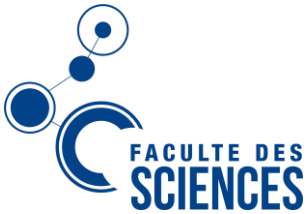

Faculté des sciences

Control Chart Monitoring of Non-Normally Distributed Time Series Data

Auteur : Benjamin Deketelaere
Promoteurs : Rainer von Sachs, Sophie Mathieu
Lecteur : Christian Ritter
Année académique 2019-2020
Master 120 en Statistique Orientation Générale, Finalité Spécialisée
Louvain School of Statistics, Biostatistics and Actuarial sciences (LSBA)

Contents

1 Introduction and objectives

Nowadays, the quality of a product is something we often hear about in the world of industry. The product in question can be a lot of different things. It can be small items such as car components, Coronavirus face masks, and so forth. It can also be network systems such as the internet system or banking system. It is important to check whether the manufactured products conform to their designed requirements because low-quality products can lead to heavy financial losses, incidents or even deaths. In order to check the product conformance to their designed requirements, we want to control the quality of products. To do this, many statistical and management techniques are available in the literature. The central part of these methodologies is called Statistical Process Control (SPC). In SPC, the quality of a production process is evaluated by the measure of several product quality characteristics. These quality characteristics are random variables. Examples of quality characteristics are the length, the width or the weight of a product.

Since the quality characteristics of interest are random variables, they come with some variability even if the production process is stable. This type of variability is mainly caused by certain factors that cannot be controled or are too difficult and too expensive to control. This variability is called "common cause variability" and cannot be reduced without changing the production process itself. In cases where only common cause variability is present in the production process, the process is considered to be in statistical control, or simply in-control (IC). Let us assume that one or several components of the production process become out of order. For instance, this can be the consequence of inadequately adjusted machines, defective materials, improper operation due to workers,.... This would imply that some quality characteristics would have a change in variability or a systematic variation (or both) from the designed requirements. If this happens, many manufactured products would not be able to meet the designed requirements. This type of variation is called "special cause variation". When a production process has special cause variation present, it is considered to be unstable, or out-of-control (OC). The objective is to distinguish special cause variability from common cause variability as soon as possible. To do this, there is already a lot of literature that covers this topic. However, a large part of this literature assumes that the quality characteristic has a normal distribution and that the distribution is identical at each measure. In practice, it is relatively unfrequent that these two assumptions hold. For instance, the quality characteristic can be a count variable like the number of small defects of a product. In this case the distribution of the quality characteristic is non-normal and often close to the Poisson distribution. Often, there is also the problem of dependency on top of that. The central goal of this master thesis is to say more about the use of SPC tools in cases where the data used is non-normal and/or dependent. In more details, we will focus on the calibration of the parameters of the CUSUM chart designed to detect a mean shift.

This master thesis is structured in six different chapters. Chapter 2 introduces the notion of Average Run length (ARL). This notion is central in SPC and will be repeatedly used in later chapters. Chapter 3 is a brief summary of the literature in the case of normal and independent data. This chapter describes Shewhart and CUSUM control charts as well as the link between them. It also explains an adequate way to calibrate the parameters of a CUSUM chart in case both the normality and indepence assumptions hold. In chapter 4 the impact of non-normality on CUSUM charts is studied. The goal of this chapter is to check whether the way of calibrating the chart described in section 3.3 is still adequate for non-normal continuous distributions. First, we study what happens for a symmetric distribution and then we focus on a skewed distribution. The main goal of chapter 5 is to discuss the influence of non-constant shifts on the run length distribution. In other words, this chapter studies the form of the stopping time distribution depending on different shift types (constant, linear, polynomial, sinusoidal). Indeed, it is possible that a mean shift appears gradually and not instantaneously. Chapter 6 is the first which dives into the problem of autocorrelated observations. The chapter begins with an example which illustrates the effects of autocorrelation on the CUSUM chart. After that, section 6.2 proposed a solution which is to apply the chart to the residuals of an ARMA model instead of applying it to the original data. If the ARMA model adequately fits the data, then the residuals are close to white noise (i.i.d. and normal data). If this is the case, then we can use the theory developed earlier to detect a possible mean shift. A whole different solution based on block-bootstrap is presented in chapter 7. In comparison with an ARMA model, the block-bootstrap has the advantage of adapting to a wider variety of correlation structure. The capabilities of the block-bootstrap method are studied in three parts. At first, section 7.1 investigates the effect of bootstrap on the run length distribution when the process observations are independent The second part presents block-bootstrap technique which allows to perform bootstrap on time-series. Finally, the third part establishes a comparison between the results obtained via the Block-bootstrap technique on the one hand, and the parametric approach described in chapter 6 on the other hand.

2 Average Run Length : Basic Concepts in SPC

In the Introduction, we talked about statistical process being in control (IC) or out of control (OC). In this section, we introduce the notion of Average Run Length (ARL). It is a central notion in SPC and it will be our tool to analyze performance of later control charts. ARL is actually defined as the mean number of observations before a control chart signals the process to be OC. Building a control chart is, depending on the data we observed, having a decision rule to determine if the process is IC or OC at a given observation. So for a given n, we would like a rule saying if the process is IC or not. It is a situation which has clear link with the one of a classical hypothesis test. For a hypothesis test, we call type I error the probability to reject the null hypothesis wrongly (i.e the probability to reject H_0 if H_0 is true). Parallel to type I error for hypothesis test, Control charts sometimes detect a process to be OC in a time it is IC. We call ARL_0 (or also IC ARL) the average run length before a control chart signals the process to be OC when the process is actually IC. We also define ARL_1 (or OC ARL) the average run length before a control chart signals the process to be OC since the shift occurrence when the process is truly OC.

We assume that an SPC is IC and that each observation has a probability α of triggering wrongly an alarm for a given control chart. Let Y be the number of observations before the first alarm signal is triggered. Y has a $\text{Geom}(\alpha)$ distribution and its expectation and variance are given by

$$\mathbb{E}[Y] = \frac{1}{\alpha} \coloneqq \text{ARL}_0 \tag{1}$$

and

$$\text{Var}[Y] = \frac{\sqrt{1-\alpha}}{\alpha} \coloneqq \sigma_0^{\text{RL}}. \tag{2}$$

So, from (2), σ_0^{RL} is the notation for the standard deviation of the number of observations before first alert signal. In a situation where α is very close to 0, we have that $\sqrt{1-\alpha} \approx 1$ and so $\text{ARL}_0 \approx \sigma_0^{\text{RL}}$. In SPC, small values for α are commonly used and so the approximation will often be accurate.

Conversely, assume now that the process is OC. We call $1-\beta$ the probability to correctly detect that the process is OC. (So β is the probability to wrongly affirm that the process is IC) Again, the run length follows a geometric distribution but this time the parameter of this distribution is $1-\beta$. It implies that

$$\text{ARL}_1 = \frac{1}{1-\beta} \tag{3}$$

and

$$\sigma_1^{\text{RL}} = \frac{\sqrt{\beta}}{1-\beta} \tag{4}$$

where σ_1^{RL} is the standard deviation of the number of observations before first alert when the process is OC.

We can say that a control chart A is better than a control chart B if both A has a larger ARL_0 than B and A has a lower ARL_1 than B are verified. However, a low type I error leads to low power which means a high type II error. The problem is equivalent here. In most situations, large ARL_0 implies large ARL_1. The reason is that an increase in ARL_0 implies a decrease in α by equation (1). A decrease in α leads to an increase in β. Finally, a rise in β means a decrease in $1-\beta$ which leads to a growth in ARL_1 by equation (3). As well as for a hypothesis tests there is no way to avoid this problem. There is rather a balance to find between high ARL_0 and low ARL_1. Usually, for a hypothesis test, we try to fix a level α and maximise the power among tests with an α-level. SPC imitates this method by fixing an ARL_0 value and try to make the ARL_1 value as small as possible.

	Process is In control	process is Out of control
CC shows process is IC	OK	β-risk
CC shows process is OC	α-risk	OK

Table 1: Control chart risks

3 Classical Control Charts : From Shewhart to CUSUM charts

In the previous chapter, the ARL concept was described. Now that we have this tool, we are interested in a way to detect whether or not a fabrication process is under control. For this, we will introduce control charts and give examples. A control chart can be defined as a chart on which every point correspond to a statistic calculated from successive samples collected in a production.[2] The x-axis of the chart is the sample number and the y-axis is the statistic value. The work is actually about constructing this statistic and evaluate when the values taken by this statistic are too unexpected for the process to be in control.

As it is written in the title, the major goal of this chapter is to display in details, at first Shewhart charts construction, and then how CUSUM charts have been built. For both Shewhart and CUSUM charts, we will focus on detecting a shift in the mean of the quantity of interest. We will assume that the variance of this quantity remains constant over time.

3.1 Shewhart charts

The Average Run Length (ARL), a comparison criteria for control charts has been described in the end of the previous chapter. Now that this tool is available, a first control chart can be constructed. It is the first control chart described in the history of SPC. The idea behind this chart is actually to repeat a hypothesis test at every single time. The test sends an alarm signal when the test declares the difference between the observed mean and a reference value to be significant. Even if many different versions have been proposed with different purposes from almost a century now, the Shewhart chart was and is still very popular in practice nowadays. The name Shewhart chart comes from a paper from 1931 by its first describer Walter Andrew Shewhart. The control chart was used for the first time in the 1920's when Shewhart was working at Bell's lab.

Before entering the topic, it's important to state that the following assumptions are used in this chapter :

1. The data are univariate.

2. When the process is IC, the observations are assumed i.i.d and following a $\mathcal{N}(\mu_0, \sigma^2)$ where μ_0 and σ are known.

3. When the process goes OC, what actually changes is the mean of the IC distribution. It means that, at a certain time point, the mean of the process gets shifted by δ. It implies the OC distribution becomes $\mathcal{N}(\mu_1, \sigma^2)$ where $\mu_1 = \mu_0 + \delta$.

3.1.1 Description of the chart

The idea behind the Shewhart chart is to repeatedly execute the following hypothesis test :

$$H_0 \ : \ \mu_i = \mu_0 \qquad vs \qquad H_1 \ : \ \mu_i \neq \mu_0 \tag{5}$$

where μ_i is the observed mean of the process at time i and μ_0 is the expected mean. To perform the test (5), a random sample of size m is collected at each time i. This sample $\{X_{ij} \mid j \in 1, \ldots, m\}$ allows to estimate

$$\overline{X_i} = \frac{1}{m} \sum_{j=1}^{m} X_{ij}$$

the mean of the process at each time i. It is known that the test statistic of (5) is given by

$$Z_i = \sqrt{m} \frac{\overline{X}_i - \mu_0}{\sigma}. \tag{6}$$

This statistic follows a standard normal distribution under H_0. If the test (5) is performed with a significance level α, then H_0 should be rejected if the observed value Z_i^* of Z_i satisfies

$$|Z_i^*| > z_{1-\frac{\alpha}{2}} \tag{7}$$

where $z_{1-\alpha/2}$ denotes the $(1-\alpha/2)$-quantile of the standard normal distribution. In conclusion, we can say that $\overline{X_i}$ provides signicant evidence of declaring the process OC at time i if

$$\overline{X}_i > \mu_0 + z_{1-\alpha/2} \frac{\sigma}{\sqrt{m}} \qquad \text{or if} \qquad \overline{X}_i < \mu_0 - z_{1-\alpha/2} \frac{\sigma}{\sqrt{m}}. \tag{8}$$

Until now, we assumed μ_0 and σ^2 to be known but in practice, it is very unusual. We now focus on a way to estimate those two parameters. For μ_0, it is natural to take as estimator the grand sample mean. By grand sample mean, we mean it to be the mean of all batch means. It can be written as

$$\widehat{\mu_0} = \overline{\overline{X}} = \frac{1}{n} \sum_{i=1}^{n} \overline{X_i} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} X_{ij}. \tag{9}$$

An estimation for the standard deviation of a sample can be obtained easily. For the sample at time i $\{X_{ij} \mid j \in 1, \ldots, m\}$ of size m the standard deviation is given by

$$s_i = \sqrt{\frac{1}{m-1} \sum_{j=1}^{m} (X_{ij} - \overline{X_i})^2}. \tag{10}$$

Then, we consider

$$s = \frac{1}{n} \sum_{i=1}^{n} s_i,$$

the mean of the standard deviation of each sample. s is an estimator for σ the global standard deviation. This estimator is biased but a correction can be found in Kenney and Keeping [3].

We now have a way to construct estimators for μ_0 and σ. However these estimators will solely be valid if the process is under control long enough when data are collected. Also, it seems clear that the values for $\widehat{\mu_0}$ and $\widehat{\sigma}$ should be checked and reupdated continuously. The Shewhart chart (8) for μ_0 and σ known can now be adapted to a case where they are not. For this, μ_0 and σ in equation (8), should be replaced by $\widehat{\mu_0}$ and $\widehat{\sigma}$, their estimators. Thus, if μ_0 and σ are unknown, we can say that the i-th sample $\{X_{ij} \mid j \in 1, \ldots, m\}$ provides significant evidence of declaring the process OC at time i if

$$\overline{X}_i > \widehat{\mu_0} + z_{1-\alpha/2}\frac{\widehat{\sigma}}{\sqrt{m}} \qquad \text{or if} \qquad \overline{X}_i < \widehat{\mu_0} - z_{1-\alpha/2}\frac{\widehat{\sigma}}{\sqrt{m}}. \tag{11}$$

In summary, a Shewhart chart for monitoring the mean of a process is a plot of $\overline{X_i}$ over time plus two horizontal lines describing the control limits. These control limits are given by $\widehat{\mu_0} \pm z_{1-\alpha/2}\frac{\widehat{\sigma}}{\sqrt{m}}$. The control chart indicates that the process is OC when a value of $\overline{X_i}$ goes out of the control limits.

