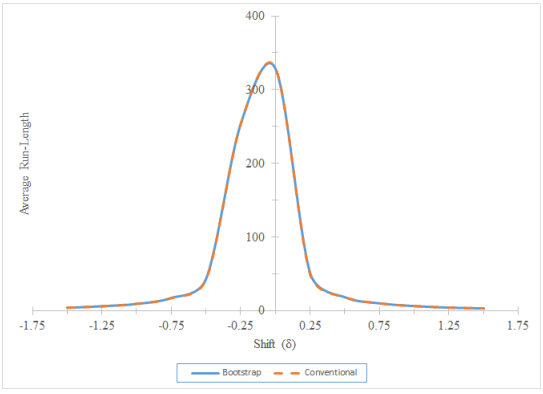
(c) $Exp(1)$



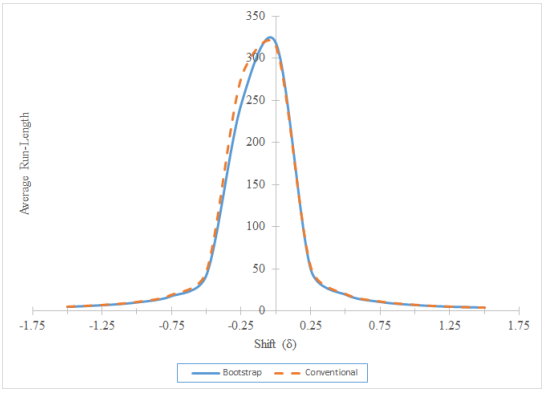
(d) $Gamma(3, 1)$



(e) $L(0, 1)$



(f) $Logistic(0, 1)$

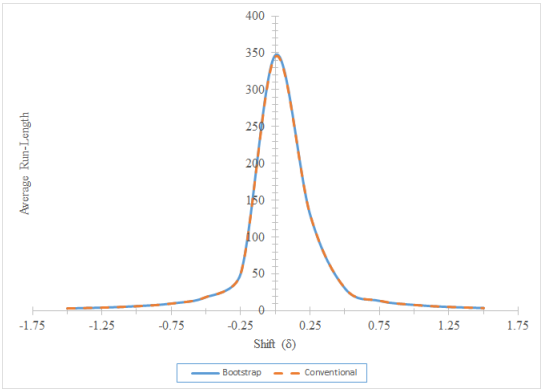


In Figure 4.5 it can be seen that all distributions, apart from the $t(3)$ distribution, attained ARL's

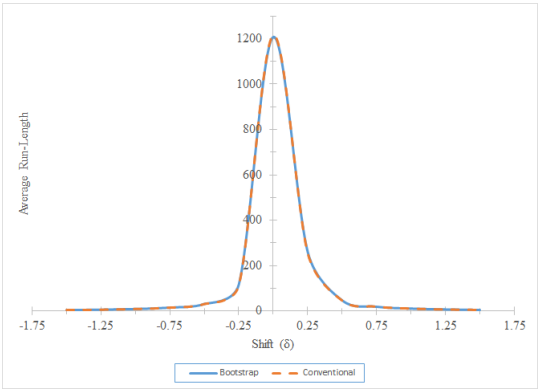
lower than the nominal value of $ARL_0 = 500$. The presence of ARL-bias can be seen in Figure 4.5a and 4.5d, respectively.

Figure 4.6: OOC ARL values for the NBEWMA-$\bar{X}$ control chart (blue graph) compared to the OOC ARL values for the conventional EWMA chart (orange graph) with estimated parameters, for $\lambda = 0.05$, $m = 100$ and $n = 10$.

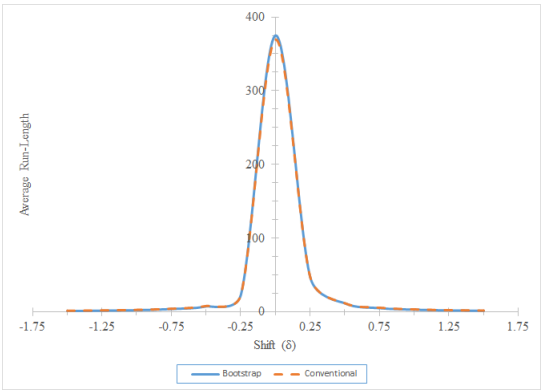
(a) $\mathcal{N}(0, 1)$



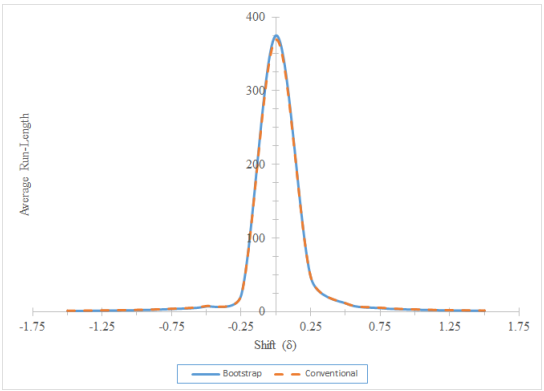
(b) $t(3)$



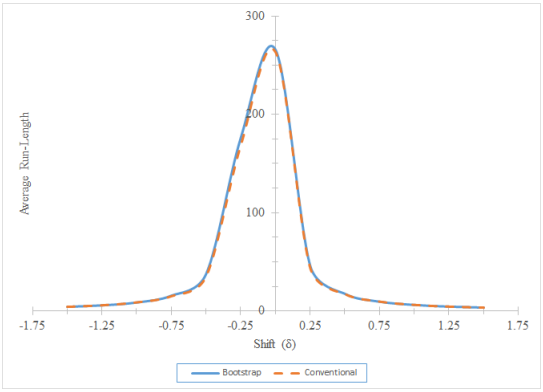
(c) $Exp(1)$



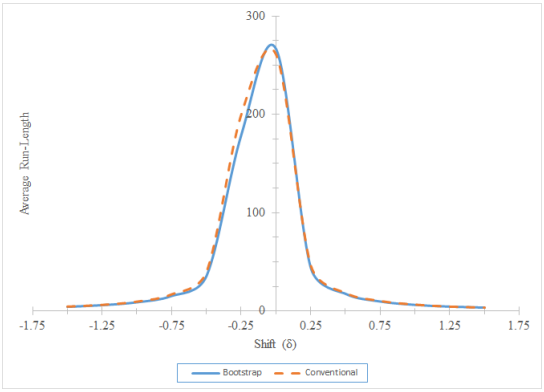
(d) $Gamma(3, 1)$



(e) $L(0, 1)$



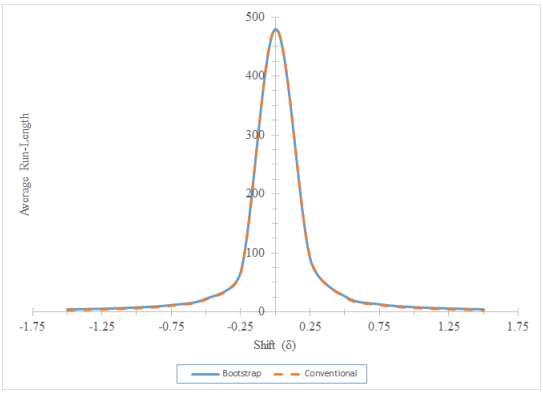
(f) $Logistic(0, 1)$



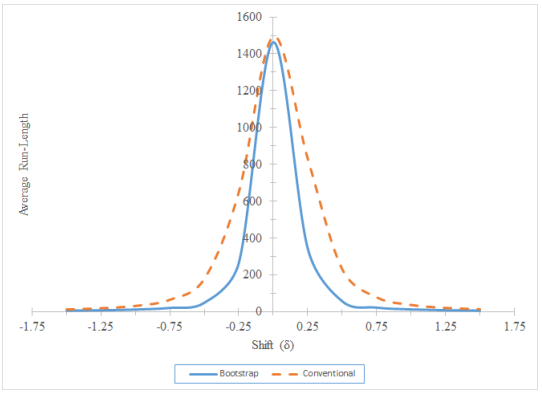
In Figure 4.6 it can be seen that the attained ARL's for all distributions, except the $t(3)$ distribution, were within approximate 50% of the target value of 500. It can also be seen in Figure 4.6 that there was no significant ARL-bias. The bootstrap chart showed no significant performance advantage compared to its conventional counterpart.

Figure 4.7: OOC *ARL* values for the NBEWMA-$\bar{X}$ control chart (blue graph) compared to the OOC *ARL* values for the conventional EWMA chart (orange graph) with estimated parameters, for $\lambda = 0.05$, $m = 1000$ and $n = 1$.

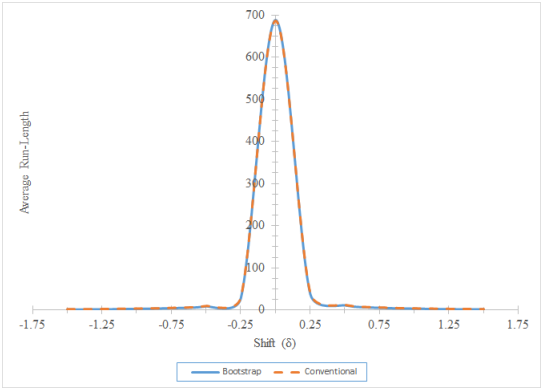
(a) $\mathcal{N}(0,1)$



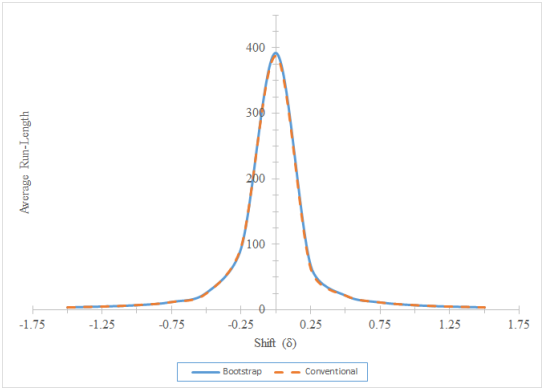
(b) $t(3)$



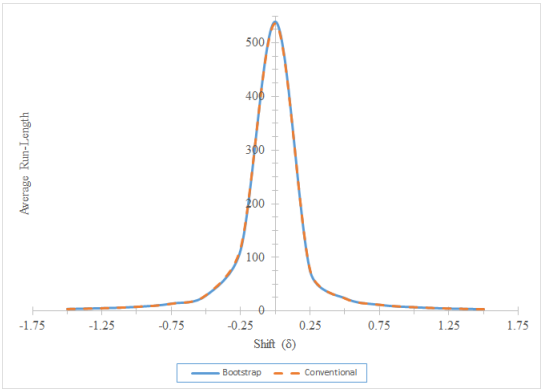
(c) $Exp(1)$



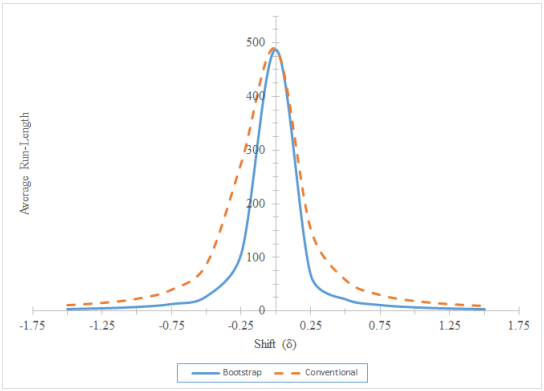
(d) $Gamma(3,1)$



(e) $L(0,1)$



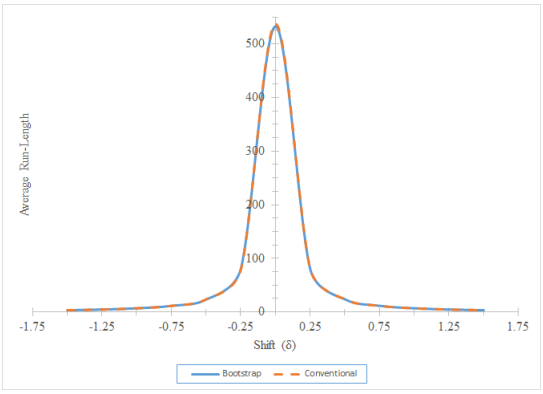
(f) $Logistic(0,1)$



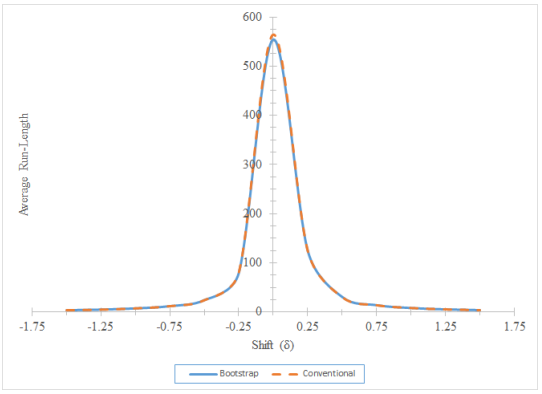
It can be seen in Figure 4.7 that the attained IC *ARL* values for the $\mathcal{N}(0,1)$, $Gamma(3,1)$, $L(0,1)$ and $Logistic(0,1)$ were within 10% of their target value $ARL_0 = 500$. For the $Logistic(0,1)$ case, the bootstrap control chart out-performed the conventional EWMA chart for $-1.5 \leq \delta \leq 1.5$. Similarly, the bootstrap control chart out-performed its conventional counterpart, for the $t(3)$ case (see Figure 4.7b), for all *non-zero* values of δ. The bootstrap chart performed similarly to the conventional chart for the $\mathcal{N}(0,1)$, $Exp(1)$, $Gamma(3,1)$ and $Logistic(0,1)$ cases, respectively.