The decision rule given by (11) evaluates the state of control at time i. This decision is based only on the data collected at this time and the charting statistic at time $i-1$ doesn't have any influence in the decision provided by the chart at time i. It is logical since a Shewhart chart is actually a classical hypothesis test $\mu = \mu_0$ versus $\mu \neq \mu_0$ for a given value α. For this reason, Shewhart charts are not very efficient in detecting a small and persistent shift in the mean. The following example is an illustration of this problem :

3.1.2 Example

Example 1. *Assume that 20 samples of size 5 each are generated from the $N(0,1)$ distribution, and another 30 samples of size 5 each are generated from the $N(0.3,1)$ distribution. These samples can be regarded as observed data collected at 50 consecutive time points from a production process. The IC distribution of the process is N(0,1) and the process has a mean shift of size 0.3. Figure 1 shows the Shewhart chart constructed from the 50 samples. In the plot, the first 20 sample means are denoted by solid dots. The last 30 sample means are denoted by circles. The lower and upper control limits of the chart are set to $-3/\sqrt{5}$ and $3/\sqrt{5} \approx 1.34$ thanks to equation* (8).

From the plot, it can be seen that the mean shift at the 20th time point is not detected by the chart. This example illustrates that Shewhart charts are not efficient in detecting a small mean shift.

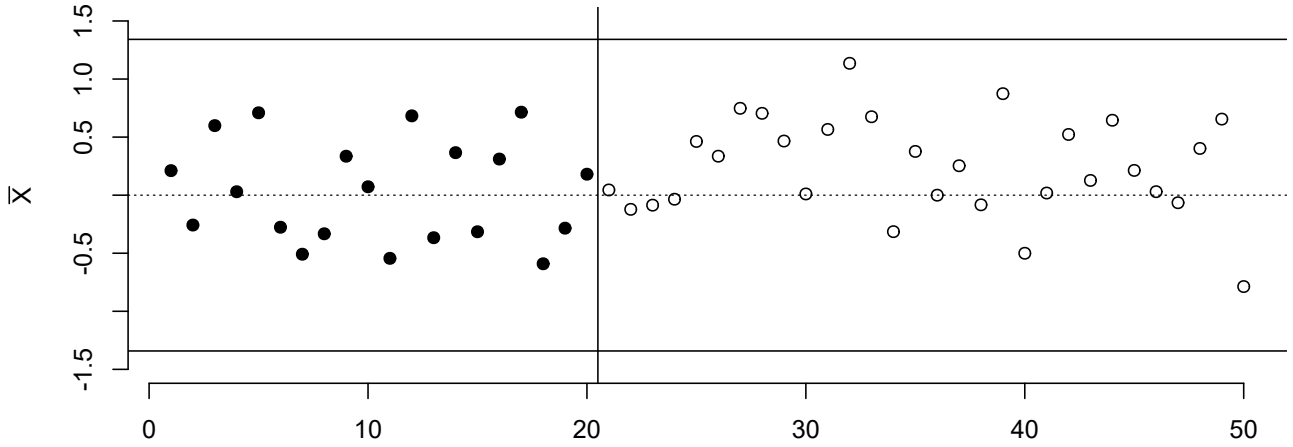

Figure 1: Shewhart chart constructed from all 50 samples with the control limits computed by the formulas (8). The first 20 sample means are denoted by solid dots, and the second 30 sample means are denoted by circles. The solid horizontal lines denote the upper and lower control limits.

3.2 CUSUM charts

In the previous subsection, Shewhart charts were described. Shewhart charts were constructed as classical hypothesis tests on the mean of a sample. Consequently, they didn't use all available information in order to make their decisions. In this section, a new chart that includes the history of the data is used. The solution presented here is the CUmulative SUM control chart (CUSUM chart). E. S. Page, from Cambridge University, was the first to propose them in 1954. In this section, two different forms of the CUSUM chart are described. Each form will come with an example. Those two forms are equivalent.

We recall that we are still working under the three assumptions enumerated in the beginning of the section on Shewhart charts. Shewhart charts were described for batched data. Here, we will describe a CUSUM chart that deals with individual data. CUSUM charts can also be applied on batched data. However, it is easier to describe and more efficient to apply on individual data. It is more efficient in the sense that it requires less observations to detect a given shift with a given level of confidence.

3.2.1 Description of the chart

Assume $X_1, X_2, \ldots$ are individual observations of a production process, μ_0 to be its mean IC, μ_1 its mean when the process is OC and $\delta = \mu_1 - \mu_0$ the shift in question. The cumulative sum of the observations is a number that contains

information about the history of the process. It also allows to see how the process has moved away from the IC mean μ_0. Consequently, we consider the statistic

$$C_n \;=\; \sum_{i=1}^{n} (X_i) - n\mu_0 \;=\; \sum_{i=1}^{n} (X_i - \mu_0)\,. \tag{12}$$

which evaluates how much the current process mean is different from μ_0. Actually, C_n is a cumulative sum of the deviations of the observations from μ_0 and is a summation of i.i.d. zero-mean random variables. From (12), it is easy to see that

$$C_n = C_{n-1} + (X_n - \mu_0), \tag{13}$$

where $C_0 = 0$. On the one hand, if the process is in control, since C_n is a random walk, we can show that C_n has the following distribution

$$C_n \;\sim\; \mathrm{N}(0, n\sigma^2). \tag{14}$$

On the other hand, if the process is OC, then there exists $1 \leq \tau \leq n$ such that the process underwent a mean shift δ at time τ. In this case, the mean of C_n depends linearly on δ. The distribution of C_n is given by, for $n \geq \tau$,

$$C_n \;\sim\; \mathrm{N}\left((n - \tau + 1)\delta, n\sigma^2\right). \tag{15}$$

From (14) and (15), we can see that the mean of C_n is 0 before the mean shift and deviates with a slope δ after the mean shift occurred. Sadly, the variance of C_n increases with n and this makes the identification of a linear trend in the mean of C_n challenging.

To avoid this problem, Page [8] proposed the following statistic to detect an upward mean shift

$$P_n^+ = P_{n-1}^+ + (X_n - \mu_0) - k, \qquad \text{for } n \geq 1, \tag{16}$$

where $P_0 = 0$, and $k > 0$ is a parameter that is named the *reference value*. Page wrote that an upward shift in the mean is detected if

$$P_n^+ - \min_{1 \leq m < n} P_m^+ > h \tag{17}$$

where h is another fixed constant called the *control limit*. Using the procedure described by (16) and (17) is equivalent to construct

$$C_n^+ = \max\left(0, C_{n-1}^+ + (X_n - \mu_0) - k\right), \qquad \text{for } n \geq 1, \tag{18}$$

where $C_0^+ = 0$ and saying that an upward shift in the mean is detected when

$$C_n^+ > h. \tag{19}$$

3.2.2 Example

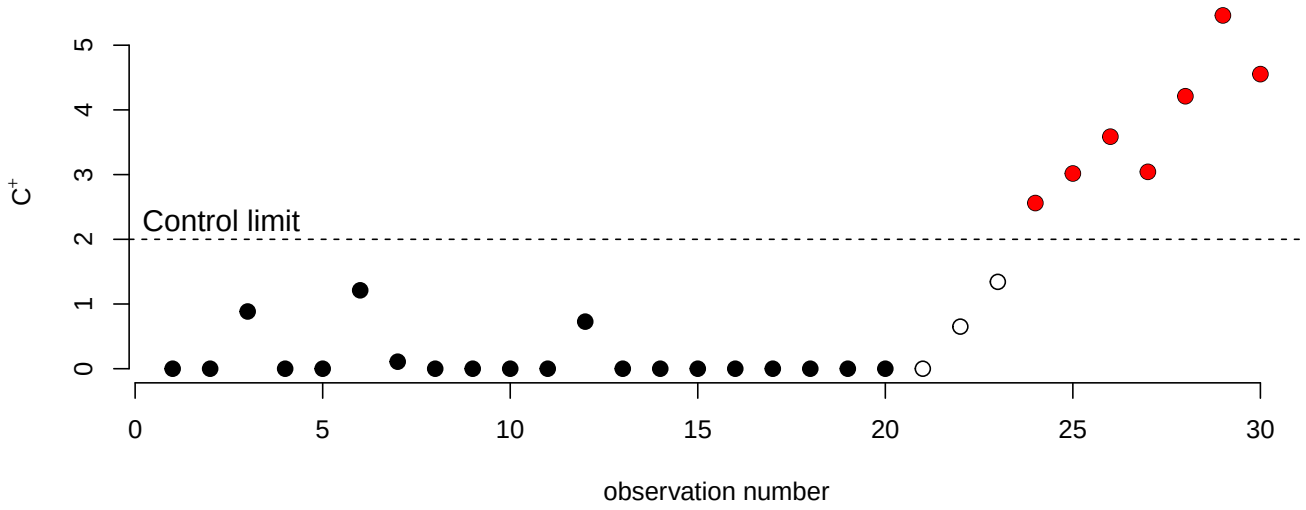

Figure 2: Example of an upward CUSUM chart (see eq (18)-(19)). All black solid dots are values for C_n^+ when the process was IC. During this time, data were sampled from a standard normal distribution. The points become empty after a shift in the mean occurred. In this example the shift is of size $\delta = 0.25$. The points for C_n^+ are drawn in red after an alarm signal was sent. The value chosen for k is 0.5 and the control limit h is 2.

Two-sided CUSUM chart

The control chart given by (18) and (19) is unable of detecting any negative shift in the mean. However we have that detecting a shift of size $-\delta$ in the original characteristic X is equivalent to detecting an upward shift of size δ in -X. By this equivalence, we can construct another CUSUM chart with the charting statistic

$$C_n^- = \min\left(0\ ,\ C_{n-1}^- + (X_n - \mu_0) + k\right) \qquad \text{for } n \geq 1, \tag{20}$$

with $C_0^- = 0$. This chart gives a signal of a downward shift if

$$C_n^- \ < \ -h. \tag{21}$$

Both the CUSUM charts (18) - (19) and (20) - (21) are one-sided control charts. If we want to detect a mean shift with the sign of δ not specified, we can consider to merge both one-sided charts. In this case, an alarm signal will be sent if (19) or (21) hold. Both one-sided control charts described above are working for individual observations. If the data comes into groups of size m, then both CUSUM charts can be constructed the same way.

3.2.3 V-mask Form

The form of the CUSUM chart defined by (18) and (19) and by (20) and (21) is called the *decision-interval* form. However, there exists another popular form of the CUSUM chart which is called the "V-mask form". The V-mask form of CUSUM chart is useful to describe because it gives a visual intuition of the chart parameters k and h.

From (14) and (15), we can draw a line representing the expected value for C_n. This line should be flat until time τ because, the process is IC before τ. After time τ, the expected value for C_n begins to deviate linearly with a slope δ. It is difficult to detect the time the expected value begins to deviate because (14) and (15) show that the variance of C_n is $n\sigma^2$ and increases with n. The V-mask form of the CUSUM chart is actually based on the statistic C_n. Intuitively, if a positive mean shift of size δ occurs at τ, then, when $n \geq \tau$, all C_n should be above a half-line starting at $(\tau, C_\tau - h)$ with a slope k (with $h > 0$ and $k < \delta$). The inclusion of the constant h and the slope k is for accommodating the increasing variability of C_n.

Since τ is not known, the half-line at $(\tau, C_\tau - h)$ and slope k cannot be used. Instead, we consider the half-line starting at $(n, C_n - h)$ going backwards with a slope k. After that, we construct the symmetric half-line of slope $-k$ starting at $(n, C_n + h)$. Those two lines form an horizontal V-shaped "mask". The last observation is located at a distance h of both sides of the V. The opening angle of the V is determined by the reference value k. The process is declared IC at time n if all the previous charting statistics C_{n-i} with $i = 1, \ldots, n-1$ are located inside the branches of the V. An example of a V-mask can be found on figure 3.

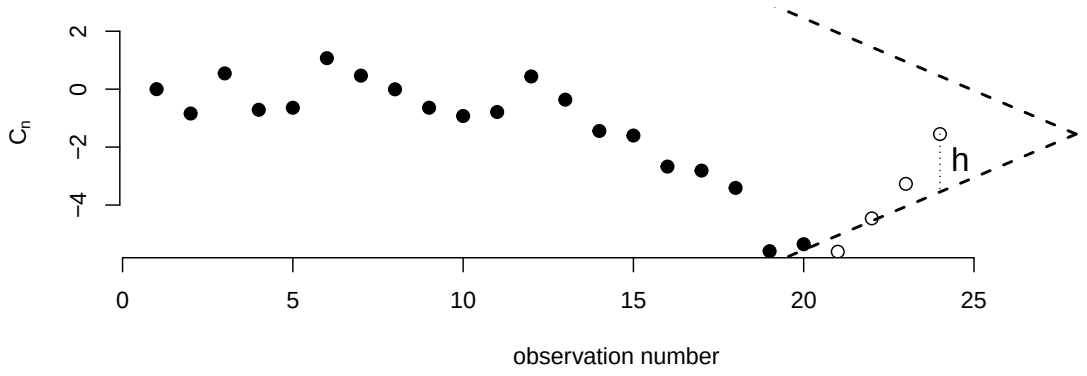

Figure 3: Example of the V-mask form of a CUSUM chart. This graph depicts the evolution of C_n over time. All black solid dots are values for C_n when the process was IC. The points become empty after a mean shift occurred. The diagonal dotted lines give the shape of the V-mask. The slope of the inferior dotted line is given by the chart parameter k. The slope of the upper branch of the V is $-k$. The value taken by h is the distance from the last value for the charting statistic to the branches of the V.

Example 3. *This is an example of the V-mask form of a CUSUM chart. The data are reused from example 2. From time 1 to 20, data were sampled from a standard normal distribution. Between time 20 and 21 a mean shift of size* $\delta = 0.25$ *is applied and data are sampled from a N(0.25,1) distribution. The mask is drawn at time 24. In this example, the values used for* k *and* h *are* 0.5 *and* 2*, respetively. We can see that some values of* C_n *are outside the mask. At time 24, the V-mask form should send an alarm signal.*

From figure 3 we can easily see that the smaller the opening angle of the mask (the smaller k), the quicker a little shift can be detected. Of course, a decrease in k implies an increase in the α-risk. The same conclusion holds for h. In general, the time at which the mean shift occurred and the size of the shift as well are not known. The V-mask form can provide an estimation for τ and δ. If there are several points out of the mask, $\widehat{\tau}$ is given by the time at which the charting statistic is the farthest from the inside of the V. In example 3, $(21, C_{21})$ is the only point outside the mask and $\widehat{\tau}$=21. Based on our estimation for τ and since the expected increase(decrease) in C_n is linear, a natural estimator for δ is

$$\widehat{\delta} = \frac{C_n - C_{\widehat{\tau}}}{n - \widehat{\tau}}. \tag{22}$$

3.3 How to calibrate the parameters of the CUSUM chart

In the previous sections, Shewhart and CUSUM charts were described. Also, for both charts, an example was given. In these examples, the values for the chart parameters were chosen arbitrarily. The next section discusses a way to calibrate the parameters of a Shewhart or a CUSUM chart. As for the two previous sections, the calibration is made under the assumption that the data are independent and sampled from a standard normal distribution.

3.3.1 Control limit (h)

The value given for h depends on the value fixed for k. In this section, we assume that a value was already determined for k. The idea for calibrating the control limit is to adjust h such that the control chart attains a pre-specified false alarm rate. This false alarm rate is characterised by a pre-established ARL_0 value.

We note that, for a value of $k > 0$, there exists a single h such that a CUSUM chart with parameters k and h has the ARL_0 value previously settled. In practice, the control limit h value can be calibrated by an algorithm with which the reference ARL_0 value is reached to a certain accuracy. The algorithm in question is described below.

Algorithm 1 Pseudo-code how to search for the value of h

Choose A_0 a reference ARL_0 and fix k.
Let $[h_L, h_U]$ be the interval frow which h is searched.
Let ϵ be a small number denoting the required estimation accuracy.
Fix n^*, the maximum number of iterations.
for i in $1, \ldots, n^*$ **do**
  $h \leftarrow \frac{h_L + h_U}{2}$
  Compute the ARL_0 value with k and h.
  if the ARL_0 value computed is included in $[A_0 - \epsilon, A_0 + \epsilon]$ **then**
    the algorithm is stopped and h is the result of the search.
  else
    update h_L or h_U : $\begin{cases} h_U \leftarrow \frac{h_L + h_U}{2} & if \quad ARL_0(h) > A_0 + \epsilon \\ h_L \leftarrow \frac{h_L + h_U}{2} & if \quad ARL_0(h) < A_0 - \epsilon \end{cases}$
  end if
end for

The algorithm described above works following an iterative bissection method. It means that h is searched in an interval which gets cut in two at every iteration. In this algorithm, A_0 is the pre-specified ARL_0 value, $[h_L, h_U]$ be the interval from which h is searched, $\epsilon > 0$ is a small number corresponding to the required accuracy. The maximum number of iteration is given by n^*. If the above algorithm doesn't stop before or at the n^*-th iteration, the searched value for h is the value defined in the n^*-th iteration.

3.3.2 Reference value (k)

In section 3.3.1, we described an algorithm to calibrate the control limit. This algorithm assumes that a value was previously fixed for the reference value k. From the distribution of C_n in (14) - (15), we saw that the variance increased over the number of observation being summed. In equation (16), k is introduced in order to counter this increase in variability.

By looking at equation (18)-(19), we can see that an observation will never give evidence of a mean shift if the difference between this observation and μ_0 is not larger than k. If for an observation X_i, we have that $X_i - \mu_0$ is positive but smaller than k, then there are two possible scenarios. First scenario is the previous charting statistic C^+_{n-1} was already 0 and in this case C^+_n remains 0. If $C^+_{n-1} \neq 0$ we still know that $C^+_{n-1} \leq h$. Otherwise, an alarm signal would have been sent before. In this case we also know that $h \geq C^+_{n-1} \geq C^+_n$.

Section 3.3.1 showed that, for a given k, one can find h such that the chart attains a pre-specified threshold. The question that arises now is how to establish the value for k. The objective is to find the couple of parameters (k, h) that minimises ARL_1 if the mean of the process is shifted by δ. This is done in the following way : At first, several values are selected for k. After that, corresponding values for h are computed such that ARL_0 is 200. These computations for h are made thanks to

algorithm 1. Then, by using Monte-Carlo simulation, we can calculate the average time needed to detect a shift of δ in the mean. This operation is repeated for a range of values of δ. Finally, ARL curves are plotted together and we look at the combination (k, h) that minimizes ARL_1. Figure 4 shows the evolution of ARL_1 in function of δ for different values of k.

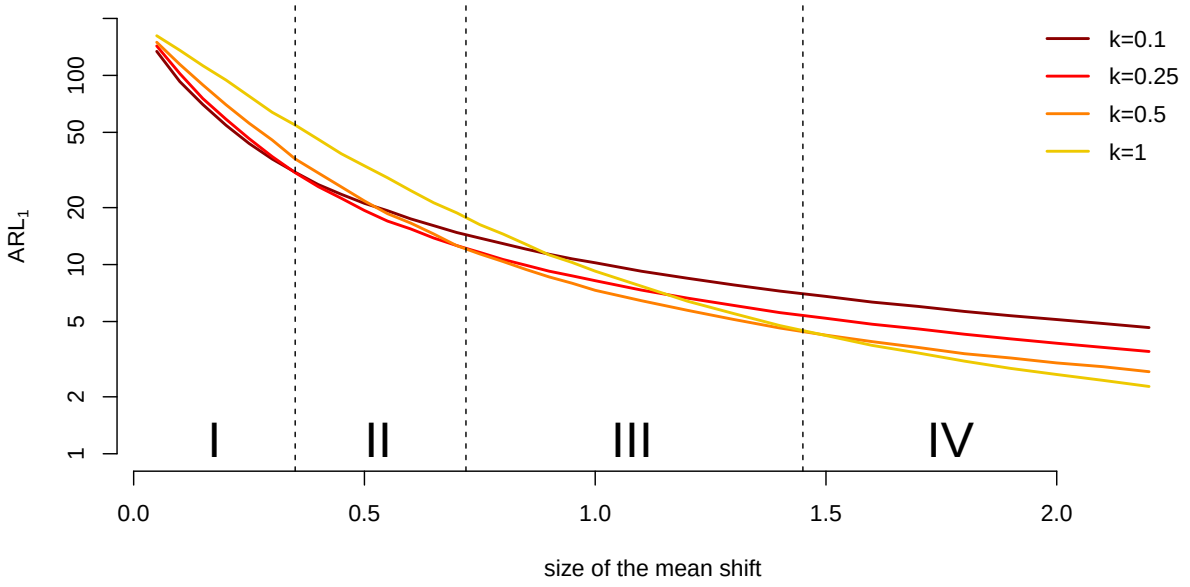

Figure 4: ARL_1 vs δ for different values of k. For an ARL_0 value of 200, given $k = 0.1,\ 0.25,\ 0.5$ and 1, h is found to be $8.520,\ 5.597,\ 3.502$ and 1.875, respectively. To better demonstrate the difference among values of k, the y-axis is in log-scale. For each combination (k, h) and each value of δ, ARL_1 is computed by averaging the run length of 50000 simulations. Regions I, II, III and IV are determined such that $k = 0.1$ minimizes ARL_1 in region I, $k = 0.25$ is minimum in region II and so forth.

In region I of figure 4, the dark red curve of $k = 0.1$ takes the lowest values. It means that for δ going from 0 to approximately 0.35, a CUSUM chart with parameters $(k, h) = (0.1, 8.52)$ will be quicker to detect a shift than taking a k value of $0.25,\ 0.5$ or 1. In region II $(0.35 < \delta < 0.725)$, the combination with $k = 0.25$ is the best of the four combinations compared here. For δ between 0.725 and 1.45, $k = 0.5$ gives the smallest ARL_1 value. In region IV, the yellow curve suggests that 1 is an appropriate choice for k.

When looking at figure 4, one realises that, for independent and normal data, the choice $k = \delta/2$ is more or less optimal. We can also see from the figure that, even if it is not optimal, a choice of k between $\delta/4$ and δ yields acceptable performance. Conveniently, $k = \delta/2$ is used in the literature. Page [8] suggested it and Moustakides [7] proved that it is optimal. In the next chapter, we will investigate if this choice $k = \delta/2$ remains adequate when the normality assumption is not satisfied.

3.4 Conclusion

In this chapter, Shewhart and CUSUM chart were described. At first, we focussed on the Shewhart chart. The Shewhart chart is actually a repetition of a classical hypothesis test on the mean. Example 1 showed how a Shewhart chart works and bringed out that the Shewhart chart is not adequate if we want to detect a small mean shift. After that, two equivalent forms of the CUSUM chart were described. There is one important difference between a Shewhart chart and a CUSUM chart. On the one hand, a Shewhart chart uses only the data observed at the current time point for detecting variation caused by special causes and it ignores all data that are observed earlier. On the other hand, CUSUM charts use all available data observed at earlier and current time points. For this reason, CUSUM charts are more appropriate if one have to deal with small mean shifts. Section 3.3 explains how we can calibrate the parameter of the CUSUM chart for normallly distributed and independent data. The reference value k should be chosen as $\delta/2$ where δ is the size of the shift we want to detect. The control limit h should be calibrated in order to attain a pre-specified false alarm rate. To do this, the searching algorithm described in algorithm 1 should be used.

We conclude this chapter by showing that the Shewhart chart is a particular case of the CUSUM chart. Indeed, if α is given by $1/\mathrm{ARL}_0$, then, for a group of data of size m, the Shewhart chart provides a signal of an upward mean shift at time i if

$$\sqrt{m}\frac{\overline{X} - \mu_0}{\sigma} > z_{1-\alpha/2} \tag{23}$$

is satisfied. The condition (23) is equivalent to the condition

$$\max\left(0, S_{n-1}^{+} + \sqrt{m}\frac{\overline{X} - \mu_0}{\sigma} - z_{1-\alpha/2}\right) > 0 \tag{24}$$

where $S_0^+ = 0$. So from (18) and (19), we can conclude that a classical Shewhart chart is a particular case of a CUSUM chart with $k = z_{1-\alpha/2}$ and $h = 0$. [9] This particular case of the CUSUM chart doesn't use a stocking memory system. Actually it's a case where the parameters of the CUSUM chart are badly calibrated.

4 CUSUM Charts for Non-Normaly Distributed Data

In the previous chapter, classical CUSUM charts were described. The theorical necessary assumptions were

1. the observations are independent and identically distributed.
2. the data are drawn from a normal distribution where the parameters for mean and variance are known.

Chapters 6 and 7 will investigate solutions in the case of dependent data. In this chapter, we discuss cases when process observations are non-normal. In particular, we focus on situations where data are drawn from known parametric distributions. Without loss of generalities, we will work with the upward CUSUM chart

$$C_n^+ = \max\left(0, C_{n-1}^+ + (X_n - \mu_0) - k\right), \qquad \text{for } n \geq 1,$$

where $C_0^+ = 0$ and saying that an upward shift in the mean is detected when

$$C_n^+ > h.$$

The only difference with chapter 3 is that the distribution of X is no longer normal. Section 3.3 showed that an appropriate calibration for the chart parameters was $k = \delta/2$ and adapting h in order to obtain a pre-specified ARL_0 threshold. The main objective of this section is to see if the rule $k = \delta/2$ is still valid for other distributions.

To do this, this chapter is separate in two parts. The first part studies the case of a symmetric distribution while the second part studies the case of a skewed distribution. So that the results are comparable with those obtained for the normal distribution, we start by standardizing the generated data. It means that, for each studied distribution, $\mu_0 = 0$ in the definition of C_n^+.

4.1 Student's t distribution

Since the student's t distribution is already centered, the standardization step solely consists in dividing by the standard deviation. The variance of a student distribution with p degree of freedom is undetermined if $p \leq 1$, infinite if $1 < p \leq 2$ and is $p/p - 2$ if $p > 2$. In this subsection, data are generated assuming that the distribution has 3 degrees of freedom.

So the CUSUM chart used in this subsection is

$$C_n^+ = \max\left(0, C_{n-1}^+ + X_n - k\right), \qquad \text{for } n \geq 1, \tag{25}$$

where $C_0^+ = 0$ and $X_i \sim t_3$. An upward shift in the mean is detected when

$$C_n^+ > h. \tag{26}$$

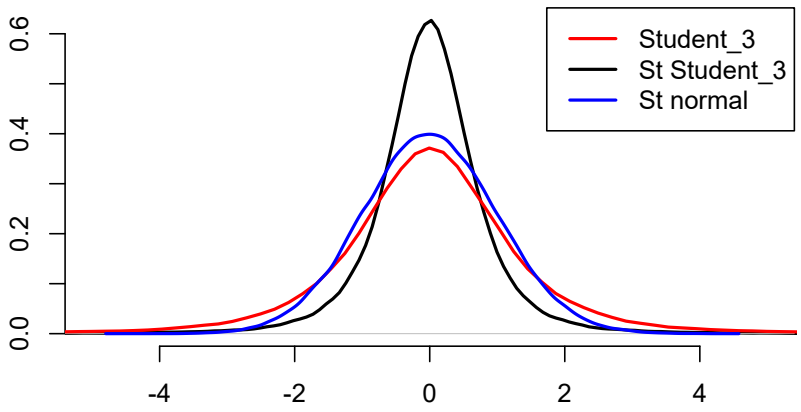

Figure 5: Density of the standardized student's t distribution. The red curve shows the distribution of the t_3 distribition. The blue curve is the p.d.f. of the classical gaussian distribution.

The hyper-parameters of the chart (25) - (26) are tuned following the same procedure as for the normal data on figure 4. At first, a reference value for ARL_0 is fixed. Then, a value for k is chosen. Finally, h is adapted in order to obtain the reference ARL_0 value. This is done by using algorithm 1 described at page 11.

The reference ARL_0 used is 200 and we compare ARL curves for $k = 0.1,\ 0.25,\ 0.5$ and 1 again. The corresponding h are 7.5, 5.05, 3.58 and 2.52, respectively. In comparison with the normal distribution, [1] h_{t_3} is smaller than h_{norm} for small k values (when k is 0.1 and 0.25). The opposite is observed for large k values (when $k = 0.5$ or $k = 1$).