Figure 4.8: OOC *ARL* values for the NBEWMA-$\bar{X}$ control chart (blue graph) compared to the OOC *ARL* values for the conventional EWMA chart (orange graph) with estimated parameters, for $\lambda = 0.05$, $m = 1000$ and $n = 5$.

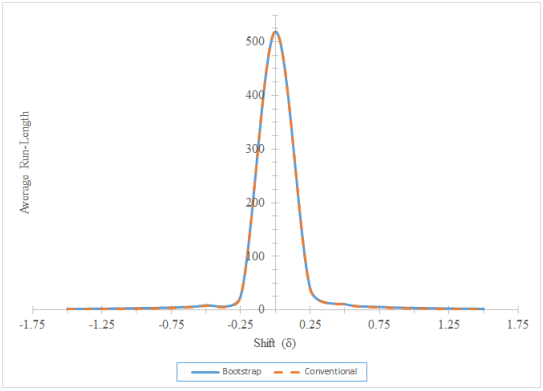
(a) $\mathcal{N}(0, 1)$



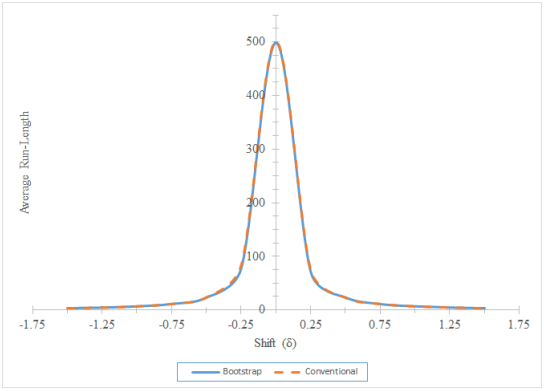
(b) $t(3)$



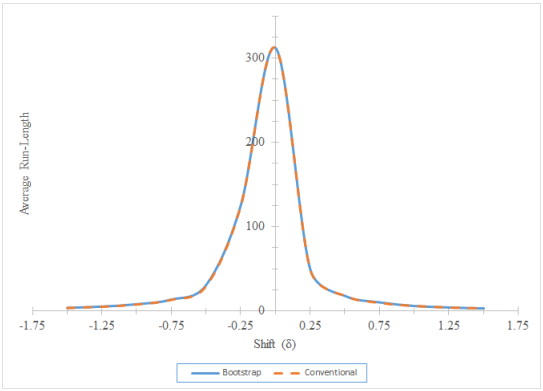
(c) $Exp(1)$



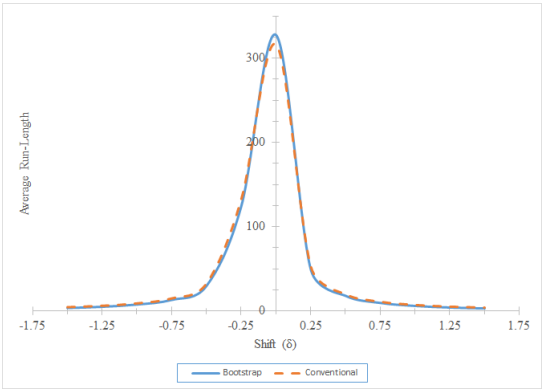
(d) $Gamma(3, 1)$



(e) $L(0, 1)$



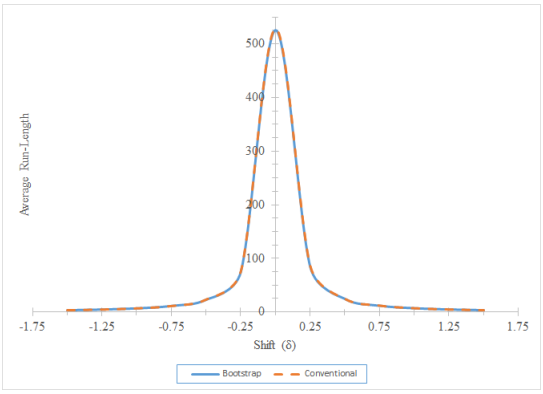
(f) $Logistic(0, 1)$



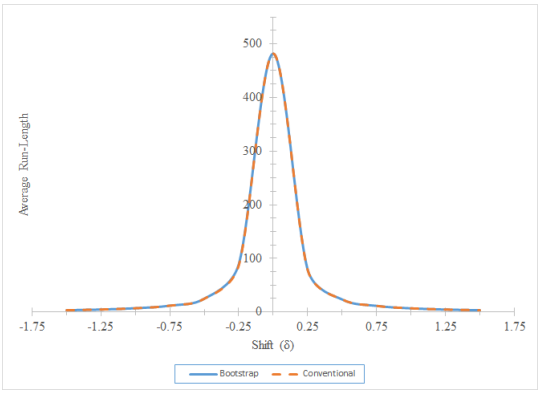
In Figure 4.8 it can be seen that the bootstrap control chart produced IC *ARL*'s within approximately 10% of $ARL_0 = 500$ for all distributions considered, except the $L(0, 1)$ and $Logistic(0, 1)$ distributions. No clear performance advantage, derived from either the bootstrap or conventional chart, was evident during this round of testing.

Figure 4.9: OOC ARL values for the NBEWMA-$\bar{X}$ control chart (blue graph) compared to the OOC ARL values for the conventional EWMA chart (orange graph) with estimated parameters, for $\lambda = 0.05$, $m = 1000$ and $n = 10$.

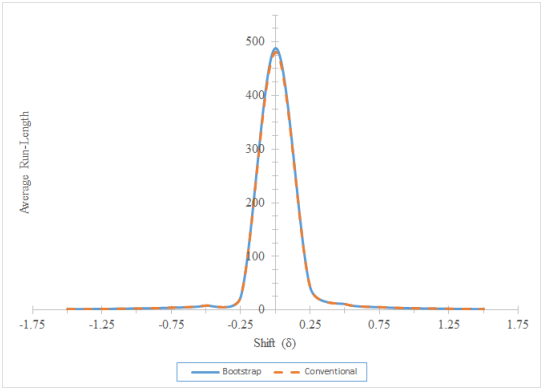
(a) $\mathcal{N}(0, 1)$



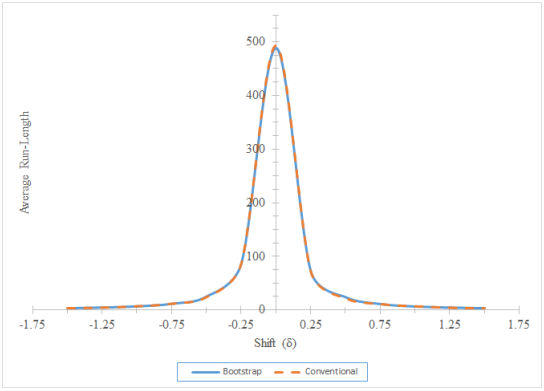
(b) $t(3)$



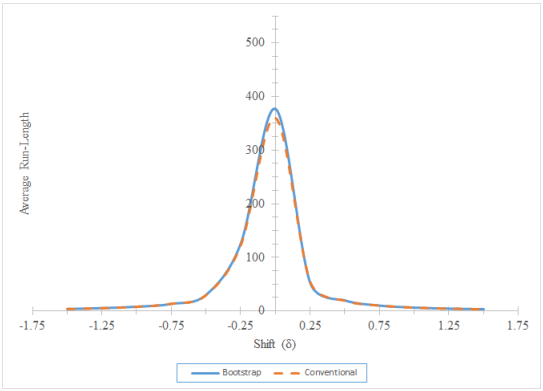
(c) $Exp(1)$



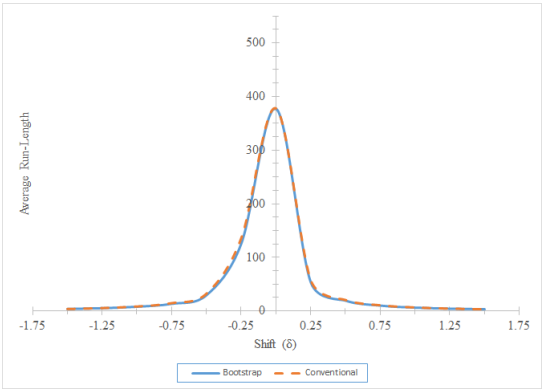
(d) $Gamma(3, 1)$



(e) $L(0, 1)$



(f) $Logistic(0, 1)$



It can be seen in Figure 4.9 that the bootstrap and conventional EWMA chart performed identically. The attained IC ARL's for the $L(0, 1)$ and $Logistic(0, 1)$ cases, respectively, were below the desired $ARL_0 = 500$. The availability of $m = 1\ 000$ subgroups of size $n = 10$, created a pool consisting of $1\ 000(10) = 10\ 000$ reference values from which the bootstrap control chart could be constructed. All distributions, except the $L(0, 1)$ and $Logistic(0, 1)$, produced IC ARL's that were within 10% of the target value. Clearly, the larger reference sample aided the selection of an appropriate L value. It can be seen in Figure 4.9 that there was no significant ARL-bias present.

4.3 Conclusion

A preliminary look at the NBEWMA-$\bar{X}$ chart, suggests that it could potentially be a viable alternative, provided that λ is sufficiently small and that the reference dataset is stable with very few extreme values. Different combinations of (m, n) were considered for the $\mathcal{N}(0,1)$, $t(3)$, $Exp(1)$, $Gamma(3,1)$, $L(0,1)$ and $Logistic(0,1)$ distributions, respectively. Figures 4.8 and 4.9, respectively, showed that the NBEWMA chart constructed using Algorithm 3.2 performs similarly to the conventional EWMA when $m = 1\ 000$.

[33] Saleh et al. (2015) showed that a large number ($m \geq 1000$) of subgroups, in the reference sample, were needed to construct EWMA control charts, with estimated process parameters, to produce $SDARL$'s that are within 10% of $ARL_0 = 500$. The results in Figures 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 and 4.9, respectively, show that m subgroups, where $m \geq 1000$, are needed to yield IC ARL's close to 500, which supports the findings of [33] Saleh et al. (2015).