We evaluate the performance of the chart by computing ARL_1 after a shift of δ in the mean is detected. This operation is repeated for a range of values of δ. Figure 6 shows the evolution of the average detection time in function of the size of the shift δ for different values of k.

In the region I of figure 6, the black curve of $k = 0.1$ has the lowest values. It means that for δ going from 0 to approximately 0.45, a CUSUM chart with parameters $(k, h) = (0.1, 7.5)$ will be quicker to detect a shift. In the region II ($0.45 < \delta < 1.1$), $k = 0.25$ gives the fastest detection speed. For shifts going from 1.1 to 2.3, $k = 0.5$ is the quicker and $k = 1$ is the better choice otherwise.

In comparison with figure 4, we observe that regions I, II and III have become larger. Region I has a length of roughly 0.45 vs 0.35 for normal data. Region II measures 0.65 on figure 6 vs 0.375 on figure 4. The size of region III is approximately 1.2 for student data and was 0.775 for normal data.

[1] h for the normal distribution was respectively 8.520, 5.597, 3.502 and 1.875.

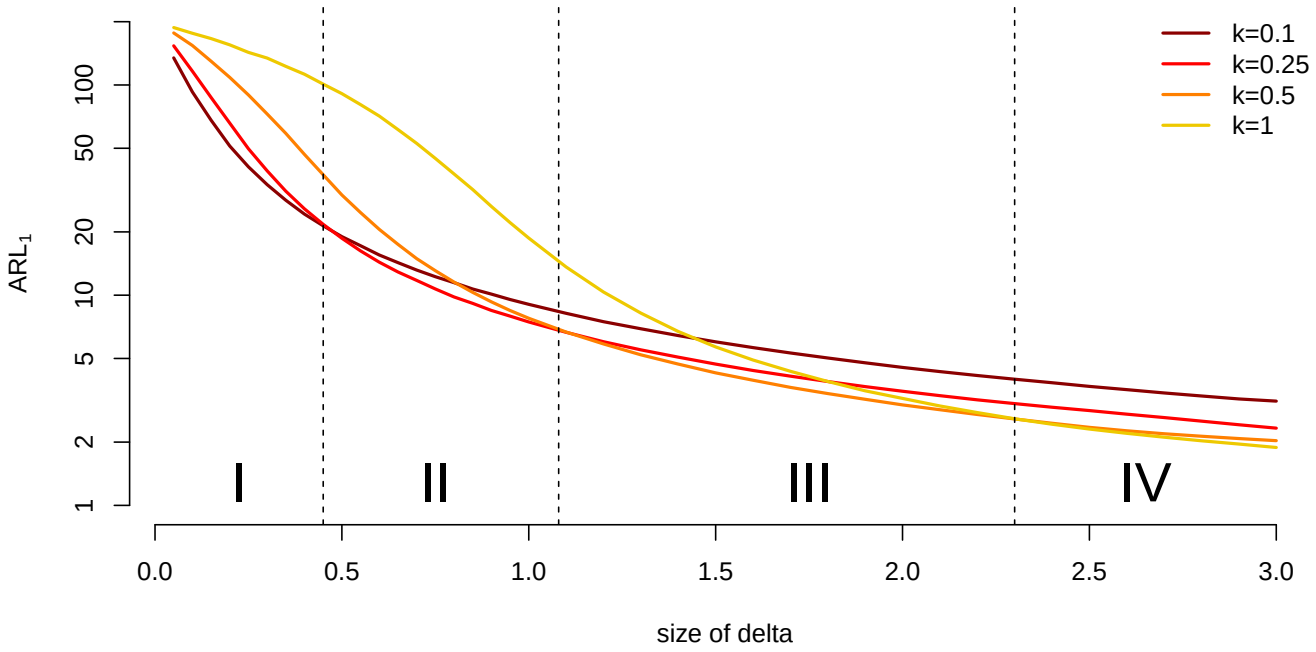

Figure 6: ARL_1 estimated in function of δ for different values of k for a student distribution with 3 degrees of freedom. The reference value for ARL_0 is 200. To better demonstrate the difference among values of k, the y-axis is in log-scale. Regions I, II, III and IV are determined such that $k = 0.1$ minimizes ARL_1 in region I, $k = 0.25$ is minimum in region II and so forth. Regions I, II, III and IV are determined such that $k = 0.1$ minimizes ARL_1 in region I, $k = 0.25$ is minimum in region II and so forth.

We wonder if the choice of $k = \delta/2$ used for normal data is still appropriate for student's t data. We look at the numerical ARL_1 values estimated in figure 6 for shifts of size $\delta = 2k$. These values are given in the table below.

	δ =0.2	δ =0.5	δ =1	δ =2
$k = 0.1$	51.22	18.99	9.05	4.53
$k = 0.25$	65.82	18.65	7.46	3.49
$k = 0.5$	108.61	29.97	7.77	3.01
$k = 1$	155.16	90.93	18.69	3.23

Table 2: Numerical values for ARL_1 estimated when $\delta = 2k$. These values are given by averaging the run length of 10 000 Monte Carlo simulations. Because of that, there is a small variability in these values.

- When $\delta = 0.2$, the parameter $k = 0.1$ yields, by far, the better result.
- When $\delta = 0.5$, $k = 0.25$ gives the best performance and $k = 0.1$ is nearly as quick as $k = 0.25$.

- When $\delta = 1$, the best result is not provided by $k = 0.5$ but by $k = 0.25$. However, they both give close ARL_1 values.
- When $\delta = 2$, $k = 0.5$ minimizes the ARL_1 value among the 4 combinations of parameters. The 3 other values for k yield nearly similar values.

Looking at these results, we discover that choosing $k = \delta/2$ doesn't always produce the best result. Furthermore, we guess that the value for k which minimize the ARL_1 should be taken smaller than $\delta/2$.

By looking at the numbers on table 2, we realise that $k = \delta/2$ gives results which are close to the best. So, even if $k = \delta/2$ does not minimize the detection time, it yields acceptable performance. In this subsection, we only consider results when data are generated from a student's t distribution with 3 degrees of freedom. It is a known result that a student distribution converges to a normal distribution when the number of degrees tends to infinity. Since $k = \delta/2$ provides good performances for data drawn from a normal distribution, it seems reasonnable to assume that the performance of the choice $k = \delta/2$ will improve as long as the number of degrees of freedom will grow. We conclude this subsection by saying that $k = \delta/2$ can be used for student data as long as the number of degrees of freedom of the student distribution is larger than 3.

4.2 Chi-square distribution

In this subsection, we repeat the operations performed previously and we apply them for data drawn from a chi-squared distribution. A χ^2_p distribution has a mean p and a standard deviation $\sqrt{2p}$. The p.d.f. of standardized χ^2 are represented on figure 7.

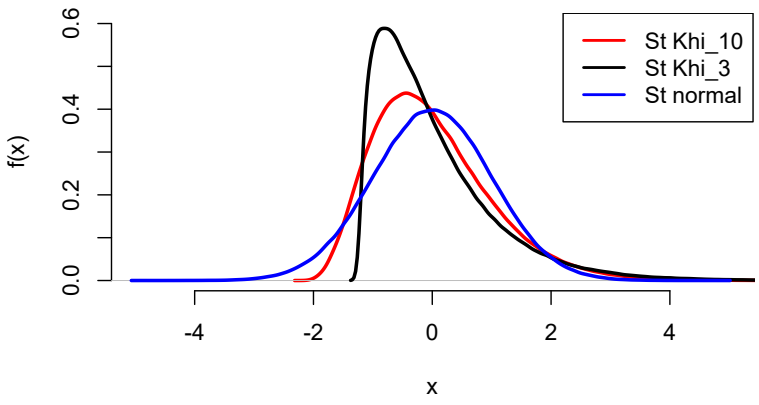

Figure 7: Probability density function of the standardized χ^2_3, χ^2_{10} and normal distribution.

We can see that the standardized χ^2 is a skewed distribution. In particular it has been derived that the skewness of χ^2_p is $\sqrt{8/p}$. By generating observations from

the chi-square distribution, we will investigate the capacity of the CUSUM chart to deal with skewed data and we will see if the choice $k = \delta/2$ is still convenient in this situation. Since the distribution of X is no longer symmetric, we will separate the discussion above in two parts. The first one investigates the influence of an upward mean shift. In the second one, we analyse the consequences of a downward shift.

4.2.1 Upward shift

The reference ARL_0 value used is again 200 and the corresponding h values for $k = 0.1,\ 0.25,\ 0.5,\ 1$ are calculated to be $8.96,\ 6.56,\ 4.88$ and 3.46. We notice that these values are all larger than the ones obtained for the normal and t_3 distribution. Figure 8 shows the evolution of ARL_1 in function of the shift size δ. In order to better demonstrate the difference among values of k, the y-axis is in log-scale.

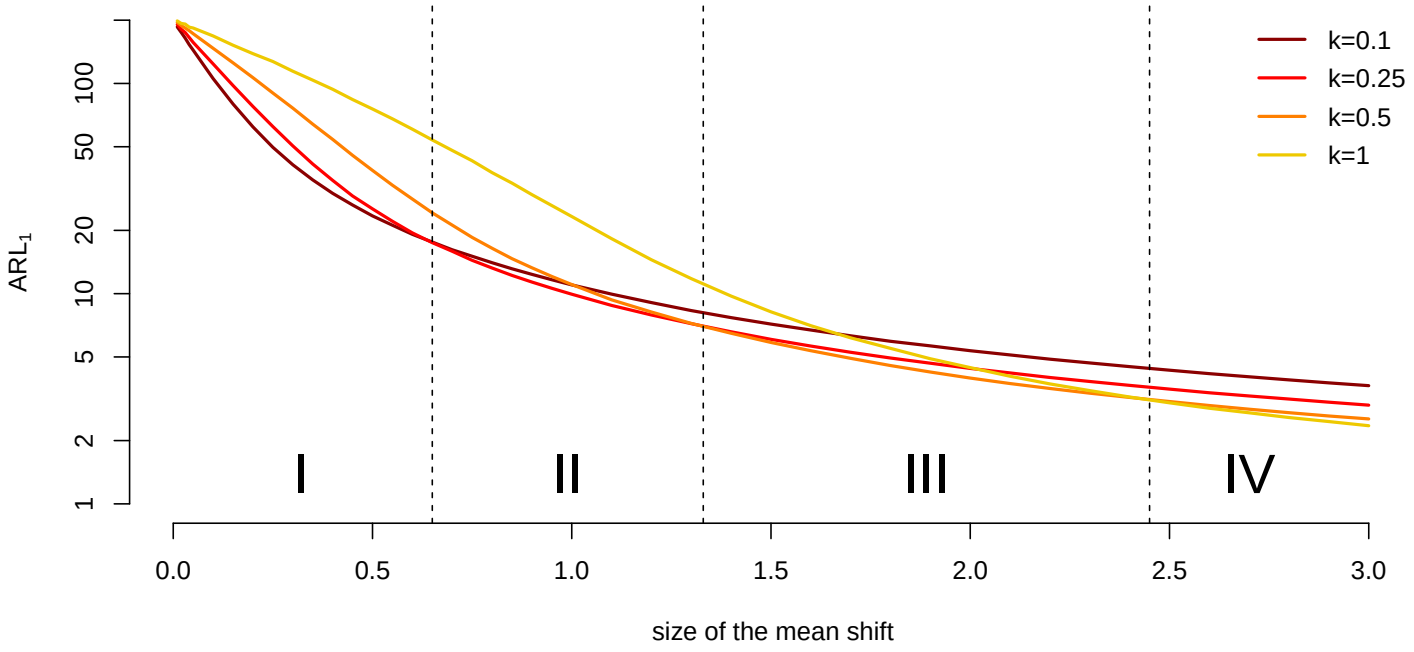

Figure 8: ARL_1 vs δ for different values of k. The data are generated from a chi-squared distribution with 3 degrees of freedom. Regions I, II, III and IV are determined such that $k = 0.1$ minimizes ARL_1 in region I, $k = 0.25$ is minimum in region II and so forth.

In the region I ($0 < \delta < 0.65$), the black curve of $k = 0.1$ has the lowest values. It means that the combination with $k = 0.1$ and $h = 8.96$ performs better than the 3 other ones when $0 < \delta < 0.65$. Similar interpretations hold for regions II, III and IV. We observe that the region I is larger for χ^2_3 data than for normal and t_3 data. We wonder if the choice of $k = \delta/2$ used for normal data is still appropriate for skewed data. We look at some numerical ARL_1 values estimated in figure 8 when

$\delta = 2k$. These values are given in the table below. As for table 2, the values are given by averaging the run length of 10 000 Monte Carlo simulations. Because of that, there is a small variability in these values.

	δ =0.2	δ =0.5	δ =1	δ =2
$k = 0.1$	62.25	23.27	11.03	5.37
$k = 0.25$	77.98	25.20	9.94	4.42
$k = 0.5$	106.39	38.11	11.05	3.99
$k = 1$	137.18	74.99	22.73	4.43

- When $\delta = 0.2$, the parameter $k = 0.1$ yields by far the better results.
- When $\delta = 0.5$, $k = 0.1$ gives the best performance again. The second best performance is $k = 0.25$.
- When $\delta = 1$, the best result is provided by $k = 0.25$. However, both $k = 0.1$ and 0.5 give close and similar ARL_1 values.
- When $\delta = 2$, $k = 0.5$ minimizes the ARL_1 value amongst the four values of k compared here. The 3 other values yield slightly larger ARL_1 values.

As for Student's t distribution, choosing $k = \delta/2$ doesn't always produce optimal results. However, there is no value of δ for which choosing $k = \delta/2$ produces really bad performances. Again, we guess that the value for k which minimizes the ARL_1 should be taken smaller than $\delta/2$. A chi-square distribution converges to a normal distribution when the number of degrees of freedom tends to infinity. Since $k = \delta/2$ provides good performances for data drawn from a normal distribution, it seems reasonable to assume that the performances of the choice $k = \delta/2$ will improve as long as the number of degrees of freedom grows.

4.2.2 Downward shift

The χ^2 distribution is skewed to the left and, until now, only upward CUSUM charts have been studied. What if the data were skewed to the right ? Or equivalently, what if we worked with a CUSUM chart designed to detect a downward shift in the mean ? We quickly present answers to these questions here. The chart used in this section is given by equations (20)-(21) and is

$$C_n^- = \min\left(0\ ,\ C_{n-1}^- + X_n + k\right) \qquad \text{for } n \geq 1, \tag{27}$$

with $C_0^- = 0$ and X_n drawn from a standardised chi-squared distribution This chart gives a signal of a downward shift if

$$C_{n-1}^- \ < \ -h. \tag{28}$$

Previously used R-code can be reused to produce figure 9. The only difference is that chart (27)-(28) must be computed instead of chart (25)-(26).

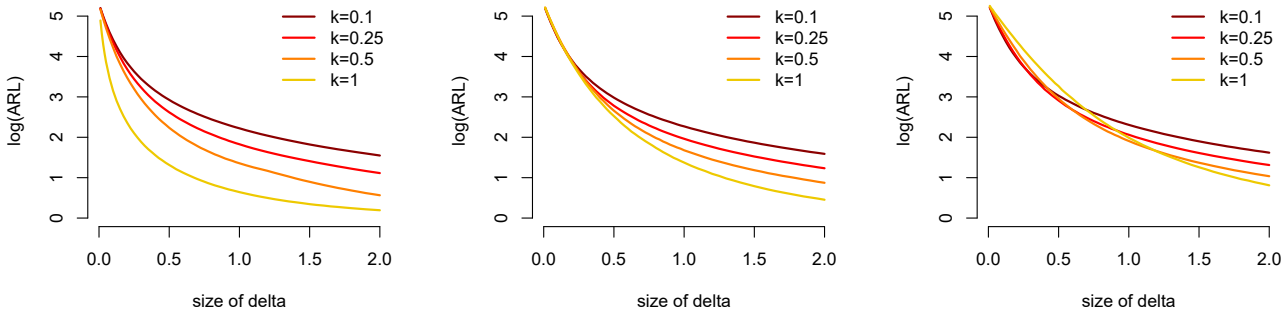

Figure 9: ARL_1 of the chart (27)-(28) in function of δ for different values of k. On the left graph, data are drawn from a χ^2_3 distribution. The middle and right graphs correspond to data simulated from a χ^2_{10} and χ^2_{100} distribution respectively.

Figure 9 shows the impact of asymmetry on the detection speed. We first note that the theoretical skewness coefficients for χ^2_3, χ^2_{10} and χ^2_{100} are 1.63, 0.89 and 0.28 respectively. The left graph shows that the choice $k = \delta/2$ is not adequate for the chart (27)-(28) with χ^2_3 data. Indeed, one can see that, no matter the value of δ, the chart with $k = 1$ has the fastest detection speed. In this situation, the CUSUM chart needs in average fewer observations to detect a downward shift than an upward one. The second graph of figure 9 depicts ARL curves for χ^2_{10} data. Performances are similar for small shifts ($\delta < 0.5$). However, the chart with the largest k provides the best performance when the shift size increases. The CUSUM chart needs, in average, more observations to detect a downward shift in χ^2_{10} data than for χ^2_3. It is barely visible on the last graph but each curve has a range of values of δ where it minimizes, among the 4 tested combinations, the ARL_1. A χ^2_{100} distribution is similar to a normal distribution. Therefore, it is logical to obtain a graph close to the one described on figure 4 for normal data.

Suppose without loss of generalities that we have to deal with right-skewed data. In a right skewed distribution, more than a half of the mass is located left to the mean. Two main scenarios can occur when we deal with data drawn from an asymmetric distribution. If we want to detect an upward mean shift, then the classical CUSUM charts work. In this case, the chart parameter $k = \delta/2$ produces acceptable performances. If it is necessary to have optimal performances, k should be chosen smaller than $\delta/2$. If we want to detect a downward mean shift, then the classical CUSUM charts provide different results. In this case, choosing $k = \delta/2$ is no more adapted. Simulations showed that for heavily skewed data, choosing a large k value will lead to better performances. The CUSUM chart is faster to detect a downward shift in the mean than an upward one. These effects are reduced as long as skewness closes up to 0.

5 CUSUM Charts for Non-Uniformly Shifted Data

Until now, we assumed that when the process goes into an out of control state, the mean of the generated data gets shifted. If the data are drawn from a normal distribution it means that the IC distribution is $\mathcal{N}(\mu_0, \sigma^2)$ and the OC distribution $\mathcal{N}(\mu_1, \sigma^2)$. Suppose now that $\mu_1 = 1$, $\mu_0 = 0$ and that a CUSUM chart is designed with parameters $k = 0.5$ and $h = 3.502$. The value for k is chosen such that it minimizes the average time needed before the shift in the mean is detected. h is computed by algorithm 1 so that the ARL_0 value is 200. In this case, the run length distribution after a million of Monte-Carlo simulations has the form presented on figure 10.

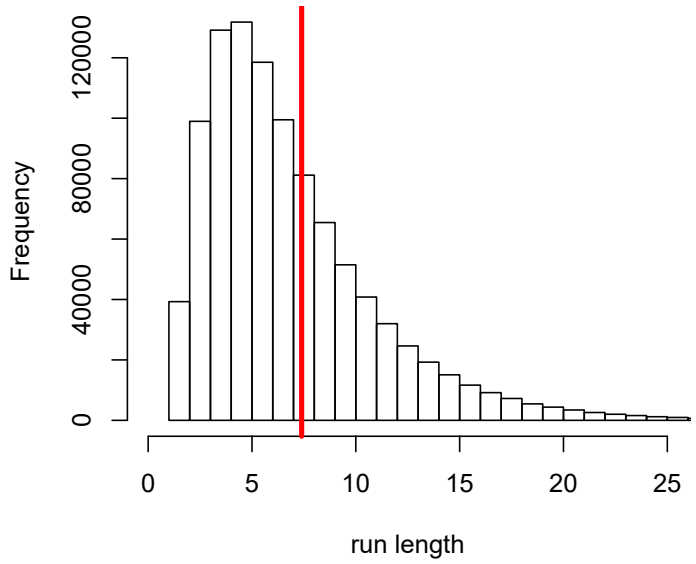

Figure 10: Run length distribution when the process is out of control. The data are generated from a normal distribution. The mean shift is of size $\delta = 1$. The red line illustrates the ARL_1 value.

It can be seen that the run length distribution is significatively skewed to the right. The skewness coefficient of this distribution is 1.66. The ARL is 7.39, standard deviation is 4.29 and the value for the median is 6. Run length has a lot of variability since quantiles 2,5% and 97,5% are 2 and 18 respectively.

What if a time dependance is introduced in μ_1 ? This chapter will study the impact of non constant shift on the run length distribution of a CUSUM chart. We will be particularly interested in the form of the distribution depending on $\mu_1(t)$. ARL_1 values cannot be simply compared since different functions for $\mu_1(t)$ will be used. Indeed, it is known that large values for $\mu_1(t)$ will quickly lead to an alarm signal. W.l.o.g. the upward CUSUM chart will be used.

5.1 Presentation of the Shift Functions and their Respective Run Length Distributions

Four differents functions for $\mu_1(t)$ are considered. They are described hereafter and plotted on figure 11.

$$\mu_{\text{lin}}(t) = \frac{t}{30}, \tag{29}$$

$$\mu_{\text{quad}}(t) = \left(\frac{t}{30}\right)^2, \tag{30}$$

$$\mu_{\text{sqrt}}(t) = \sqrt{\frac{t}{30}}, \tag{31}$$

$$\mu_{\text{sin}}(t) = \sin\left(\frac{t}{1.9}\right). \tag{32}$$

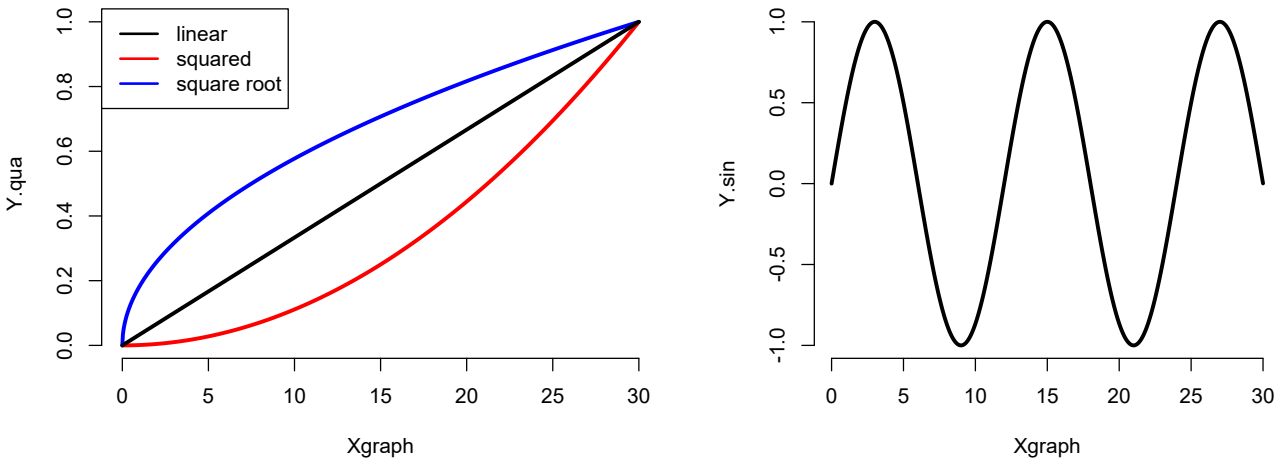

Figure 11: Visualisation of the shift functions that will be used.

The coefficients of these four functions have been arbitrarily chosen such that $\mu_1(t)$ takes values between 0 and 1 on an interval sufficiently large to see clearly the run length distribution. If $\mu_1(t) < 0$ for some t the upward CUSUM chart is not appropriate and the change in mean will never be detected. Conversely, if the change in means is too big, it will be almost instantaneously detected and this is not the goal here. In equation (32), the coefficient 1.9 is chosen in order to have 3 as the number of complete positive periods in $[0, 30]$.

For each of the $\mu_1(t)$ functions presented above, a million of Monte Carlo simulations are performed in order to have a representative idea of the run length distribution. The chart parameters $k = 0.5$ and $h = 3.502$ are used again (ARL_0=200). The results are given on figure 12. The vertical red lines represent the ARL for each OC mean function.

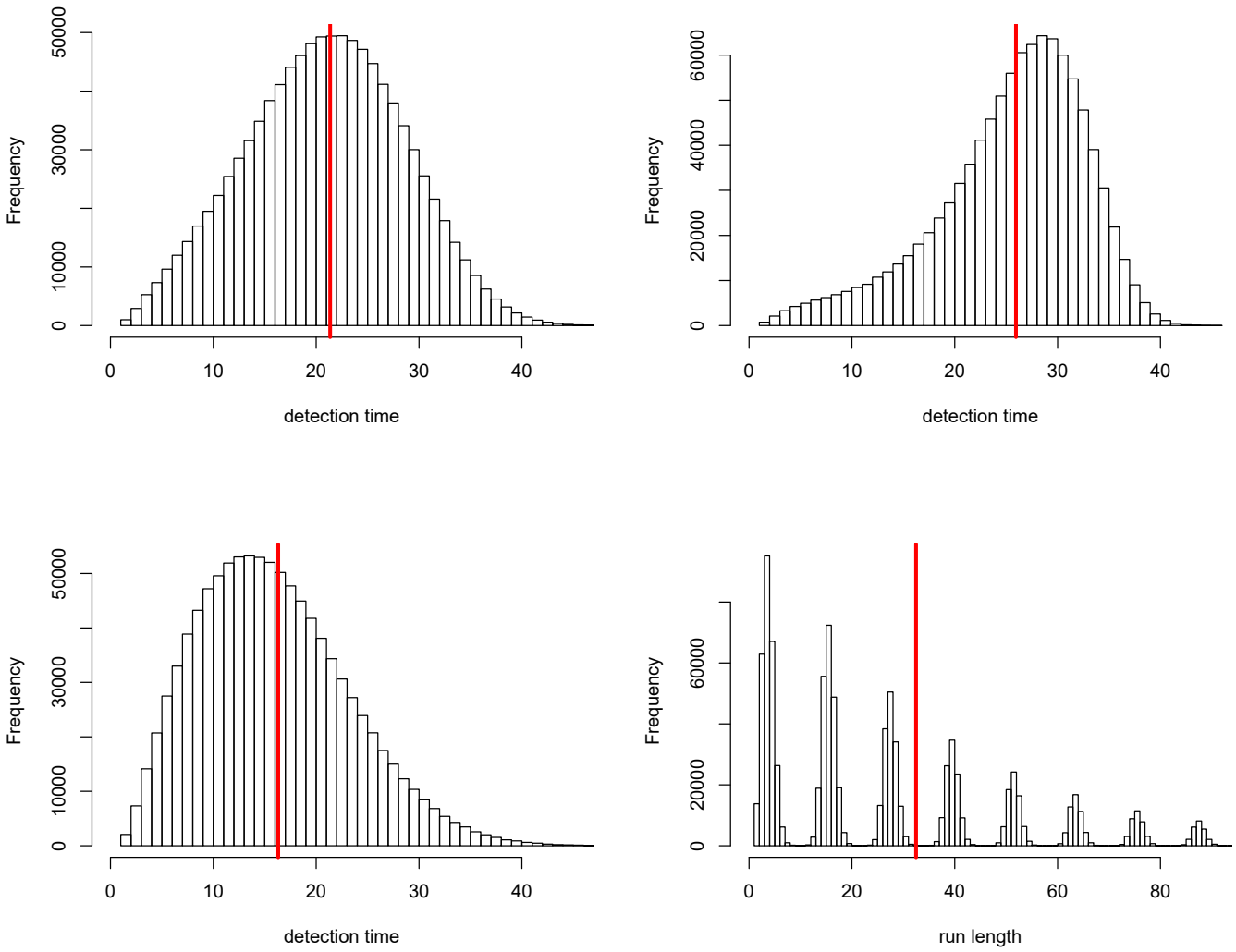

Figure 12: Run length distributions for different shift functions. The first histogram corresponds to the shift μ_{lin}. The top right graph corresponds to μ_{quad}. The bottom left histogram is for μ_{sqrt} and the bottom right corresponds to μ_{sin}. The vertical red lines represent the average run length for each shift function.