The relatively large size of the IC reference sample required to construct the NBEWMA chart, brings its practicality in doubt. Practioners would need to consider the variability of the NBEWMA chart and the economic implications of collecting an adequate number of IC reference samples to construct it. The two-stage approach proposed by authors like [1] Ambartsoumian and Jeske (2015) could prove to be a better trade-off. [22] L. Liu, X. Zi, J. Zhang and Z. Wang (2013) proposed using what they termed a nonparametric adaptive EWMA (NAE) control chart. The NAE is effectively a self-starting chart (i.e. the process is monitored from the start with no IC reference data being collected first). [22] Liu et al. (2013) assumed that the the process readings collected over time came from the following change-point model:

$$x_i \sim \begin{cases} F(x, \mu_0) & \text{for} \quad t = 1, 2, \ldots, \tau, \\ F(x, \mu_1) & \text{for} \quad t = \tau + 1, \tau + 2, \ldots, \end{cases}$$

where $\tau \in \mathbb{N}$ is the unknown change point, μ_0 and μ_1 are the IC and OOC location parameters, respectively, and $F(\cdot)$ is an unknown continuous distribution function. Let R_n denote the nth sequential rank of x_n amoung $x_1, \ldots, x_n$, i.e.

$$R_n = \sum_{j=1}^{n} \mathbf{1}\{x_n \geq x_j\}. \tag{4.1}$$

[22] Liu et al. (2013) noted that the distribution of R_n varies as n increases. When the process is IC, [22] Liu et al. (2013) showed that

$$\mathbb{E}[R_n|\text{Process IC}] = \frac{(n+1)}{2}$$

and

$$Var[R_n|\text{Process IC}] = \frac{(n+1)(n-1)}{12},$$

provided that $n \geq 2$. It was argued by [22] Liu et al. (2013) that the standardized sequential rank, i.e.

$$R_n^* = \frac{R_n - \mathbb{E}\left[R_n|\text{Process IC}\right]}{\sqrt{Var\left[R_n|\text{Process IC}\right]}}, \tag{4.2}$$

should be used. It is reasonable to assume that a nonparametric EWMA (NEWMA) control chart based on R_n^* can be defined as

$$Z_n = (1-\lambda)Z_{n-1} + \lambda R_n^*, \tag{4.3}$$

where the initial value $Z_0 = 0$ and $\lambda \in (0, 1]$ is a suitable smoothing constant. [22] Liu et al. (2013) modified the NEWMA chart to be robust to various magnitudes of shifts - motivated by [4] Capizzi and Masarotto (2003) - by setting

$$Z_n = (1-\eta)Z_{n-1} + \eta R_n^*, \tag{4.4}$$

where $Z_0 = 0$ and the weight η is given by

$$\eta = 1 - \frac{1-\lambda}{\max\left\{1, \left|\bar{R}_{n,k}^*\right|/\omega\right\}}, \tag{4.5}$$

where $\bar{R}_{n,k}^* = k^{-1}\sum_{i=0}^{k-1} R_{n-i}^*$ denotes the average of the last k R_n^*'s, and k, ω, λ are pre-specified constants. [22] Liu et al. (2013) recommended using $(k, \omega, \lambda) = (5, 1.2, 0.03)$, since it was shown to provide a balanced performance against both small and large shifts in the process mean.

The NBEWMA proposed in this mini-dissertation can be accompanied by the NAE used by [22] Liu et al. (2013), effectively emulating the two-stage approach [1] Ambartsoumian and Jeske (2015) followed. Without loss of generality, the NAE used by [22] Liu et al. (2013) can be adapted, by re-defining R_n in Equation 4.1 as the nth sequential rank of ψ_n among $\psi_1, \ldots, \psi_n$, where ψ_i is some arbitrary statistic used to monitor the process. The process can then be monitored using the NAE until a sufficient number of IC subgroups are collected, say $m \geq 1\,000$.

Chapter 5

Summary of research and recommendations for future research

This mini-dissertation focused on constructing control charts, using bootstrap methods, to detect location shifts in the process distribution. The bootstrap offers an alternative to the parametric methods, typically utilised in a classical SPC environment. The nonparametric bootstrap, originally proposed by Efron (1979), is relatively easy to implement and makes naive assumptions about the statistical distribution.

[2] Bajgier (1992), [34] Seppala et al. (1995) and Liu and Tang (1996) discussed bootstrap implementations for the Shewhart control chart. The weaknesses of these methods were discussed by [18] Jones and Woodall (1998), more specifically, the variability of the bootstrap charts' IC ARL. When the underlying process distribution is skew, the bootstrap Shewhart control charts seem to perform better than their traditional counterparts.

The Shewhart bootstrap control charts were implemented in a Phase II setting to monitor the process distribution. Since Shewhart charts tend to be good at detecting larger shifts (see [27] Montgomery (2009)), a bootstrap control chart can be used in the Phase I setting to detect and remove outliers. This could potentially yield reasonable results, provided that the control limits are constructed using a trimmed sample (e.g. removing the bottom 25% of observations and top 25% of observations).

The most pressing issue with Shewhart control charts constructed using bootstrap procedures is the variability, with respect to their IC ARL's, we proposed applying [2] Bajgier's (1992) procedure, with the addition of constructing the control limits using the BCa procedure proposed by [11] Efron and Tibshirani (1994). In Section 2.2 it was shown that the BCa method could potentially reduce the variability of the IC ARL. More research should be done to find ways of minimising the variability, in terms of performance, of the bootstrap control limits for Shewhart charts.

[7] Chatterjee and Qiu (2009) and [1] Ambartsoumian and Jeske (2015) constructed CUSUM charts using the bootstrap, but with fundamentally different approaches. [7] Chatterjee and Qiu (2009) constructed a conventional upper one-sided CUSUM for detecting positive shifts in the process

mean by using a set of control limits, conditioned on the charting statistic and, what they termed, the sprint length (the number of observations since the last zero observation). The method proposed by [7] Chatterjee and Qiu (2009) seems robust and applicable for many scenarios, but the relationship between the variables necessary to construct the control chart is not well understood at present.

[1] Ambartsoumian and Jeske (2015) used a two-stage approach, where the IC Phase I process data is sequentially collected over time using a SRC control chart. When a sufficient number of observations were collected, [1] Ambartsoumian and Jeske (2015) used [31] Page's (1954) optimal CUSUM to monitor the process. [31] Page's (1954) optimal CUSUM uses the ratio of the OOC process distribution's PDF over the IC process distribution's PDF. [1] Ambartsoumian and Jeske (2015) used the AKDE to estimate both the IC and OOC PDF's.

One potential problem with the NDEC chart proposed by [1] Ambartsoumian and Jeske (2015) is the intense computation time required to construct the control chart, since a large number of IC observations are required to obtain a sufficiently good estimate of the IC process distribution's PDF. Fortunately, [1] Ambartsoumian and Jeske (2015) argued that the OOC process distribution can be approximated by a Beta distribution, from where the PITC was created. The PITC is computationally less intensive than its NDEC counterpart and provides similar levels of performance.

In future some more focus should be given to the performance of bootstrap CUSUM charts when very little reference data is available or obtaining large reference samples is economically unfeasible.

A parametric bootstrap (generating samples from a known distribution with, or without, estimating parameters) was suggested for the EWMA control chart by [13] Gandy and Kvaløy (2011). This approach assumes, without loss of generality, that the IC process distribution is $\mathcal{N}(0, \sqrt{n})$ and an iterative approach is followed to construct the control limits. We proposed a nonparametric bootstrap EWMA control chart. The chart showed signs of ARL-bias in our conditional run-length tests when a shift was introduced in the process mean. Knoth and Schmid (2015) recommended using small values of λ to reduce the impact of ARL-bias. Despite setting $\lambda = 0.05$, ARL-bias was still noted for some distributions. A simple solution to this problem can be to implement the NBEWMA with asymmetrically placed control limits, i.e.

$$UCL_i^* = \hat{\theta}_0^* + L_u \hat{\sigma}_0^* \sqrt{\left(\frac{\lambda}{2-\lambda}\right)(1-(1-\lambda)^{2i})}, \tag{5.1}$$

$$CL_i^* = \hat{\theta}_0 \tag{5.2}$$

and

$$LCL_i^* = \hat{\theta}_0^* + L_l \hat{\sigma}_0^* \sqrt{\left(\frac{\lambda}{2-\lambda}\right)(1-(1-\lambda)^{2i})}. \tag{5.3}$$

Algorithm 3.2 can be implemented, but with added layers of complexity namely: finding values for L_u and L_l, respectively, and ensuring that the OOC ARL is always less than the IC ARL. The

variability of the IC *ARL*'s for the NBEWMA chart should also be investigated. An EWMA chart transforms into a Shewhart chart when $\lambda = 1$, it is therefore reasonable to assume that it too will struggle with high levels of variability, with regard to its IC *ARL*.

Bootstrap control charts have the potential of becoming viable alternatives to their parametric counterparts derived in the Phase I setting, provided that they are constructed using a sufficient amount of data. With the availability of self starting charts, like the SRC employed by [1] Ambartsoumian and Jeske (2015), the economic feasibility and sustainability of bootstrap control charts is brought into serious doubt. The bootstrap could perhaps be better suited in a Phase I environment or to emulate the statistical distribution of a process so that informed decisions can be made on the most appropriate chart for a particular problem.

Appendix A

SAS programs used in the mini-dissertation

A.1 SAS programs used in Chapter 2

A.1.1 SAS program to calculate the ARL's of [2] Bajgier's (1992) bootstrap control chart and conventional control chart

SAS program that simulates a $\chi^2(5)$ in-control process distribution

```
1  options ls=100 nodate pageno=1 formdlim="_";
2
3  %let m=20;
4  %let n=5;
5  %let sim=1e4;
6  %let seed=88;
7  %let alpha=0.0027;
8  proc iml;
9  reset nolog;
10 m=&m;
11 n=&n;
12 x=j(&m,&n,.);
13 bigN = &m*&n;
14 call randseed(&seed);
15 call randgen(x,"CHI",5); *generates m*n random values from CHI(5);
16 xbarbar = mean(x)[,:];
17 sp2 = mean(var(x`))[,:];
18 print sp2;
19 a=log(sqrt(2));
```

```
b=lgamma((m*(n-1)+1)/2);
c=log ( sqrt (m*(n-1)));
d=lgamma((m*(n-1))/2);
lc4m=a+b-c-d;
c4m=exp(lc4m);
*print c4m;
muhat=xbarbar;
sigma_0=sqrt(sp2)/c4m;
print muhat sigma_0;
x = shape(x,bigN,1); *vec operator on m by n matrix x;

result=j(1000,2,.);
alpha = &alpha;
do i = 1 to 1000;
     xbar_b = j(1000,1,.);
     do j = 1 to 1000;
          u = int(ranuni(j(&n,1,0))#bigN)+1;
          x_b = x[u];
          xbar_b[j,1]=mean(x_b);
     end; *end do j;W
     p = (alpha/2) || (1-(alpha/2));
     call qntl(q,xbar_b,p);
     result[i,] = q';
end; *end do i;
result_final = mean(result);

*print result_final;
LL_se = sqrt(var(result[,1]));
UL_se = sqrt(var(result[,2]));
LL_avg = result_final[1,1];
UL_avg = result_final[1,2];
print LL_avg LL_se;
print UL_avg UL_se;
free result xbar_b;

*Phase II;

n=&n;