The histogram on the upper left corner shows the run length distribution when the mean of the out of control distribution is linear. ARL is 21.36, run length standard deviation is 7.66 and skewness is -0.03. The skewness coefficient means that the distribution is symmetric and quantiles 2,5% and 97,5% are 6 and 36 respectively. The frequency of stopping times grows steadily at the beginning, reaches a maximum close to the average run length and then decreases approximately at the same rate it increased. The observed distribution is approximately normal. Small runs are not frequently observed because $\mu_{\text{lin}}(t)$ is close to 0 for small t values. As t grows, $\mu_{\text{lin}}(t)$ takes larger value and the probability of an OC signal increases too. For large t, the detection frequency is small because the control chart has already flagged the process as being OC.

The histogram on the upper right corner shows the run length distribution when the mean of the generated data is $\mu_{\text{quad}}(t)$. ARL is 25.95, run length standard deviation is 7.21, the skewness is -0.77 and thus the distribution is skewed. Quantiles 2.5% and 97.5% are given by 8 and 37. Short runs are not frequently observed because $\mu_{\text{quad}}(t)$ takes small values for small t (see figure 11). The detection frequency is large for t close to 30 because $\mu_{\text{quad}}(t)$ takes values close to 1 for the first time. It doesn't happen that there are long runs. Long runs would imply that $\mu_{\text{quad}}(t)$ is large, and if that is the case, then it is likely that an OC signal has already been sent before. More generally, suppose that the OC mean has the form

$$\left(\frac{x}{a}\right)^b$$

with $a\ ,\ b \in \mathbb{N}_*$. Then, a does not impact the shape of the run length distribution but rather influences the scale of the distribution. ARL would grow if a increases. Also, an augmentation in a would imply an increase in the run length standard deviation. The function x^b converges to 0 when b tends towards infinity on $[0\ ;1[$. It implies that, when b increases, run length frequency will be smaller than the one on figure 12 for $t\ <\ a$. An augmentation in b would lead to an increase in ARL and a decrease in standard deviation.

The histogram on the lower left corner of figure 12 corresponds to $\mu_{\text{sqrt}}(t)$. ARL is 16.31 in this case and standard deviation is 7.28. The skewness coefficient is 0.54 which means that more than the half of the probability mass is situated left to the ARL vertical red line. Quantiles 2.5% and 97.5% are 5 and 32 respectively. This situation is in some sense the opposite of a polynomial shift. Again, no OC signal is sent at time $t = 1$ and a relatively small amount at time $t = 2$. However the frequency of stopping times grows quickly for relatively small values of t. After the peak is reached, the frequency decreases steadily. This phenomenon happens because the square root function $\sqrt{x}$ is closer to 1 than a linear or a monomial function when x is between 0 and 1 (see figure 11). More generally, suppose that the mean of the OC distribution has the following form :

$$\left(\frac{x}{c}\right)^{\frac{1}{d}} = \sqrt[\text{d}]{\frac{x}{c}}$$

with $c\ ,\ d \in \mathbb{N}_*$. This case is similar to the case of a polynomial function in the sense that c is only a scale parameter which doesn't influence the form of the distribution. The function $x^{1/d}$ converges to 1 when d tends towards infinity on $]0\ ;1]$. So, as long as d increases, $\mu_{\text{sqrt}}(t)$ converges to the uniform shift function used in the previous chapters. It implies that the run length distribution will be more skewed. If d is very large, we can expect the run length distribution to have a shape similar to the one on figure 10.

Finally, the last histogram of figure 12 depicts the run length distribution when the mean is $\mu_{\sin}(t)$. ARL is 32.48 and standard deviation is 32.68. Skewness is 1.98 and thus the distribution is heavily skewed. Quantiles 2.5% and 97.5% of the distribution are 3 and 123. Small hills are observed. They all have the same width and their heights decrease as time progresses. The frequency of stopping times is zero or very small when $\mu_{\sin}(t)$ is negative.

5.2 Conclusion

In this chapter, we tried to get an idea of what the run length distribution could look like for different types of shifts. Figure 12 showed that the form of the distribution varies enormously from one shift to another. In order to be able to apply on real data what has been observed in this chapter, it is needed to have an idea of the type of shift encountered. The type of shift can be known based on scientific knowledge about the production process or based on past experiences. Otherwise, then we assume that a dataset is available, which is used for the calibration of the chart. On this dataset, we can evaluate the form of the shift function. To do this, we can use a classical nonparametric regression like the Nadaraya-Watson kernel regression. Based on this regression, we can visually estimate the form of the shift for this dataset. After that, we assume that the shift determined by nonparametric regression is the same for the online monitoring. Based on our estimated shift and on the results of figure 12, a general idea of the form of the run-length distribution can be given.

6 Autocorrelated Observations

In the previous chapters, we designed classical CUSUM chart and analysed its performances for several forms and sizes of shift. Until now, we assumed that the process observations are independent and identically distributed. In this section, we consider situations where process observations collected at different time points are correlated. As for the previous chapters, we focus on the CUSUM chart designed to detect an upward shift in the mean

$$C_n^+ = \max\left(0, C_{n-1}^+ + (X_n - \mu_0) - k\right), \qquad \text{for } n \geq 1, \tag{33}$$

where $C_0^+ = 0$. Firstly, we will illustrate the effects of autocorrelated observations on CUSUM charts. Then, a parametric solution to the autocorrelation problem will be presented.

6.1 Impact of correlation on the ARL

We can generate dependent data by simulating an AR(1) process. Observations following an auto-regressive process of order 1 are generated as follows, for $n \geq 1$:

$$X_n = \phi X_{n-1} + e_n. \tag{34}$$

In the preceding equation, X_0 is defined as 0. $\{e_n : n \geq 1\}$ are the errors terms and they are here assumed to follow a standard normal distribution. The error terms are independent. The AR(1) process of equation (34) is stationary if $-1 < \phi < 1$. In this process, the autocovariance for a lag $l \geq 0$ is given by

$$\text{Cov}(X_t, X_{t+l}) = \sum_{k=0}^{\infty} \phi^k \phi^{k+l} = \phi^l \frac{1}{1-\phi^2}. \tag{35}$$

We note that ϕ appears at 2 different places in the autocovariance of the serie. $1/(1-\phi^2)$ is the overall variance of the AR(1) process. $1-\phi^2$ decreases when $|\phi|$ increases. So, the overall variance of the process rises when $|\phi|$ increases. In addition, the autocorrelation of lag l of an AR(1) process is given by ϕ^l. So, the higher $|\phi|$, the higher the absolute value of autocorrelation. The intensity of the autocorrelation decreases when the lag grows.

Figure 13 describes the ARL_0 of the CUSUM chart given by (33) for a range of values of ϕ going from -0.99 to 0.99. The parameters of the chart are calibrated such that the ARL_0 for independent data is 200. The h values are computed thanks to the searching algorithm described on page 11. The h found for $k = 0.5$ and $k = 0.75$ are 3.502 and 2.481, respectively. For each value of ϕ, 50 000 time-series following an AR(1) process are simulated. The ARL_0 presented on the graph is computed by averaging the 50 000 calculated run length.

We can see the ARL_0 is smaller than 200 for ϕ between 0 and 1. We also observe that the computed ARL_0 is similar for both k values when ϕ is positive. However, these values become significatively different when ϕ is negative. We remark that the

ARL_0 for $k = 0.5$ is larger than for $k = 0.75$ when ϕ is negative. Even if the results are much more similar, the reverse inegality holds for $0 < \phi < 1$.

Small ARL_0 values are obtained for ϕ close to -1. Conversely, both ARL_0 curves increase when ϕ gets closer to 1. This is due to more variance in the process. The variance of the process observation is minimal and is 1 when $\phi = 0$. It increases as $|\phi|$ gets close to 1. By taking a closer look on the run length for ϕ close to 1, we realise that most runs are short or very short and some are very long. On average, 60% of the runs are 3 or less. If ϕ is close to 1, an AR(1) process is close to a random walk. If, at some point of the walk, the process takes large negative values, then it might take a long time until the process takes on a positive value.

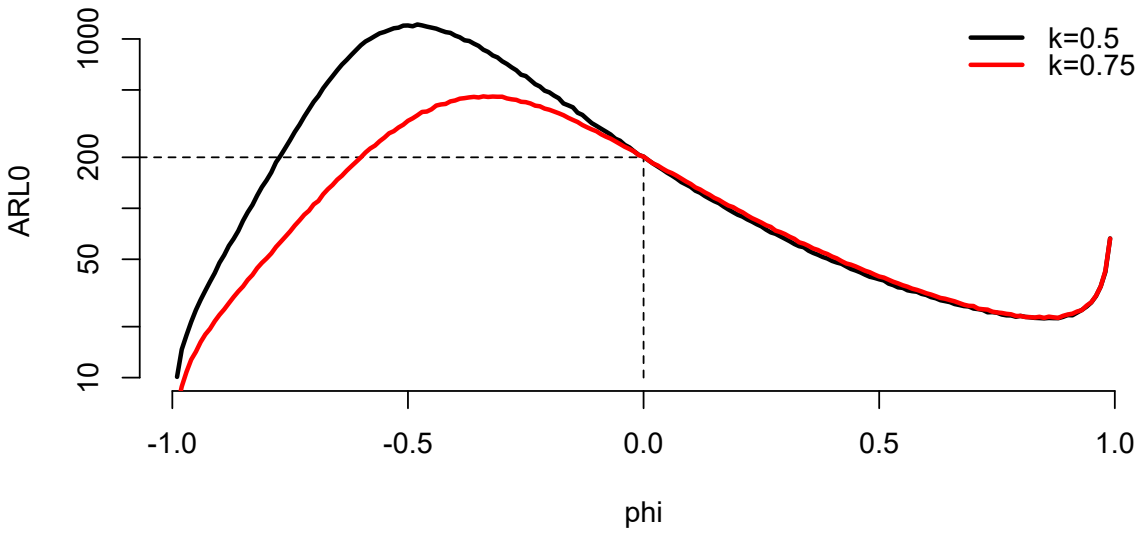

Figure 13: ARL_0 in function of ϕ for data generated from an AR(1) process. The black curve corresponds to $(k, h) = (0.5, 3.502)$ and the red to $(k, h) = (0.75, 2.481)$. The y-axis is in log-scale.

On the one hand, an ARL_0 value larger than 200 would lead to an augmentation in ARL_1. It means that more observations will be needed to detect a mean shift in the data. On the other hand, an ARL_0 smaller than the pre-specified value is not desired either. If it happens, the ARL_1 would have a smaller value than for independent data. A consequence of this is that the chart would send too many false alarm signals. It means that the process would be stopped for unnecessary reasons too frequently. In conclusion, we have seen that correlated observations have an impact on the runs that cannot be neglected.

6.2 How to deal with correlated data ?

In this part we describe a strategy, and the idea behind it, which copes with autocorrelated observations. The method is also presented in Qiu [9]. The idea is actually to model the correlation and apply the CUSUM chart on the residuals of this model. Since we are working with time series, statistical models for time series analysis will be used here. The Auto-Regressive Moving Average (ARMA) model was described by Box and Jenkins. It provides a description of a stationary process by coefficients of two polynomials. The model equation is given by

$$X_n = c + \sum_{i=1}^{p} \phi_i X_{n-i} + \sum_{j=1}^{q} \theta_j e_{n-j} + e_n \tag{36}$$

where $c, \phi_1, \dots \phi_p, \theta_1, \dots, \theta_q$ are the model coefficients. $\{e_n : n \geq 1\}$ are the errors terms and they are here assumed to follow a normal distribution with mean 0 and variance σ_e^2. When $q = 0$, the ARMA model becomes an auto-regressive model of order p noted by AR(p). When $p = 0$, the ARMA model is a moving average model of order q (MA(q)). It is possible to find p and q such that information criteria like AIC and BIC are minimised. These information criteria are calculated on the basis of a trade-off between fitting the data and not using too much variables. After the best[2] ARMA(p, q) model is fitted, it is possible to extract the residuals $\hat{e}_n$ for $n \geq 1$. These residuals are given by

$$\hat{e}_n = X_n - \hat{X}_n, \tag{37}$$

where $\hat{X}_n$ is the fitted value for X_n.

If the process is in control up to point n and if the chosen ARMA model is adequate for the data, then the set of $\hat{e}_n$ is approximately independent and identically distributed. Additionaly, $\{\hat{e}_n : n \geq 1\}$ are normally distributed with mean 0 and variance $\widehat{\sigma_e^2}$ with $\widehat{\sigma_e^2}$ the classical variance estimator of the residuals. Since $\hat{e}_n$ is i.i.d. and normal, the classical assumptions of CUSUM charts described at page 15 are satisfied. It means that the classical CUSUM chart that we developped earlier can be used on the residuals. Suppose we are interested in detecting an upward mean shift, then the charting statistic is, for $n \geq 1$,

$$C_n^+ = \max\left(0, C_{n-1}^+ + (\hat{e}_n - \mu_0) - k\right) \tag{38}$$

where $C_0^+ = 0$. Later, the method described above will be called the classical parametric method. We illustrate the use of chart (38) by the following example.