```

```
59 ARL_0=1/alpha;
60 stop=ARL_0 * 10;
61
62 do delta = -3 to 3 by 0.25;
63     runl=j(&sim,1,.);
64     shift = delta*(sqrt(2*5)/sqrt(n));
65     do i = 1 to &sim;
66         k=0;
67         flag=0;
68         do j = 1 to stop until (flag=1);
69             k=k+1;
70             *u=int(ranuni(j(n,1,0))#bigN)+1;
71             *x_obs = x[u];
72             x_obs = j(&n,1,.);
73             call randgen(x_obs,"CHI",5);
74             x_obs = x_obs + shift;
75             xbar = mean(x_obs);
76             *print xbar LL UL;
77             if (xbar<=LL_avg) | (xbar>=UL_avg) then flag=1;
78             *print flag;
79         end; *end do until;
80         runl[i]=k;
81     end; *end do i;
82     ARL = mean(runl);
83     if delta=0 then rem_ARL=ARL;
84     SDRL = sqrt(var(runl));
85     call qntl(q,runl,{0.05, 0.25, 0.5, 0.75, 0.95});
86     output1 = delta || ARL || SDRL || (q`);
87     output = output // output1;
88 end; *end do delta;
89 print "Run-Length Distribution for Bootstrap Control Limits with various
      shifts in process mean";
90 print output[colname={"delta" "arl" "sdrl" "p5" "p25" "p50" "p75" "p95
      "}];
91 *print rem_ARL;
92
93 create arl_data_bootstrap from output[colname={"delta" "arl" "sdrl" "p5"
      "p25" "p50" "p75" "p95"}];
94 append from output;
```

```
95  free output output1;
96
97  *Finding the optimal value for k;
98  ARL_0 = rem_ARL;
99  print ARL_0;
100 k_i_1 = 2; *first initial guess for k;
101 k_i = 3.5; *second initial guess for k;
102
103 LCL_i_1 = muhat - k_i_1*sigma_0/sqrt(n);
104 UCL_i_1 = muhat + k_i_1*sigma_0/sqrt(n);
105 LCL_i = muhat - k_i*sigma_0/sqrt(n);
106 UCL_i = muhat + k_i*sigma_0/sqrt(n);
107
108 FAR_i_1 = 1-(PROBCHI(UCL_i_1*n,5*n)-PROBCHI(LCL_i_1*n,5*n));
109 FAR_i = 1-(PROBCHI(UCL_i*n,5*n)-PROBCHI(LCL_i*n,5*n));
110
111 ARL_i_1 = 1/FAR_i_1;
112 ARL_i = 1/FAR_i;
113
114 print ARL_i_1 ARL_i;
115
116 perc_error1 = (abs(ARL_i_1-ARL_0)/ARL_0)*100;
117 perc_error2 = (abs(ARL_i-ARL_0)/ARL_0)*100;
118
119 it_matrix = (1 || k_i_1 || ARL_i_1 || perc_error1) // (2 || k_i || ARL_i
      || perc_error2);
120 c=2;
121 max_it = 30;
122 flag=0;
123 TOL=0.01; *0.1% error;
124 do it = 1 to max_it until (flag=1);
125     c=c+1;
126     k_i_1 = it_matrix[c-1,2];
127     k_i_2 = it_matrix[c-2,2];
128     ARL_i_1 = it_matrix[c-1,3];
129     ARL_i_2 = it_matrix[c-2,3];
130     k_i = ((ARL_0-ARL_i_1)*(k_i_1-k_i_2)/(ARL_i_1-ARL_i_2))+k_i_1;
131     LCL_i=muhat - k_i*sigma_0/sqrt(n);
132     UCL_i = muhat + k_i*sigma_0/sqrt(n);
```

```
133      IF LCL_i<0 then LCL_i=0;
134      FAR_i = 1-(PROBCHI(UCL_i*n,5*n)-PROBCHI(LCL_i*n,5*n));
135      ARL_i = 1/FAR_i;
136      perc_error = (abs(ARL_i - ARL_0))/ARL_0*100;
137      it_matrix = it_matrix // (c || k_i || ARL_i || perc_error);
138
139      if perc_error <= TOL then flag=1;
140  end; *end do it;
141  print it_matrix;
142
143  k = it_matrix[c,2];
144
145  LCLhat = muhat - k*sigma_0/sqrt(n);
146  UCLhat = muhat + k*sigma_0/sqrt(n);
147
148  print "Estimated Control Limits";
149  print LCLhat UCLhat;
150
151  runl=j(&siCodem,1);
152  flag=0;
153  do delta = -3 to 3 by 0.25;
154      shift = delta*(sqrt(10)/sqrt(5));
155      do i = 1 to &sim;
156              k=0;
157              flag=0;
158              do j = 1 to stop until (flag=1);
159                  k=k+1;
160                  *u=int(ranuni(j(n,1,0))#bigN)+1;
161                  *x_obs = x[u];
162                  x_obs = j(&n,1,.);
163                  call randgen(x_obs,"CHI",5);
164                  x_obs = x_obs + shift;
165                  xbar = mean(x_obs);
166                  *print xbar LL UL;
167                  if (xbar<=LCLhat) | (xbar>=UCLhat) then flag=1;
168                  *print flag;
169              end; *end do until;
170              runl[i]=k;
171          end; *end do i;
```

```
172      ARL=mean(runl);
173      SDRL=sqrt(var(runl));
174      call qntl(q,runl,{0.05, 0.25, 0.5, 0.75, 0.95});
175      output1 = delta || ARL || SDRL || (q');
176      output = output // output1;
177      *print ARL;
178  end; *end do delta;
179
180  create arl_data_conventional from output[colname={"delta" "arl" "sdrl" "
         p5" "p25" "p50" "p75" "p95"}];
181  append from output;
182
183  print "Run-Length Distribution for Estimated Control Limits with various
         shifts in process mean";
184  print output[colname={"delta" "arl" "sdrl" "p5" "p25" "p50" "p75" "p95
         "}];
185  free output output1;
186  quit;
187  data bootstrap1 (keep = delta arl1);
188  set arl_data_bootstrap;
189  arl1 = arl;
190  run;
191  data conventional1 (keep = delta arl2);
192  set arl_data_conventional;
193  arl2 = arl;
194  run;
195
196  data combined;
197  merge bootstrap1 conventional1;
198  run;
199  goptions reset=all;
200  symbol1 color=blue i=join;
201  symbol2 color=green i=join;
202  proc gplot data=combined;
203  plot (arl1 arl2)*delta /overlay;
204  run;
205  symbol;
206  quit;
207  resetline;
```

SAS program that simulates a $\mathcal{N}(0,1)$ in-control process distribution

```
1  options ls=100 nodate pageno=1 formdlim="_";
2
3  %let m=20;
4  %let n=5;
5  %let sim=1e4;
6  %let seed=88;
7  %let alpha=0.0027;
8  proc iml;
9  reset nolog;
10 m=&m;
11 n=&n;
12 x=j(&m,&n,.);
13 bigN = &m*&n;
14 call randseed(&seed);
15 call randgen(x,"NORMAL",0,1); *generates m*n random values from CHI(5);
16 xbarbar = mean(x)[,:];
17 sp2 = mean(var(x`))[,:];
18 print sp2;
19 a=log(sqrt(2));
20 b=lgamma((m*(n-1)+1)/2);
21 c=log ( sqrt (m*(n-1)));
22 d=lgamma((m*(n-1))/2);
23 lc4m=a+b-c-d;
24 c4m=exp(lc4m);
25 *print c4m;W
26 muhat=xbarbar;
27 sigma_0=sqrt(sp2)/c4m;
28 print muhat sigma_0;
29 x = shape(x,bigN,1); *vec operator on m by n matrix x;
30
31 result=j(1000,2,.);
32 alpha = &alpha;
33 do i = 1 to 1000;
34      xbar_b = j(1000,1,.);
35      do j = 1 to 1000;
36           u = int(ranuni(j(&n,1,0))#bigN)+1;
37           x_b = x[u];
```

```
38          xbar_b[j,1]=mean(x_b);
39      end; *end do j;
40      p = (alpha/2) || (1-(alpha/2));
41      call qntl(q,xbar_b,p);
42      result[i,] = q';
43  end; *end do i;
44  result_final = mean(result);
45
46  *print result_final;
47  LL_se = sqrt(var(result[,1]));
48  UL_se = sqrt(var(result[,2]));
49  LL_avg = result_final[1,1];
50  UL_avg = result_final[1,2];
51  print LL_avg LL_se;
52  print UL_avg UL_se;
53  free result xbar_b;
54
55  *Phase II;
56
57  n=&n;
58
59  ARL_0=1/alpha;
60  stop=ARL_0 * 10;
61
62  do delta = -3 to 3 by 0.25;
63      runl=j(&sim,1,.);
64      shift = delta*(sqrt(1)/sqrt(n));
65      do i = 1 to &sim;
66          k=0;
67          flag=0;
68          do j = 1 to stop until (flag=1);
69              k=k+1;
70              *u=int(ranuni(j(n,1,0))#bigN)+1;
71              *x_obs = x[u];
72              x_obs = j(&n,1,.);
73              call randgen(x_obs,"NORMAL",0,1);
74              x_obs = x_obs + shift;
75              xbar = mean(x_obs);
76              *print xbar LL UL;
```

```
77              if (xbar<=LL_avg) | (xbar>=UL_avg) then flag=1;
78              *print flag;
79          end; *end do until;
80          runl[i]=k;
81      end; *end do i;
82      ARL = mean(runl);
83      if delta=0 then rem_ARL=ARL;
84      SDRL = sqrt(var(runl));
85      call qntl(q,runl,{0.05, 0.25, 0.5, 0.75, 0.95});
86      output1 = delta || ARL || SDRL || (q`);
87      output = output // output1;
88  end; *end do delta;
89  print "Run-Length Distribution for Bootstrap Control Limits with various
        shifts in process mean";
90  print output[colname={"delta" "arl" "sdrl" "p5" "p25" "p50" "p75" "p95
        "}];
91  *print rem_ARL;
92
93  create arl_data_bootstrap from output[colname={"delta" "arl" "sdrl" "p5"
        "p25" "p50" "p75" "p95"}];
94  append from output;
95  free output output1;
96
97  *Finding the optimal value for k;
98  ARL_0 = rem_ARL;
99  print ARL_0;
100 k_i_1 = 2.8; *first initial guess for k;
101 k_i = 2.9; *second initial guess for k;
102
103 LCL_i_1 = muhat - k_i_1*sigma_0/sqrt(n);
104 UCL_i_1 = muhat + k_i_1*sigma_0/sqrt(n);
105 LCL_i = muhat - k_i*sigma_0/sqrt(n);
106 UCL_i = muhat + k_i*sigma_0/sqrt(n);
107
108 FAR_i_1 = 1-CDF('NORMAL',UCL_i_1,0,1/sqrt(n))+CDF('NORMAL',LCL_i_1,0,1/
        sqrt(n));
109 FAR_i = 1-CDF('NORMAL',UCL_i,0,1/sqrt(n))+CDF('NORMAL',LCL_i,0,1/sqrt(n))
        ;
110
```

```
111 ARL_i_1 = 1/FAR_i_1;
112 ARL_i = 1/FAR_i;
113
114 print ARL_i_1 ARL_i;
115
116
117 perc_error1 = (abs(ARL_i_1-ARL_0)/ARL_0)*100;
118 perc_error2 = (abs(ARL_i-ARL_0)/ARL_0)*100;
119
120 it_matrix = (1 || k_i_1 || ARL_i_1 || perc_error1) // (2 || k_i || ARL_i
        || perc_error2);
121 c=2;
122 max_it = 30;
123 flag=0;
124 TOL=0.01; *0.1% error;
125 do it = 1 to max_it until (flag=1);
126     c=c+1;
127     k_i_1 = it_matrix[c-1,2];
128     k_i_2 = it_matrix[c-2,2];
129     ARL_i_1 = it_matrix[c-1,3];
130     ARL_i_2 = it_matrix[c-2,3];
131     k_i = ((ARL_0-ARL_i_1)*(k_i_1-k_i_2)/(ARL_i_1-ARL_i_2))+k_i_1;
132     LCL_i=muhat - k_i*sigma_0/sqrt(n);
133     UCL_i = muhat + k_i*sigma_0/sqrt(n);
134     FAR_i = 1-CDF('NORMAL',UCL_i,0,1/SQRT(5))+CDF('NORMAL',LCL_i,0,1/sqrt
            (5));
135     *print FAR_i;
136     ARL_i = 1/FAR_i;
137     perc_error = (abs(ARL_i - ARL_0))/ARL_0*100;
138     it_matrix = it_matrix // (c || k_i || ARL_i || perc_error);
139
140     if perc_error <= TOL then flag=1;
141 end; *end do it;
142 print it_matrix;
143
144 k = it_matrix[c,2];
145
146 LCLhat = muhat - k*sigma_0/sqrt(n);
147 UCLhat = muhat + k*sigma_0/sqrt(n);
```

```
148
149 print "Estimated Control Limits";
150 print LCLhat UCLhat;
151
152 runl=j(&sim,1);
153 flag=0;
154 do delta = -3 to 3 by 0.25;
155     shift = delta*(sqrt(1)/sqrt(n));
156     do i = 1 to &sim;
157             k=0;
158             flag=0;
159             do j = 1 to stop until (flag=1);
160                 k=k+1;
161                 *u=int(ranuni(j(n,1,0))#bigN)+1;
162                 *x_obs = x[u];
163                 x_obs = j(&n,1,.);
164                 call randgen(x_obs,"NORMAL",0,1);
165                 x_obs = x_obs + shift;
166                 xbar = mean(x_obs);
167                 *print xbar LL UL;
168                 if (xbar<=LCLhat) | (xbar>=UCLhat) then flag=1;
169                 *print flag;
170             end; *end do until;
171             runl[i]=k;
172         end; *end do i;
173     ARL=mean(runl);
174     SDRL=sqrt(var(runl));
175     call qntl(q,runl,{0.05, 0.25, 0.5, 0.75, 0.95});
176     output1 = delta || ARL || SDRL || (q`);
177     output = output // output1;
178     *print ARL;
179 end; *end do delta;
180
181 create arl_data_conventional from output[colname={"delta" "arl" "sdrl" "
    p5" "p25" "p50" "p75" "p95"}];
182 append from output;
183
184 print "Run-Length Distribution for Estimated Control Limits with various
    shifts in process mean";
```

```
185 print output[colname={"delta" "arl" "sdrl" "p5" "p25" "p50" "p75" "p95
        "}];
186 free output output1;
187 quit;
188 data bootstrap1 (keep = delta arl1);
189 set arl_data_bootstrap;
190 arl1 = arl;
191 run;
192 data conventional1 (keep = delta arl2);
193 set arl_data_conventional;
194 arl2 = arl;
195 run;
196
197 data combined;
198 merge bootstrap1 conventional1;
199 run;
200 goptions reset=all;
201 symbol1 color=blue i=join;
202 symbol2 color=green i=join;
203 proc gplot data=combined;
204 plot (arl1 arl2)*delta /overlay;
205 run;
206 symbol;
207 quit;
208 resetline;
```

A.1.2 SAS program used in the simulation study of the subgroup bootstrap proposed by [34] Seppala et al. (1995)

The $Exp(1)$ case

```
options ls=100 nodate pageno=1;

%let m=20;
%let n=5;
%let alpha=0.0027;
%let sim=1e4;
goptions reset=all border;
proc iml;
reset nolog;
m=&m;
n=&n;
alpha=&alpha;
sim=&sim;
do j = 1 to sim;
     x=j(m,n,.);
     call randgen(x,"GAMMA",1);
     xbar_sub = (mean(x`))`;
     e_ij = x-xbar_sub;
     xbarbar = mean(mean(x)`);
     a=sqrt(n/(n-1));
     bigB = 2000;
     e_ij = shape(e_ij,m*n,1); *vec(e_ij) operator;
     x = shape(x,m*n,1);
     xbar_bCode = j(bigB,1,.0);
     do i = 1 to bigB;
          u=int(ranuni(j(n,1,0))#(m*n))+1;
          e_istar = e_ij[u];
          x_istar = xbarbar + a#e_istar;
          *print x_istar;
          xbar_b[i] = mean(x_istar);
     end; *end do i;
     *print xbar_b;
     call qntl(LCL_j,xbar_b,alpha/2);
     call qntl(UCL_j,xbar_b,1-(alpha/2));
     z_p = Quantile("NORMAL",1-alpha/2);
```

```
36      LCL_std = xbarbar - z_p * sqrt(var(x))/sqrt(n);
37      UCL_std = xbarbar + z_p * sqrt(var(x))/sqrt(n);
38      *print LCL_std UCL_std;
39      if LCL_std < 0 then LCL_std=0;
40      CVG_std_j = CDF("GAMMA",n*UCL_std,n,1)-CDF("GAMMA",n*LCL_std,n,1);
41      ARL_std_j = 1/(1-CVG_std_j);
42      *print ARL_std_j;
43      *print LCL_j UCL_j;
44      CVG_j = CDF("GAMMA",n*UCL_j,n,1)-CDF("GAMMA",n*LCL_j,n,1);
45      FAR_j = 1-CVG_j;
46      ARL_j = 1/FAR_j;
47      output1 = j || LCL_j || UCL_j || CVG_j || FAR_j || ARL_j;
48      output2 = j || ARL_std_j;
49      *print output1;
50      output_b = output_b // output1;
51      output_s = output_s // output2;
52  end; *end do j;
53  *print output;
54  *print output_s;
55  ARL_dist = output_b[,6];
56  avg_ARL = mean(output_b[,6]);
57  ARL_se = sqrt(var(output_b[,6]))/sqrt(sim);
58  p = {0.05 0.25 0.5 0.75 0.95};
59  call qntl(ARL_q,ARL_dist,p);
60  ARL_b_data = avg_ARL || ARL_se || ARL_q`;
61
62  ARL_s_dist = output_s[,2];
63  call qntl(ARL_q,ARL_s_dist,p);
64  ARL_s_data = mean(ARL_s_dist) || (sqrt(var(ARL_s_dist))/sqrt(sim)) || ARL
        _q`;
65  print m n alpha;
66  print "Bootstrap ARL Statistics";
67  l = {"Average" "SE" "p5" "Q1" "Q2" "Q3" "p95"};
68  print ARL_b_data[colname=l];
69  print "Standard ARL Statistics";
70  print ARL_s_data[colname=l];
71
72  create arl_data_b from arl_dist[colname={"ARL"}];
73  append from arl_dist;
```

```
74  create arl_data_s from arl_s_dist[colname={"ARL"}];
75  append from arl_s_dist;
76
77  quit;
78
79  data arl_data_b;
80  set arl_data_b;
81  Group="Bootstrap";
82  run;
83  data arl_data_s;
84  set arl_data_s;
85  Group="Standard";
86  run;
87  data together;
88  set arl_data_b arl_data_s;
89  run;
90  proc sort data=together;
91  by Group;
92  run;
93  /*
94  goptions reset = all;
95  axis1 value=("Bootstrap" "Standard") label=('Control Chart');
96  axis2 label = ('Average Run-Length (ARL)' a=90 justify=center);
97  symbol
98      interpol=boxt5
99      co=blue
100     bwidth=4
101     value=dot
102     cv=red
103     height=2;
104 axis1
105     label=none
106     value=(t=1 "Bootstrap" t=2 "Standard")
107     offset=(5,5)
108     length=50;
109 */
110 proc sgplot data=together;
111     xaxis max=6000 label='In-Control Average Run-Length';
112     yaxis label='Type of Control Chart';
```

```
113     hbox ARL /category=group WHISKERPCT=5 nooutliers;
114 run;
115 quit;
116 resetline;
```

The $\mathcal{N}(0,1)$ case

```
1  options ls=100 nodate pageno=1;
2
3  %let m=20;
4  %let n=5;
5  %let alpha=0.0027;
6  %let sim=1e4;
7
8  proc iml;
9  reset nolog;
10 m=&m;
11 n=&n;
12 alpha=&alpha;
13 sim=&sim;
14 do j = 1 to sim;
15      x=j(m,n,.);
16      call randgen(x,"NORMAL",0,1);
17      xbar_sub = (mean(x`))`;
18      e_ij = x-xbar_sub;
19      xbarbar = mean(mean(x)`);
20      a=sqrt(n/(n-1));
21      bigB = 2000;
22      e_ij = shape(e_ij,m*n,1); *vec(e_ij) operator;
23      x = shape(x,m*n,1);
24      xbar_b = j(bigB,1,.0);
25      do i = 1 to bigB;
26           u=int(ranuni(j(n,1,0))#(m*n))+1;
27           e_istar = e_ij[u];
28           x_istar = xbarbar + a#e_istar;
29           *print x_istar;
30           xbar_b[i] = mean(x_istar);
31      end; *end do i;
32      *print xbar_b;
```

```
33      call qntl(LCL_j,xbar_b,alpha/2);
34      call qntl(UCL_j,xbar_b,1-(alpha/2));
35      z_p = Quantile("NORMAL",1-alpha/2);
36      LCL_std = xbarbar - z_p * sqrt(var(x))/sqrt(n);
37      UCL_std = xbarbar + z_p * sqrt(var(x))/sqrt(n);
38      *print LCL_std UCL_std;
39      CVG_std_j = CDF("NORMAL",UCL_std,0,1/sqrt(n))-CDF("NORMAL",LCL_std
            ,0,1/sqrt(n));
40      ARL_std_j = 1/(1-CVG_std_j);
41      *print ARL_std_j;
42      *print LCL_j UCL_j;
43      CVG_j = CDF("NORMAL",UCL_j,0,1/sqrt(n))-CDF("NORMAL",LCL_j,0,1/sqrt(n
            ));
44      FAR_j = 1-CVG_j;
45      ARL_j = 1/FAR_j;
46      output1 = j || LCL_j || UCL_j || CVG_j || FAR_j || ARL_j;
47      output2 = j || ARL_std_j;
48      *print output1;
49      output_b = output_b // output1;
50      output_s = output_s // output2;
51  end; *end do j;
52  *print output;
53  *print output_s;
54  ARL_dist = output_b[,6];
55  avg_ARL = mean(output_b[,6]);
56  ARL_se = sqrt(var(output_b[,6]))/sqrt(sim);
57  p = {0.05 0.25 0.5 0.75 0.95};
58  call qntl(ARL_q,ARL_dist,p);
59  ARL_b_data = avg_ARL || ARL_se || ARL_q`;
60
61  ARL_s_dist = output_s[,2];
62  call qntl(ARL_q,ARL_s_dist,p);
63  ARL_s_data = mean(ARL_s_dist) || (sqrt(var(ARL_s_dist))/sqrt(sim)) || ARL
        _q`;
64  print m n alpha;
65  print "Bootstrap ARL Statistics";
66  l = {"Average" "SE" "p5" "Q1" "Q2" "Q3" "p95"};
67  print ARL_b_data[colname=l];
68  print "Standard ARL Statistics";
```

```
print ARL_s_data[colname=l];
create arl_data_b from arl_dist[colname={"ARL"}];
append from arl_dist;
create arl_data_s from arl_s_dist[colname={"ARL"}];
append from arl_s_dist;
quit;

data arl_data_b;
set arl_data_b;
Group="Bootstrap";
run;
data arl_data_s;SAS code for finding  for the NBEWMA Chart

set arl_data_s;
Group="Standard";
run;
data together;
set arl_data_b arl_data_s;
run;
proc sort data=together;
by Group;
run;
quit;
proc sgplot data=together;
    xaxis max=1500 label='In-Control Average Run-Length';
    yaxis label='Type of Control Chart';
    hbox ARL /category=group WHISKERPCT=5 connect=mean nooutliers;
run;
resetline;
```