Example 4. *We generate data from an AR(2) process with parameters $\phi_1 = 0.9$, $\phi_2 = -0.4$, $\sigma_e^2 = 1$ and standardise them. The first 100 observations of the time series are visible on the first graph of figure 14. We look at the time taken by the classical CUSUM chart to detect an upward mean shift of size 1. The chart parameters are $(k, h) = (0.5, 3.5)$. From the second plot of figure 14, it can be seen that the classical chart applied to the standardised original time-series gives numerous alarm signals. The first signal occurs early, at time 12. After that, we*

[2] in the sense of AIC and BIC

use the function arima in R to fit an AR(1) and an AR(2) model to the data. If we fit an AR(1) model, the estimated coefficient is $\hat{\phi}_1 = 0.632$. *Then, the chart* (38) *is calculated. The charting statistic is plotted on the left bottom graph of figure 14. This time, there is only one alarm signal and it occured at the 55th time points. If we fit an AR(2) model, estimations of the two auto-regressive coefficients are* $\hat{\phi}_1 = 0.882$ *and* $\hat{\phi}_2 = -0.396$. *The CUSUM chart is calculated with the residuals and the chart statistic is plotted on the last graph in the next figure. Two signals are sent at the 62 and 63rd time points and a signal was almost sent at time 60.*

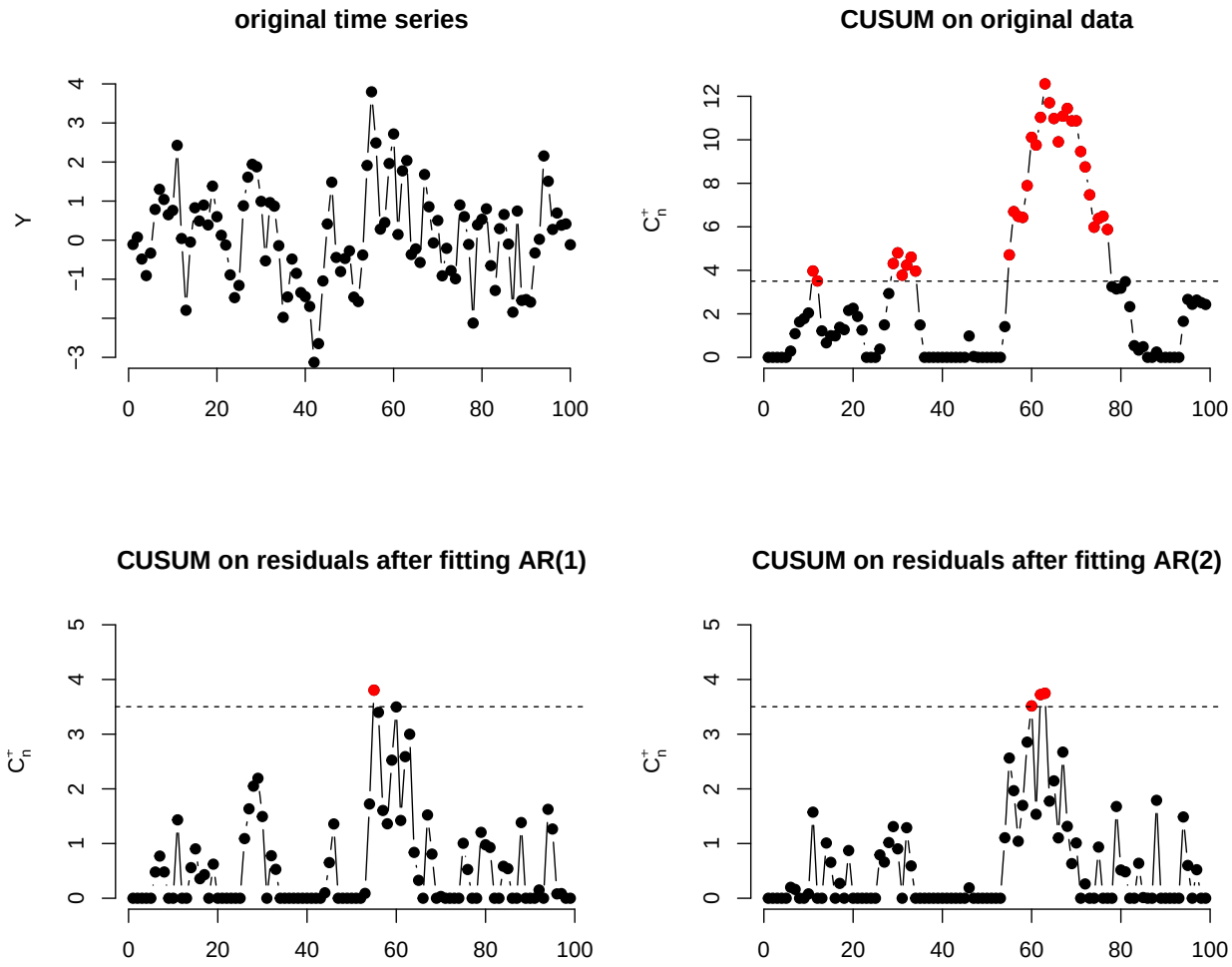

Figure 14: The top-left graph is the original time series. It has been generated by an AR(2) process of parameters $\phi_1 = 0.9$ and $\phi_2 = -0.4$. The top-right graph is the classical CUSUM chart applied on original data (after standardisation) with parameters $(k, h) = (0.5, 3.502)$. Both bottom graphs are chart (38) applied to the residuals of the fitted AR(1) and AR(2) models.

Based on the discussion around figure 13 at page 30, signals at times 11-12 and 29-34 of the chart for original data might be due to the autocorrelation present in the

data. Since chart (38) *deals better with autocorrelation, these signals are no longer sent. When we fit an AR(1) model to AR(2) generated data, the correlation is only partially captured by the model. This implies the residuals of AR(1) are less close to white noise than those of the AR(2). However, the figure shows that it seems the main part of the autocorrelation is captured by the AR(1) model in this case.*

6.3 Conclusion

In this chapter, we have seen that the presence of autocorrelation in the process observations cannot be neglected. A possible way to deal with autocorrelation is to fit the best ARMA model to the data and then, apply the CUSUM chart to the residuals. If the data are adequately modelled by the ARMA model, then the residuals are close to white noise (i.e. independent normal data). However, this solution is not perfect since the process observations may not be sufficiently well modelled by an ARMA model. This solution will be compared with another one in the next chapter.

7 Bootstrapping the run-length of CUSUM charts

Chapter 6 presented an approach to deal with correlated data. This approach fits an ARMA model to the data and applies the CUSUM chart on the residuals. In this chapter, a whole different method based on bootstrap is presented. As a first step, we analyse the results provided by bootstrap when data are normal and independent. In the second part of the chapter, a bootstrap technique is applied on correlated data. After that, we try to establish a fair comparison between the performance of the bootstrap-based method developped in this chapter and the parametric approach described in chapter 6.

7.1 Independent data

In this section 7.1, we focus on the effect of bootstrap when the process observations are independent. In particular, we will analyse mainly the effect of 2 factors : n_B, the size of the bootstrap sample and B, the number of bootstrap replicates on the ARL_1. As for the ARL_1, we will try to understand the effects of B and n_B on the run length standard deviation (SDRL). We will also see how the size of the shift δ affects the results provided by bootstrap.

7.1.1 Effects of n_B and B

7.1.1.1 Presentation of the study

Without loss of generalities, we are focussing on the CUSUM chart designed to detect an upward shift in the mean described in equation (18) and (19). For a prespecified ARL_0 value and k, a value for h can be estimated thanks to the searching algorithm described at page 11.

Since we are working with i.i.d. data, we use the basic bootstrap procedure. The basic bootstrap resamples from the empirical distribution function of the original sample. Here, we denote the size of the original sample by n_B. B is the number of resamples. Algorithm 2 describes how the bootstrap can be applied to the run-length of a CUSUM chart. Algorithm 2 can be summarized by the following steps :

1. Generate a sample of size n_B from the OC distribution.
2. Compute the run length by drawing observations from the sample of step 1 and repeat this B times.
3. Compute the mean and the standard deviation of the B run lengths.
4. Replicate steps 1-3 M times and average the M values for mean and standard deviation.

In a first time, data are simulated from a standard gaussian distribution. Thus, the OC distribution is a normal distribution with mean $\mu_1 = \mu_0 + \delta$ and variance 1.

Algorithm 2 Estimate ARL_1 value by bootstrap

Let M be the number of Monte-Carlo simulations. M must be a large positive integer.
Let N be a large number defining the maximum possible run length.
Let n_B be the sample size of the bootstrap sample.
Let B be the number of bootstrap replicates.
Fix values for parameters of the control charts (k and h) and for the parameters of the IC and OC distributions.
for m in $1, \ldots, M$ **do**
  Put the charting statistic to 0.
  Draw a sample of size n_B from the OC distribution.
  for b in $1, \ldots, B$ **do**
    for n in $1, \ldots, N$ **do**
      Draw an observation from the bootstrap sample
      Update the value of the charting statistic
      if the charting statistic is smaller than h **then**
        let $n = n + 1$ and continue the loop.
      else
        the charting statistic is larger than h and $\text{RL}_b^m = n$ (where RL_b^m is the run length computed at the m-th simulation and the b-th replication)
        break
      end if
    end for
  end for
  $\text{RL}^m \leftarrow \frac{1}{B}\sum_{b=1}^{B}\text{RL}_b^m$
  $\text{SDRL}^m \leftarrow \sqrt{\frac{1}{B}\sum_{b=1}^{B}(\text{RL}_b^m - \text{RL}^m)^2}$
end for
$\text{ARL} \leftarrow \frac{1}{M}\sum_{m=1}^{M}\text{RL}^m$
$\text{SDRL} \leftarrow \frac{1}{M}\sum_{m=1}^{M}\text{SDRL}^m$

7.1.1.2 Results

The results given in table 3 are obtained when the shift size is $\delta = 1$. We have seen on figure 4 that an optimal choice of k is given by $k = \delta/2 = 0.5$. For $k = 0.5$ and a prespecified ARL_0 value of 200, algorithm 1 gives a value for the control limit of $h = 3.501$. Now that a value for the parameters of the chart is available, algorithm 2 is used to estimate the ARL_1 for different values of n_B and B. For each value of (n_B, B), $M = 50000$ simulations are performed. On the next table, we compare values obtained by bootstrap to a *reference value*. This *reference value* is also obtained by Monte-Carlo simulations. At each simulation, a sample is drawn and the run length of the CUSUM chart is computed. Finally, the *reference value* is computed by averaging the run lengths computed at each simulation.

On table 3 it can be observed that no matter what the combination of values for n_B and B is, the bootstrap estimation for the ARL_1 is larger than the reference ARL_1 value. However, the size of the overestimation decreases as the size of the

bootstrap sample increases. The number of bootstrap replicates B does not impact the estimated ARL_1. Run length standard deviation also decreases when n_B increases. As for the ARL_1, B does not influence the SDRL.

n_B	B	ARL_1	bias	SDRL
50	200	7.749	0.362	4.554
	500	7.717	0.331	4.533
	1500	7.749	0.362	4.566
	5000	7.752	0.365	4.573
100	200	7.583	0.196	4.409
	500	7.536	0.150	4.390
	1500	7.542	0.155	4.401
	5000	7.533	0.146	4.396
200	200	7.485	0.098	4.343
	500	7.482	0.095	4.346
	1500	7.467	0.080	4.338
	5000	7.481	0.094	4.354
500	200	7.416	0.029	4.288
	500	7.413	0.026	4.294
	1500	7.410	0.023	4.293
	5000	7.410	0.023	4.314
1000	200	7.422	0.035	4.293
	500	7.405	0.018	4.287
	1500	7.393	0.006	4.283
	5000	7.411	0.025	4.300

Table 3: Effect of bootstrap on normal CUSUM chart when $\mu_0 = 0$, $\mu_1 = 1$, $k = 0.5$ and $h = 3.5$. For this combination of k and h, the ARL_0 value is 200. The bias is computed as the difference between the ARL_1 and the *reference value*. The *reference value* is computed to be 7.388 in Qiu [9].

Table 5 in appendix reproduces the simulation study presented here for data drawn from a t_3 distribution. The results are similar to those obtained for normally distributed data.

7.1.2 Performance according to δ

7.1.2.1 Average Run Length

We are now interested in what happens when shift size δ takes different values. So, the chart (18)-(19) is used but this time with different values for δ. Figure 15 shows the estimated ARL_1 by bootstrap for a range of values of δ going from 0 to 1.5. For each value of δ, we begin by fixing $k = \delta/2$. Then, for all δ, we have to find the h which leads to the prespecified ARL_0 value. This h is computed through algorithm 1. Then for every couple (k, h), data are drawn from a normal distribution with mean δ and variance 1. After that, the *reference ARL_1 curve* is computed and the curve is plotted in black on figure 15. All values of this *reference curve* are

computed in the same way as the reference value in the table 3. Finally, ARL_1 is estimated by the bootstrap method described in algorithm 2 for different sample sizes n_b and plotted on figure 15.

The pre-specified ARL_0 value used is 200. For each value of δ, we estimate ARL_1 by 20000 Monte Carlo simulations. The ARL_1 values estimated by bootstrap are obtained using algorithm 2. Since we had seen previously that B seemed to have no effect on the estimation of ARL_1, the results of figure 15 are all obtained for the fixed value $B = 500$.

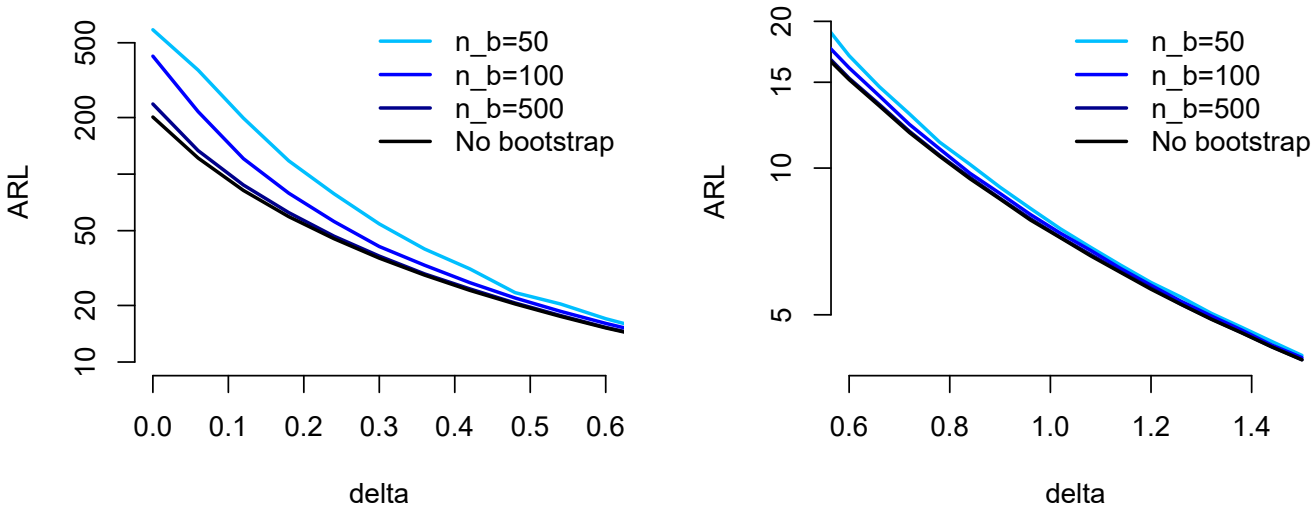

Figure 15: Effect of bootstrap on the run lengths of a CUSUM chart for different shift sizes. For each value of δ, the chart parameters are $k = \delta/2$ and h is adjusted such that the ARL_0 value is 200. The blue curves represent the ARL_1 estimated by bootstrap for different sample sizes n_B. The black curve represents the *reference ARL_1 curve* computed without bootstrap. The graph is separated in two parts for a better visualisation. To better demonstrate the difference among curves, the y-axis is in log-scale.