A.1.3 SAS program used to construct the BCa control chart

```
options ls=100 nodate pageno=1;

%let m=20;
%let n=5;
%let alpha=0.0027;
%let sim=1e4;
goptions reset=all border;
proc iml;
reset nolog;
m=&m;
n=&n;
alpha=&alpha;
sim=&sim;
do j = 1 to sim;
    x=j(m,n,.);
    call randgen(x,"GAMMA",1);
    bigB = 2e3;
    x = shape(x,m*n,1);
    xbar_b = j(bigB,1,.0);
    xbarbar = mean(x);
    do i = 1 to bigB;
        u=int(ranuni(j(n,1,0))#(m*n))+1;
        x_istar = x[u];
        xbar_b[i] = mean(x_istar);
    end; *end do i;
    B=bigB;
    ind = xbar_b < xbarbar;
    zhat_0 = Quantile("NORMAL",mean(ind));
    *print zhat_0;
    xbar=mean(x);
    do i = 1 to n;
        ifCode i=1 then;
            do;
            x_idel = x[(i+1):n,1];
            *print xbar_idel_b;
            end;
        if i=n then;
```

```
38              do;
39                  x_idel = x[1:(n-1),1];
40                  *print xbar_idel_b;
41              end;
42          if i>1 && i<n then;
43          do;
44              x_idel = xbar_b[1:(i-1),1] // xbar_b[(i+1):n,];
45              *print xbar_idel_b;
46          end;
47          calc = mean(x_idel);
48          thetahat_i = thetahat_i // calc;
49      end;
50      thetahat = mean(x);
51      num = j(1,n,1)*((thetahat - thetahat_i)##3);
52      denom = 6*(j(1,n,1)*((thetahat - thetahat_i)##2))**(1.5);
53      *print num denom;
54      ahat = num/denom;
55      *print zhat_0 ahat;
56      z_L = Quantile("NORMAL",alpha/2);
57      z_U = Quantile("NORMAL",1-alpha/2);
58      *print z_L z_U;
59      in1 = zhat_0 + (zhat_0+z_L)/(1-ahat*(zhat_0+z_L));
60      alpha1 = CDF("NORMAL",in1,0,1);
61      in2 = zhat_0 + (zhat_0+z_U)/(1-ahat*(zhat_0+z_U));
62      alpha2 = CDF("NORMAL",in2,0,1);
63      *print alpha1 alpha2;
64      p = alpha1 || alpha2;
65      call qntl(LCL_j,xbar_b,alpha1);
66      call qntl(UCL_j,xbar_b,alpha2);
67      free thetahat_i;
68      z_p = Quantile("NORMAL",1-alpha/2);
69      LCL_std = xbarbar - z_p * sqrt(var(x))/sqrt(n);
70      UCL_std = xbarbar + z_p * sqrt(var(x))/sqrt(n);
71      *print LCL_std UCL_std;
72      if LCL_std < 0 then LCL_std=0;
73      CVG_std_j = CDF("GAMMA",n*UCL_std,n,1)-CDF("GAMMA",n*LCL_std,n,1);
74      ARL_std_j = 1/(1-CVG_std_j);
75      *print ARL_std_j;
76      *print LCL_j UCL_j;
```

```
77      CVG_j = CDF("GAMMA",n*UCL_j,n,1)-CDF("GAMMA",n*LCL_j,n,1);
78      FAR_j = 1-CVG_j;
79      ARL_j = 1/FAR_j;
80      output1 = j || LCL_j || UCL_j || CVG_j || FAR_j || ARL_j;
81      output2 = j || ARL_std_j;
82      *print output1;
83      output_b = output_b // output1;
84      output_s = output_s // output2;
85  end; *end do j;
86  *print output;
87  *print output_s;
88  ARL_dist = output_b[,6];
89  avg_ARL = mean(output_b[,6]);
90  ARL_se = sqrt(var(output_b[,6]))/sqrt(sim);
91  p = {0.05 0.25 0.5 0.75 0.95};
92  call qntl(ARL_q,ARL_dist,p);
93  ARL_b_data = avg_ARL || ARL_se || ARL_q';
94
95  ARL_s_dist = output_s[,2];
96  call qntl(ARL_q,ARL_s_dist,p);
97  ARL_s_data = mean(ARL_s_dist) || (sqrt(var(ARL_s_dist))/sqrt(sim)) || ARL
        _q';
98  print m n alpha;
99  print "Bootstrap ARL Statistics";
100 l = {"Average" "SE" "p5" "Q1" "Q2" "Q3" "p95"};
101 print ARL_b_data[colname=l];
102 print "Standard ARL Statistics";
103 print ARL_s_data[colname=l];
104
105 create arl_data_b from arl_dist[colname={"ARL"}];
106 append from arl_dist;
107 create arl_data_s from arl_s_dist[colname={"ARL"}];
108 append from arl_s_dist;
109
110 quit;
111
112 data arl_data_b;
113 set arl_data_b;
114 Group="Bootstrap";
```

```
115 run;
116 data arl_data_s;
117 set arl_data_s;
118 Group="Standard";
119 run;
120 data together;
121 set arl_data_b arl_data_s;
122 run;
123 proc sort data=together;
124 by Group;
125 run;
126 /*
127 goptions reset = all;
128 axis1 value=("Bootstrap" "Standard") label=('Control Chart');
129 axis2 label = ('Average Run-Length (ARL)' a=90 justify=center);
130 symbol
131     interpol=boxt5
132     co=blue
133     bwidth=4
134     value=dot
135     cv=red
136     height=2;
137 axis1
138     label=none
139     value=(t=1 "Bootstrap" t=2 "Standard")
140     offset=(5,5)
141     length=50;
142 */
143 proc sgplot data=together;
144     xaxis max=1000 label='In-Control Average Run-Length';
145     yaxis label='Type of Control Chart';
146     hbox ARL /category=group WHISKERPCT=5 connect=mean nooutliers;
147 run;
148 quit;
149 resetline;
```

A.1.4 SAS code used to construct and measure the performance of the PITC and TC charts for the $Exp(1)$ case

A.1.4.1 SAS program to find $\hat{a}$ and $\hat{b}$ for the PITC chart

```
1  options pageno=1 nodate ls=100;
2
3  %let bigN=15000;
4
5  proc iml;
6      reset nolog;
7      bigN=&bigN;
8      Y=j(bigN,1,.);
9      call randseed(123456);
10     call randgen(Y,"GAMMA",1,1);
11
12     sigmahat = sqrt(var(Y));
13     N=nrow(Y);
14     call qntl(q,Y,{0.25 0.75});
15     IQRhat = q[2]-q[1];
16     Ahat = min(sigmahat,IQRhat/1.34);
17     print Ahat;
18     hhat = Ahat*(4/(3*bigN))**(1/5);
19     print hhat;
20     /* fhat_y's are generated here */
21         f0hat_kde = j(N,1,.);
22     do i = 1 to N;
23         yval = Y[i];
24         vec = (j(n,1,yval)-Y)/hhat;
25         vec = PDF('NORMAL',vec,0,1);
26         test = sum(vec);
27         f0hat_kde[i] = sum(vec)/(N*hhat);
28     end; *end do i;
29     *print f0hat_kde;
30     ghat = geomean(f0hat_kde);SAS code for illustrative example 1
31     *print ghat;
32     lambda = (ghat/f0hat_kde)##0.5;
33     *print lambda;
34     hj = lambda##hhat;
35     *print hj;
```

```
36
37      start akde(x,hj,y);
38          n=nrow(y);
39          vec = (x-y)/hj;
40          vec = PDF('NORMAL',vec,0,1);
41          val = sum(vec)/n;
42          return(val);
43      finish akde;
44      start akde_cdf(x,hj,y);
45          n=nrow(y);
46          vec = (x-y)/hj;
47          vec = CDF('NORMAL',vec,0,1);
48          val = sum(vec)/n;
49          return(val);
50      finish akde_cdf;
51
52      *u=ranuni(123);
53      *print u;
54      start akde_cdf_inv(u,hj,y);
55          n=nrow(y);
56          y1=y;
57          TOL=1/(1e6);
58          call sort(y1,{1});
59          help_vec = y1 || (1:n)` || (1:n)`;
60          help_vec[,3] = help_vec[,3]/n;
61          dist = abs(help_vec[,3]-u);
62          help_vec = help_vec || dist;
63          call sort(help_vec,{4});
64          x1 = help_vec[1,1];
65          x2 = help_vec[2,1];
66          *print help_vec;
67          *print x1 x2;
68          Fhat_x1 = akde_cdf(x1,hj,y);
69          Fhat_x2 = akde_cdf(x2,hj,y);
70          *print x1 Fhat_x1;
71          *print x2 Fhat_x2;
72          it=30;
73          it_matrix = (1:2)` || (x1 // x2) || (Fhat_x1 // Fhat_x2);
74          *print it_matrix;
```

```
75          k=2;
76          flag=0;
77          do i = 1 to it until (flag=1);
78              k=k+1;
79              x_n = it_matrix[k-2,2];
80              x_n_1 = it_matrix[k-1,2];
81              Fhat_n = it_matrix[k-2,3];
82              Fhat_n_1 = it_matrix[k-1,3];
83              x_n_2 = x_n_1 - ((x_n_1-x_n)/(Fhat_n_1-Fhat_n))*(Fhat_n_1-u);
84              Fhat_n_2 = akde_cdf(x_n_2,hj,y);
85              it_matrix = it_matrix // (k || x_n_2 || Fhat_n_2);
86              test = abs(Fhat_n_2 - Fhat_n_1);
87              if test<=TOL then flag=1;
88              *print it_matrix;
89          end;
90          last = it_matrix[k,2];
91          return(last);
92      finish akde_cdf_inv;
93
94      no_ints = 1000;
95      h = (1-0)/no_ints;
96      bigK=0.5*1;
97      do i = 1 to no_ints;
98          xi_star = ((0+i*h) + (0+(i-1)*h))/2;
99          val1 = akde_cdf_inv(xi_star,hj,y);
100         val2 = akde_cdf(val1-bigK,hj,y);
101         val2 = 1 - val2;
102         keep1 = keep1 // (val2*h);
103         keep2 = keep2 // (val2*2*xi_star*h);
104     end;
105     m1hat = sum(keep1);
106     m2hat = sum(keep2);
107     print m1hat m2hat;
108
109     ahat = (m1hat**2 - m1hat*m2hat)/(m2hat-m1hat**2);
110     bhat = ((m1hat-m2hat)*(1-m1haSAS code for finding  for the NBEWMA
            Chart
111 t))/(m2hat-m1hat**2);
112
```

```
113     print ahat bhat;
114 quit;
115 resetline;
```

A.1.4.2 SAS program to find the threshold H value for the PITC chart

```
1  options pageno=1 nodate ls=100;
2
3  proc iml;
4  reset nolog;
5  ahat=1.400553;
6  bhat=0.9053615;
7  sims=10000;
8  h=4.72;
9  runl=j(sims,1,.);
10 do i = 1 to sims;he derivation of the equations are shown in Section  of
       this dissertation.
11      flag=0;
12      s0_hat = 0;
13      k=0;
14      do j = 1 to 20000 until (flag=1);
15           k=k+1;
16           u=ranuni(0);
17           val = s0_hat+(ahat-1)*log(u)+(bhat-1)*log(1-u)-log(beta(ahat,bhat
                 ));
18           s1_hat = max(0,val);
19           if s1_hat>=h then flag=1;
20           s0_hat=s1_hat;
21      end;
22      runl[i]=k;
23 end;
24 arl = mean(runl);
25 print arl;
26 quit;
```

A.1.4.3 SAS Program to measure the performance of PITC chart

```
1  options pageno=1 nodate ls=100;
2
```

```
proc iml;
reset nolog;
ahat=1.400553;
bhat=0.9053615;
sims=10000;
h=4.72;
cache = {0 0.1 0.25 0.5 0.75 1};
dtotal = ncol(cache);
do d = 1 to dtotal;
    runl=j(sims,1,.);
    delta = cache[d]*1;
    do i = 1 to sims;
        flag=0;
        s0_hat = 0;
        k=0;
        do j = 1 to 22000 until (flag=1);
            if j>=2000 then k=k+1;
            x=j(1,1,0);
            call randgen(x,"GAMMA",1,1);
            x = x + delta;
            u=CDF("GAMMA",x,1,1);
            val = s0_hat+(ahat-1)*log(u)+(bhat-1)*log(1-u)-log(beta(ahat,
                bhat));
            s1_hat = max(0,val);
            if (s1_hat>=h) & (j<2000) then s0_hat=0;
            if (s1_hat>=h) & (j>=2000) then flag=1;
            if (s1_hat<h) then s0_hat=s1_hat;
        end;
        runl[i]=k;
    end;
    arl = mean(runl);
    stats = stats // (cache[d] || arl);
    *print arl;
end;
print stats;
quit;
```

A.1.4.4 SAS program to find the threshold value H for the TC chart

```
1  options pageno=1 nodate ls=100;
2
3  %let bigN=15000;
4
5  proc iml;
6  reset nolog;
7  bigN=&bigN;
8  Y=j(bigN,1,.);
9  call randseed(123456);
10 call randgen(Y,"GAMMA",1,1);
11 start edf(x,y);
12     prob = mean(x<=y);
13     return(prob);
14 finish edf;
15
16 start RandBetween(n, min, max);
17     u = j(n,1,0);
18     u = int(ranuni(u)#(max+1-min))+min;
19     return(u);
20 finish;
21
22 H=12.7;
23 alpha=0.5;
24 nosims=10000;
25 runl=j(nosims,1,.);
26 do i = 1 to nosims;
27         k=0;
28         flag=0;
29         S0_hat=0;
30         do j =  1 to 20000 until (flag=1);
31             k=k+1;
32             u1 = RandBetween(1,0,bigN)/bigN;
33             val = S0_hat + u1 - alpha;
34             S1_hat = max(0,val);
35             *print S1_hat;
36             if S1_hat>=H then flag=1;
37             S0_hat = S1_hat;
38         end;
39     runl[i]=k;
```

```
40 end;
41 *print runl;
42 arl=mean(runl);
43 print arl;
44 quit;
45 resetline;
```

A.1.4.5 SAS program to measure the performance of TC chart

```
1  options pageno=1 nodate ls=100;
2
3  %let bigN=15000;
4
5  proc iml;
6  reset nolog;
7  bigN=&bigN;
8  Y=j(bigN,1,.);
9  call randseed(123456);
10 call randgen(Y,"GAMMA",1,1);
11 start edf(x,y);
12     prob = mean(x<=y);
13     return(prob);
14 finish edf;
15
16 start RandBetween(n, min, max);
17     u = j(n,1,0);
18     u = int(ranuni(u)#(max+1-min))+min;
19     return(u);
20 finish;
21
22 H=12.7;
23 alpha=0.5;
24 nosims=10000;
25 cache = {0 0.1 0.25 0.5 0.75 1};
26 bigD = ncol(cache);
27 do d = 1 to bigD;
28     delta = cache[d]*1;
29     runl=j(nosims,1,.);
30     do i = 1 to nosims;
```

```
31          k=0;
32          flag=0;
33          S0_hat=0;
34          do j =  1 to 22000 until (flag=1);
35               if j>=2000 then k=k+1;
36               x=log(1-ranuni(0))*-1;
37               if j>=2000 then x=x+delta;
38               u1 = CDF("GAMMA",x,1,1);
39               *u1 = edf(x,y);
40               val = S0_hat + u1 - alpha;
41               S1_hat = max(0,val);
42               if S1_hat>=H & j<2000 then S0_hat=0;
43               if S1_hat>=H & j>=2000 then flag=1;
44               if S1_hat<H then S0_hat=S1_hat;
45               *S0_hat = S1_hat;
46          end;
47          runl[i]=k;
48     end;
49     *print runl;
50     arl=mean(runl);
51     *print arl;
52     output = output // (delta || arl);
53 end;
54 print output;
55 quit;
56 resetline;
```

A.2 SAS programs used in Chapter 3

A.2.1 SAS code for illustrative example 1

```
1 options nodate ls=100 formdlim="_" pageno=1;
2
3 %let n = 5; *sample size;
4 %let m = 20; *subgroup size;
5 %let seed = 1988; *random seed;
6
7 proc iml;only
8 reset nolog;
```

```
n=&n;
m=&m;
bigN = m*n;
x = j(m,n,.);section

call randseed(&seed);
call randgen(x,'GAMMA',1);
print x;
x=shape(x,bigN,1);
/*
    The Expected Value and Variance for the Sampling Distribution
    of the Bootstrap Sample Mean is Calculated below;
*/
B=1e6;
xbar_b = J(B,1,.);
do i = 1 to B;
    u=int(ranuni(J(n,1,0))#bigN+1);
    x_b = x[u,1];
    xbar_b[i,1]=x_b[:,];
end; *end do i;
avg_xbarb=mean(xbar_b);
sigma_xbarb=sqrt(var(xbar_b));
print m n;
print avg_xbarb sigma_xbarb;

free xbar_b;

start tune_L(sim,m,n,vec_x,z0,lambda,L,mu,sigma);

    bigN=m*n;
    runl=j(sim,1,.);
    do i = 1 to sim;
        flag=0;
        k=0;
        z_0=z0;
        do j = 1 to 5000 until (flag=1);
            k=k+1;
            u=int(ranuni(j(n,1,0))#bigN)+1;
```

```
47              UCL_i = mu+L*sigma*sqrt(((lambda)/(2-lambda))*(1-((1-lambda)*
                    *(2*k))));
48              LCL_i = mu-L*sigma*sqrt(((lambda)/(2-lambda))*(1-((1-lambda)*
                    *(2*k))));
49              x_bi = vec_x[u,1];
50              xbar_bi = x_bi[:,];
51              z_i = lambda*xbar_bi+(1-lambda)*z_0;
52              z_0=z_i;
53              if (z_i>=UCL_i) | (z_i<=LCL_i) then flag=1;
54          end; *end do j;The following Saleh et al. () showed that an EWMA
                chart, with estimated values for  and , where  requires about
                 samples of size .
55          runl[i,1]=k;
56      end; *end do i;
57      ARL=runl[:,];
58      return(ARL);
59  finish tune_L;
60
61  *test = tune_L(1e4,&m,&n,x,avg_xbarb,0.05,2.6323,avg_xbarb,sigma_xbarb);
62
63  TOL=0.05; *5% tolerance for error;
64  lambda=0.1;
65  z0=avg_xbarb;
66  sim=1e3;
67  ARL_0=500;
68  ARL_1 = tune_L(sim,&m,&n,x,z0,lambda,2,avg_xbarb,sigma_xbarb);
69  ARL_2 = tune_L(sim,&m,&n,x,z0,lambda,3,avg_xbarb,sigma_xbarb);
70  perc_error1 = (abs(ARL_1-ARL_0)/ARL_0);
71  perc_error2 = (abs(ARL_2-ARL_0)/ARL_0);
72
73  it_matrix = (1 || 2 || ARL_1 || perc_error1) // (2 || 3 || ARL_2 || perc_
        error2);
74  k=3;
75  max_it = 10;
76  flag=0;
77
78  do it = 1 to max_it until (flag=1);
79      L_i_1 = it_matrix[k-1,2];
80      L_i_2 = it_matrix[k-2,2];
```

```
81      ARL_i_1 = it_matrix[k-1,3];
82      ARL_i_2 = it_matrix[k-2,3];
83      L_i = ((ARL_0-ARL_i_1)*(L_i_1-L_i_2)/(ARL_i_1-ARL_i_2))+L_i_1;
84      ARL_i = tune_L(sim,&m,&n,x,z0,lambda,L_i,avg_xbarb,sigma_xbarb);
85      perc_error = (abs(ARL_i - ARL_0))/ARL_0;
86      it_matrix = it_matrix // (k || L_i || ARL_i || perc_error);
87      k=k+1;
88
89      if perc_error <= TOL then flag=1;
90  end; *end do it;
91  print lambda;
92  print it_matrix;
93
94  quit;
95  resetline;
```

A.2.2 SAS code for illustrative example 2

```
options nodate ls=100 formdlim="_" pageno=1;

%let m=20;
%let n=5;
%let seed=123456;

proc iml;
reset nolog;
m=&m;
n=&n;
call randseed(&seed);
ref = j(m,n,.);
call randgen(ref,"GAMMA",1);
bigN=m*n;
vec_X = shape(ref,bigN,1); *performs vec operator on ref sample matrix;
*print vec_X;

*naive bootstrap estimators for muxbar_0 and sigmaxbar_0;

B=1e5; *number of replications;
xmed_b = J(B,1,.);
rep=J(B,1,0);
do i = 1 to B;
     u=int(ranuni(J(n,1,0))#bigN+1);
     x_b = vec_X[u,1];
     *print x_b;
     xmed_b[i,1]=median(x_b);
     *print xbar_b;
end; *end do i;

*print xbar_b;
xmed0_hat = mean(xmed_b);
sigmahat_0 = sqrt(var(xmed_b));
print xmed0_hat sigmahat_0;

*Get Control Limits by using Bootstrap Samples;
sim=10000; *number of simulations;
```

```
38
39 L=2.656; *guess for L;
40 lambda=0.05;
41 runl=j(sim,1,.);
42 do i = 1 to sim;
43      flag=0;
44      k=0;
45      z_0=xmed0_hat;
46      do j = 1 to 5e3 until (flag=1);
47           k=k+1;
48           u=int(ranuni(j(n,1,0))#bigN)+1;
49           UCL_i = xmed0_hat + L*sigmahat_0*sqrt(((lambda)/(2-lambda))*
                 (1-((1-lambda)**(2*k))));
50           LCL_i = xmed0_hat - L*sigmahat_0*sqrt(((lambda)/(2-lambda))*
                 (1-((1-lambda)**(2*k))));
51           x_bi = vec_x[u,1];
52           xmed_bi = median(x_bi);
53           z_i = lambda*xmed_bi + (1-lambda)*z_0;
54           z_0 = z_i;
55           *print xbar_bi LCL_i UCL_i;
56           if (z_i>=UCL_i) | (z_i<=LCL_i) then flag=1;
57      end; *end do j;
58      runl[i,1]=k;
59 end; *end do i;
60 *print runl;
61 arl=runl[:,];
62 print arl;
63
64 quit;
65 resetline;
```

A.3 SAS programs used in Chapter 4

A.3.1 SAS code for finding L for the NBEWMA Chart

A.3.1.1 The $\mathcal{N}(0,1)$ case

```
options ls=100 pageno=1 nodate formdlim="_";

%let m=1000;
%let n=10;
%let seed=1234;
%let sim=10000;
proc iml;
reset nolog;
m=&m;
n=&n;
seed=&seed;
sim=&sim;

/* Distribution */
x=j(m,n,.);
call randseed(seed);
call randgen(x,"NORMAL",0,1);
x=shape(x,m*n,1);
bigN=m*n;

/* Calculates the exact bootstrap mean and variance */
xbarbar = mean(x);
sigmahat_xbar0 = sqrt(sum((x-xbarbar)##2)/(m*n*n));

print xbarbar sigmahat_xbar0;


start generate(n);
     u=ranuni(j(n,1,0));
     x = quantile('T',u,3);
     return(x);
finish generate;
L=2.638;
lambda=0.05;
```

```
35 ARL_0=500;
36 max=ARL_0*10;
37 runl=j(sim,1,.);
38 do i = 1 to sim;
39     k=0;
40     flag=0;
41     z0=xbarbar;
42     do j = 1 to max until (flag=1);
43         k=k+1;
44         LCL = xbarbar-L*sigmahat_xbar0*sqrt(((lambda)/(2-lambda))*(1-((1-
               lambda)**(2*k))));
45         UCL = xbarbar+L*sigmahat_xbar0*sqrt(((lambda)/(2-lambda))*(1-((1-
               lambda)**(2*k))));
46         u = int(ranuni(j(n,1,0))#(bigN-1))+1;
47         x_b = x[u];
48         xbar_b = mean(x_b);
49         z1 = lambda*xbar_b + (1-lambda)*z0;
50         if (z1<=LCL) | (z1>=UCL) then flag=1;
51         z0=z1;
52     end;
53     runl[i]=k;
54 end;
55 ARL=mean(runl);
56 print arl;
57 quit;
58 resetline;
```

A.3.1.2 Necessary amendment to the SAS v9.4 program in Section A.3.1.1 when the underlying process distribution is non-normal

The code required to amend Line 17 of the program listed in Section A.3.1.1 is given below for specific cases.

$t(3)$ distribution

```
call randgen(x,"T",3);
```

$Exp(1)$ distribution

```
call randgen(x,"GAMMA",1);
```

Gamma$(3,1)$ **distribution**

```
call randgen(x,"GAMMA",3);
```

L$(0,1)$ **distribution**

```
*Define Signum Function;
start sgn(x);
     ind1 = -1#(x<0);
     ind2 = (x>0);
     product = ind1 + j(nrow(x),ncol(x),0) + ind2;
     return(product);
finish sgn;
*End Define Signum Function;

*Start Laplace Quantile Function;
start lap_pitc(u,mu,b);
     u=u-0.5;
     X = mu - b#sgn(u)#log(1-2#abs(u));
     return(x);
finish lap_pitc;
*End Laplace Quantile Function;

call randgen(x,"UNIFORM",0,1);
x = lap_pitc(x,0,1);
```

Logistic$(0,1)$ **distribution**

```
/* Logistic Distribution Quantile Function */
start quantile_logistic(p,mu,s);
     q = mu + s#log(p/(1-p));
     return(q);
finish quantile_logistic;

call randgen(x,"Uniform",0,1);
x = quantile_logistic(x,0,1);
```

A.3.2 SAS code used to monitor the IC and OOC performance of the NBEWMA chart

A.3.2.1 $\mathcal{N}(0,1)$ case

```
options ls=100 pageno=1 nodate formdlim="_";

%let m=1000; *number of subgroups;
%let n=10; *sample size;
%let seed=1234;
%let sim=10000; *number of simulations;

proc iml;
reset nolog;
m=&m;
n=&n;
seed=&seed;
sim=&sim;

/* Distribution */
x=j(m,n,.);
call randseed(seed);
call randgen(x,"NORMAL",0,1);
x=shape(x,m*n,1);
bigN=m*n;

/* Calculates the exact bootstrap mean and variance */
xbarbar = mean(x);
sigmahat_xbar0 = sqrt(sum((x-xbarbar)##2)/(m*n*n));

print xbarbar sigmahat_xbar0;

*Generates random of size n from distribution;
start generate(n);
    u=ranuni(j(n,1,0));
    x = quantile('NORMAL',u,0,1);
    return(x);
finish generate;

L=2.638;
```

```
36 lambda=0.05; *smoothing parameter;
37
38 ARL_0=500; *nominal IC ARL_0;
39 max=ARL_0*20;
40 runl=j(sim,1,.);
41 do delta = -1.5 to 1.5 by 0.25;
42     sigma=1;
43     shift = delta*(sigma/sqrt(n));
44     do i = 1 to sim;
45         k=0;
46         flag=0;
47         z0=xbarbar;
48         do j = 1 to max until (flag=1);
49             k=k+1;
50             LCL = xbarbar-L*sigmahat_xbar0*sqrt(((lambda)/(2-lambda))*
                   (1-((1-lambda)**(2*k))));
51             UCL = xbarbar+L*sigmahat_xbar0*sqrt(((lambda)/(2-lambda))*
                   (1-((1-lambda)**(2*k))));
52             x = generate(n);
53             x = x + shift;
54             xbar = mean(x);
55             z1 = lambda*xbar + (1-lambda)*z0;
56             if (z1<=LCL) | (z1>=UCL) then flag=1;
57             z0=z1;
58         end;
59         runl[i]=k;
60     end;
61     ARL=mean(runl);
62     SDRL=sqrt(var(runl));
63     call qntl(q,runl,{0.05, 0.25, 0.5, 0.75, 0.95});
64     output1 = delta || arl || sdrl || (q');
65     output = output // output1;
66 end;
67 print output[colname={"delta" "ARL" "SDRL" "p5" "p25" "p50" "p75" "p95
       "}];
68 create aarl from output[colname={"delta" "ARL" "SDRL" "p5" "p25" "p50" "
       p75" "p95"}];
69 append from output;
70 quit;
```

```
71
72 PROC EXPORT DATA= WORK.Aarl
73 OUTFILE= "c:\Masters\NBEWMA_N01.csv"
74 DBMS=CSV REPLACE;
75 PUTNAMES=YES;
76 RUN;
77
78 resetline;
```

A.3.2.2 Necessary amendment to the SAS v9.4 program in Section A.3.2.1 when the underlying process distribution is non-normal

The code required to amend Line 18 of the program listed in Section A.3.2.1 is given below for specific cases.

$t(3)$ distribution

```
call randgen(x,"T",3);
```

$Exp(1)$ distribution

```
call randgen(x,"GAMMA",1);
```

$Gamma(3,1)$ distribution

```
call randgen(x,"GAMMA",3);
```

$L(0,1)$ distribution

```
*Define Signum Function;
start sgn(x);
    ind1 = -1#(x<0);
    ind2 = (x>0);
    product = ind1 + j(nrow(x),ncol(x),0) + ind2;
    return(product);
finish sgn;
*End Define Signum Function;

*Start Laplace Quantile Function;
start lap_pitc(u,mu,b);
```

```
    u=u-0.5;
    X = mu - b#sgn(u)#log(1-2#abs(u));
    return(x);
finish lap_pitc;
*End Laplace Quantile Function;

call randgen(x,"UNIFORM",0,1);
x = lap_pitc(x,0,1);
```

Logistic$(0,1)$ **distribution**

```
/* Logistic Distribution Quantile Function */
start quantile_logistic(p,mu,s);
    q = mu + s#log(p/(1-p));
    return(q);
finish quantile_logistic;

call randgen(x,"Uniform",0,1);
x = quantile_logistic(x,0,1);
```

The code required to amend Line 31 of the program listed in Section A.3.2.1 is given below for specific cases.

$t(3)$ **distribution**

```
x = TINV(u,3);
```

Exp(1) **distribution**

```
x = quantile('GAMMA',u,1)
```

Gamma$(3,1)$ **distribution**

```
x = quantile('GAMMA',u,3)
```

$L(0,1)$ **distribution**

```
x = lap_pitc(u,0,1);
```

Logistic$(0,1)$ **distribution**

```
x = quantile_logistic(u,0,1);
```

Appendix B

Results used in the mini-dissertation

B.1 Closed-Form Expressions for the Expected Value and Variance of $\bar{X}^*$

We may represent the in-control bootstrap distribution as:

x^*	$P(X^* = x^*)$
X_{11}	$\frac{1}{mn}$
X_{12}	$\frac{1}{mn}$
$\vdots$	$\vdots$
X_{1n}	$\frac{1}{mn}$
X_{21}	$\frac{1}{mn}$
$\vdots$	$\vdots$
X_{2n}	$\frac{1}{mn}$
$\vdots$	$\vdots$
X_{m1}	$\frac{1}{mn}$
X_{m2}	$\frac{1}{mn}$
$\vdots$	$\vdots$
X_{mn}	$\frac{1}{mn}$

We treat the X_{ij}'s as fixed. Now,

$$\mathbb{E}_{\hat{F}_0}\left(X_i^*|\chi\right) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij} = \bar{\bar{X}},$$

where

$$\chi = \left(X_{11}, X_{12}, \ldots, X_{1n}, X_{21}, X_{22}, \ldots, X_{2n}, \ldots, X_{m1}, X_{m2}, \ldots, X_{mn}\right).$$

Using the above result, it follows that

$$\begin{aligned}\mathbb{E}_{\hat{F}_0}\left(\bar{X}^*|\chi\right) &= \mathbb{E}_{\hat{F}_0}\left(\frac{1}{n}\sum_{i=1}^{n} X_i^*|\chi\right)\\ &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{\hat{F}_0}\left(X_i^*|\chi\right)\\ &= \frac{1}{n}\sum_{i=1}^{n}\bar{\bar{X}}\\ &= \bar{\bar{X}}.\end{aligned}$$

The variance of X_i^* is calculated as

$$\begin{aligned}V\left(X_i^*|\chi\right) &= \mathbb{E}_{\hat{F}_0}\left(\left(X_i^* - \bar{\bar{X}}\right)^2|\chi\right)\\ &= \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(X_{ij} - \bar{\bar{X}}\right)^2,\end{aligned}$$

subsequently

$$\begin{aligned}V\left(\bar{X}^*|\chi\right) &= V\left(\frac{1}{n}\sum_{i=1}^{n} X_i^*|\chi\right)\\ &= \frac{1}{n^2}\sum_{i=1}^{n} V\left(X_i^*|\chi\right)\\ &= \frac{V\left(X_i^*|\chi\right)}{n}\\ &= \frac{1}{mn^2}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(X_{ij} - \bar{\bar{X}}\right)^2.\end{aligned}$$

Bibliography

[1] T. Ambartsoumian and D.R. Jeske. Nonparametric CUSUM Control Chart and Their Use in Two-Stage SPC Applications. *Journal of Quality Technology*, 47(3):264, July 2015.

[2] S.M. Bajgier. The Use of Bootstrapping to Construct Limits on Control Charts. In *Proceedings of the Decision Science Institution*, pages 1611–1613, San Diego, CA, 1992.

[3] D. Brook and D.A. Evans. An Approach to the Probability Distribution of CUSUM Run Length. *Biometrika*, 59(3):539–549, December 1972.

[4] G. Capizzi and G. Masarotto. An adaptive exponentially weighted moving average control chart. *Technometrics*, 45:199–207, 2003.

[5] S. Chakraborti, S.W. Human, and M.A. Graham. Phase i statistical process control charts: An overview and some results. *Quality En*, 21(1):52–62, 2009.

[6] S. Chakraborti and M.A. van de Wiel. *A nonparametric control chart based on the Mann-Whitney statistic*, volume Volume 1 of *Collections*, chapter ver, pages 156–172. Institute of Mathematical Statistics, Beachwood, Ohio, USAver, 2008.

[7] S. Chatterjee and P. Qiu. Distribution-Free Cumulative Sum Control Chart Using Bootstrap-Based Control Limits. *The Annals of Applied Statistics*, 3(1):349–369, 2009.

[8] Y.M. Chou, R.L. Mason, and J.C. Young. The control chart for individual observations from a multivariate non-normal distribution. *Communications in Statistics - Theory and Methods*, 30(8-9):1937–1949, 2006.

[9] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.

[10] B. Efron. Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 82(397):171–185, March 1987.

[11] B. Efron and R.J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1994.

[12] F.F. Gan. An optimal design of cumulative sum control chart based on median run length. *Communications in Statistics: Simulation and Computation*, 23(2):485–503, 1994.

[13] A. Gandy and J.T. Kvaløy. Guaranteed Conditional Performance of Control Charts via Bootstrap Methods. *Scandinavian Journal of Statistics*, 40:647–688, 2011.

[14] S.W. Human and M.A. Graham. *Average run-lengths and operating characteristic curves*, volume 1, pages 159–168. New York: John Wiley, 2007.

[15] N. Iranpanah and M. Mokhtari Moghadam. Bootstrap Method for Mean Control Limits of Statistical Process. *Materials & Energy*, 10(1):655–672, 2010.

[16] D.R. Jeske, V. Montes De Oca, W. Bischoff, and M. Marvasti. CUSUM techniques for timeslot sequences with applications to network surveillance. *Computational Statistics and Data Analysis*, 53(12):4332–4344, October 2009.

[17] L.A. Jones, C.W. Champ, and S.E. Rigdon. The Performance of Exponentially Weighted Moving Average Charts with Estimated Parameters. *Technometrics*, 43(2):156–167, 2001.

[18] L.A. Jones and W.H. Woodall. The Performance of Bootstrap Control Charts. *Journal of Quality Technology*, 30(4):362–375, October 1998.

[19] M.B.C. Khoo, V.H. Wong, Wu Z., and P. Castagliola. Optimal designs of the multivariate syntheticchart for monitoring the process mean vectorbased on median run length. *Quality and Re*, 27(8):981–997, December 2011.

[20] S. Knoth and W. Schmid. *Frontiers in Statistical Quality Control*. Springer, 11 edition, 2015.

[21] Y.L. Lio and Chanseok Park. A bootstrap control chart for inverse Gaussian percentiles. *Journal of Statistical Computation and Simulation*, 80(3):287–299, 2010.

[22] L. Liu, X. Zi, J. Zhang, and Z Wang. A Sequential Rank-Based Nonparametric Adaptive EWMA Control Chart. *Communications in Statistics - Simulation and Computation*, 42(4):841–859, December 2013.

[23] Y. Liu and J. Tang. Control Charts for Dependent and Independent Measurements Based on Bootstrap Methods. *Journal of the American Statistical Association*, 91(436):1694–1700, December 1996.

[24] J.F. MacGregor and T.J. Harris. The exponentially weighted moving variance. *Journal of Quality Technology*, 25:106–118, 1993.

[25] M.A. Mahmoud, G.R. Henderson, E.K. Epprecht, and W.H. Woodall. Estimating the standard deviation in quality-control applications. *Journal of Quality Technology*, 42(4):348–357, October 2010.

[26] D. McDonald. A CUSUM Procedure Based on Sequential Ranks. *Naval Research Logistics*, 37:627–646, 1990.

[27] D.C. Montgomery. *Introduction to Statistical Quality Control*. John Wiley & Sons, Inc., 6th edition, 2009.

[28] Daniel J. Mundform, Jay Schaffer, Myoung-Jin Kim, Dale Shaw, Ampai Thongteeraparp, and Pornsin Supawan. Number of Replications Required in Monte Carlo Simulation Studies: A Synthesis of Four Studies. *Journal of Modern Applied Statistical Methods*, 10(1), 2011. Article 4.

[29] C.H. Ng and K.E. Case. Development and evaluation of control charts using exponentially weighted moving averages. *Journal of Quality Technology*, 25:106–118, 1989.

[30] M.D. Nichols and W.J. Padgett. A Bootstrap Control Chart for Weibull Percentiles. *Quality and*, 22:141–151, 2006.

[31] E. Page. Continuous Inspection Schemes. *Biometrika*, 41:100–115, 1954.

[32] Poovich Phaladiganon, Seoung Bum Kim, Victoria C. P. Chen, Jun-Geol Baek, and Sun-Kyoung Park. Bootstrap-based T^2 Multivariate Control Charts. *Communications in Statistics - Simulation and Computation*, 40(5):645–662, 2011.

[33] N.A. Saleh, M.A. Mahmoud, L. Jones-Farmer, I. Zwetsloot, and W.H. Woodall. Another Look at the EWMA Control Chart with Estimated Parameters. *Journal of Quality Technology*, 47(2):363–382, 2015.

[34] T. Seppala, H. Moskowitz, R. Plante, and J. Tang. Statistical Process Control via the Subgroup Bootstrap. *Journal of quality technology*, 27:139–153, 1995.

[35] T.T. Seppala. *Nonparametric statistical process control: Bootstrap control charts.* PhD thesis, Purdue University, 1994.

[36] W.A. Shewhart. Some Applications of Statistical Methods to the Analysis of Physical and Engineering Data. *Bell Labs T*, 3(1):43–87, January 1924.

[37] B.W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall/CRC, London, UK, 1986.

[38] A.W. Wortham and L.J. Ringer. Control via exponential smoothing. *The Logistics Review*, 7:33–40, 1971.

[39] E. Yashchin. Analysis of CUSUM and Other Markov-Type Control Schemes by Using Empirical Distributions. *Technometrics*, 34(1):54–63, February 1992.