Figure 15 shows that the differences between the blue curves and the black curve reduce when δ increases. It can also be seen that, no matter the value of δ, the bias induced by the bootstrap decreases when the sample size increases. For large values of δ, the curves are almost superimposed and therefore the difference is small. The bootstrap should not be used for small shift sizes. For example, if $\delta = 0.1$, the black curve provides an ARL_1 value smaller than 100. However, for a bootstrap sample of size 50, the ARL_1 computed with bootstrap is larger than 200.

Figure 22 in appendix reproduces the simulation study for data drawn from a t_3 distribution. The results are similar to those obtained for normally distributed data.

7.1.2.2 Run length variability

Previously, results shown on table 3 suggested that the size of the bootstrap sample influences the SDRL. In particular, we had seen that an increase in n_b implies a decrease in the SDRL. The results of table 3 are obtained for $\delta = 1$, $k = 0.5$ and $h = 3.5$. In this part, we are interested in the effect of bootstrap on the SDRL for different values of δ. In order to evaluate this, we use the same simulations performed for the ARL_1 and use them again to compute the SDRL. The SDRL is computed as described in algorithm 2. The results obtained are shown on figure 16.

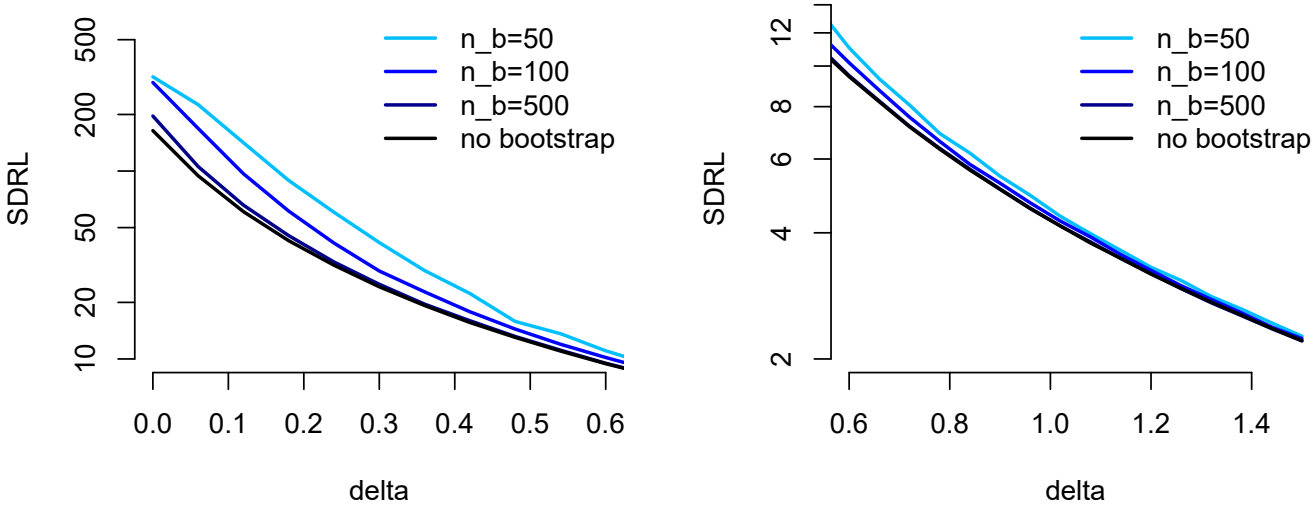

Figure 16: SDRL estimated by bootstrap for different values of δ and for different sample sizes n_B. The black curve represents the *reference SDRL curve* when no bootstrap is performed. The graph is separated in two parts for a better visualisation. The y-axis is in log-scale.

On figure 16, we can see that the differences between the blue curves and the black curve reduce when δ increases. It can also be seen that, no matter the value of δ, the difference caused by the bootstrap decreases when the size of the bootstrap sample increases. For large values of δ, the blue curves are close to the black curve and the impact of bootstrap on the SDRL is close. For small values of δ, and small bootstrap sample sizes, the SDRL computed by bootstrap is a lot larger than the SDRL of the *reference curve*.

7.1.3 Conclusion

In this chapter, we have analysed the influence of bootstrap on the run lengths of the CUSUM chart when the process observations are independent. We have seen that the ARL_1 estimated by bootstrap tends to overestimate the *reference* ARL_1

computed without bootstrap. The size of the overestimation decreases when the sample size increases. The number of bootstrap replicates B doesn't influence the computed ARL_1. After that, we have observed the influence of bootstrap on the ARL_1 for different sizes of mean shift. For large mean shifts and whatever the size of the bootstrap sample, we have seen that the difference caused by bootstrap on the ARL_1 is nearly inexistant. However, the bootstrap should not be used for small shifts close to 0. The results observed for the SDRL are similar to those observed for the ARL_1. So, it means the SDRL obtained by bootstrap overestimates the reference SDRL computed without bootstrap. The size of the overestimation decreases when the size of the bootstrap sample or the size of the shift increases.

7.2 Correlated data

In the first part of this chapter, the influence of bootstrap on the run lengths of the CUSUM chart was studied for independant data. In this part, we extend this to correlated data. In section 7.1, a basic bootstrap was performed. The idea behind the basic bootstrap is to resample directly from the empirical c.d.f. in order to mimic the statistical properties of the original sample. It is no longer appropriate to do this for correlated data since the basic bootstrap samples observations one by one. By sampling one by one, the correlated feature of the data is not reproduced. In order to deal with the time dependency, the block-bootstrap is introduced.

7.2.1 Block-bootstrap : bootstrap for time series

The method discussed hereafter is called "Non-overlapping block-bootstrap". The concept behind block-bootstrap is to cut the original autocorrelated data in blocks of the same length l. We note b is the largest number of non-overlapping blocks of length l that we can construct. The bootstrap time series is then constructed by sampling with replacement m out of the b blocks. Thus, all the bootstrap series have length ml. Other more elaborate methods exist such as Moving (or overlapping) block-bootstrap and Circular block-bootstrap but they will not be used here.

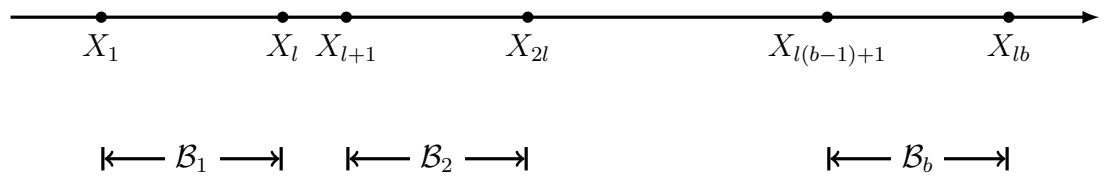

Figure 17: Non-Overlapping block-bootstrap scheme. The original time series is cut in b blocks of length l. The bootstrap time series is constructed by sampling with replacement out of these b blocks.

A parameter that comes with the block-bootstrap procedure is the block length l. If l is too small, then the time depencies won't be sufficiently captured by the

block-bootstrap. For instance, the block-bootstrap with $l = 1$ is exactly the basic bootstrap. If l is too large, there won't be enough blocks and the bootstrap samples may look too similar. Classical choices for the block size l are Cn^{δ}, with n the length of the original time series, C a constant and $\delta \in (0; \frac{1}{2})$ [4].

7.2.2 Parametric vs Block-bootstrap approach

In this section, we will compare the results obtained by block-bootstrap and those given by the parametric method described in part 6.2. The main interest will be to compare both approaches in a situation where the correlation structure is complex. At first, we will present the way used to compute the ARL_1 values for the two methods. Then, we will present two different ways for comparing the performance in more detail. After that, for these two different comparisons, we will introduce the results and draw some conclusions.

7.2.2.1 Presentation of the 2 different approaches

In this part, we present the way the ARL_1 is computed for both methods. The main steps in the computation of the ARL_1 by what we call *the parametric method* are summarized in algorithm 3 below.

Algorithm 3 ARL_1 computed by the parametric method

Let M be the number of Monte-Carlo simulations. M must be a large positive integer.
Let N be a large number defining the maximum run length possible.
Give a value for k, ϕ and θ.
Recalibrate h depending on k, ϕ and θ by using algorithm 1.
for m in $1, \ldots, M$ **do**
  Simulate a time series from an ARMA(1,1) model with parameters ϕ and θ.
  Fit an AR(1) model to the time series and extract residuals.
  Standardize the residuals.
  Shift the data (if necessary).
  Compute the CUSUM chart and calculate the run length with the shifted data.
end for
Compute the average and the standard deviation of the M runs.

In algorithm 3, the data are simuated from an ARMA(1,1) process. We note ϕ the coefficient of the autoregressive part of the model. θ is the coefficient for the moving average part. To this ARMA(1,1) process, an AR(1) model is fitted and the CUSUM chart is calculated by using the residuals of this AR(1) model. The solution described in chapter 6.2 proposes to fit the best ARMA model to the observations. Here we decide to fit only an AR(1) model to the data because we're trying to reproduce a situation where the original data have a correlation structure too complex to be correctly modelled by an ARMA model.

The residuals used in the parametric method don't have the same variance as the original data. In order to have the same variability in the data used to calculate the CUSUM chart, we have to standardise them.

Algorithm 3 explains that the control limit h should be recalibrated for any change in the value of k, ϕ or θ. The recalibration is based on the searching algorithm described in algorithm 1. h is adjusted such as the ARL_0 computed on the residuals of the fitted model reaches the desired value. This step in which h is recalibrated will be the most time consuming in the next simulations. In order to spare some time, the searching algorithm is slightly improved. In the first iterations of the algorithm, the estimated ARL_0 is often far from the targetted value. Since it is far from the reference value, it is unlikely that the algorithm begins to search on the wrong side if the number M of simulations is reduced. In order to speed the algorithm, the number of observation for the two or three first iterations is decreased. For later iterations, we keep the original number of simulations.

Now that we have described the steps for the computation of the ARL_1 by the parametric method, we'll do the same for the block-bootstrap method. The main steps in the computation of the ARL_1 by block-bootstrap are summarized in algorithm 4 below.

Algorithm 4 ARL_1 computed by non-overlapping block-bootstrap

Let M be the number of simulations.
Let N be a large number defining the maximum run length possible.
Give a value for k, ϕ and θ.
Let B be the number of bootstrap replicates.
Let l be the length of each block.
Round N/l down. It is the number of blocks.
For a value of (k, ϕ, θ), use the same value for h than the one computed for the parametric method.

for m in $1, \ldots, M$ **do**
  Simulate a time series from an ARMA(1,1) model with parameters ϕ and θ.
  Standardize the time series.
  Shift the data (if necessary).
  Cut the time series into blocks of length l.
  for b in $1, \ldots, B$ **do**
    Sample with replacement numbers from 1 to $\left\lfloor \frac{N}{l} \right\rfloor$.
    Construct X_b^*.
    Run the CUSUM chart and calculate RL_b^*.
  end for
  Compute the ARL_m and the SDRL_m by calculating the mean and the standard deviation of RL_b^* ($b \in 1, \ldots, B$).
end for
$\frac{1}{M}\sum_{m=1}^{M} \text{ARL}_m$ and $\frac{1}{M}\sum_{m=1}^{M} \text{SDRL}_m$ are the estimations for the ARL and SDRL.

Once a value is fixed for k, ϕ and θ, the control limit h can be found by applying

algorithm 1 on the original data. As for the parametric method, the searching algorithm computes the ARL_0 values by applying the CUSUM chart on the residuals of the fitted AR(1) model. Therefore, the control limit h is the same for the parametric and the block-bootstrap method. Previously, we have seen that the number of bootstrap replicates B doesn't influence the ARL_1 or the SDRL. In order to save time, B is chosen to be 500. It's a number which allows to perform correctly the bootstrap procedure without consuming too much time.

7.2.2.2 First comparison

In the last section, 2 methods for computing the ARL_1 were given. In this part, we establish a first comparison between those 2 methods. In particular, we focus on the effect of ϕ and θ, the parameters of the ARMA model from which data are generated, on the performance of the two methods. We want to estimate the ARL_1 and the SDRL provided by both methods for each value of (ϕ, θ). To do this, a 2-dimensional grid is constructed. ϕ and θ take values in $0.8, 0.6, 0.4, \ldots, -0.8$ so that there is a total of 81 different couples of (ϕ, θ). The results hereafter are obtained after performing $M = 10000$ simulations for both approaches. N is the largest run possible and is chosen to be 2000. The block-length l is 10 so that the time series is made of 200 different blocks. The chart used is designed to detect a shift in the mean of size $\delta = 1.5$. By using the relation $k = \delta/2$, the value for k is 0.75. The control limit h is adjusted so that the reference ARL_0 value of 200 is reached. The values computed for the ARL_1 are plotted on figure 18 for different values of ϕ and θ.

Results

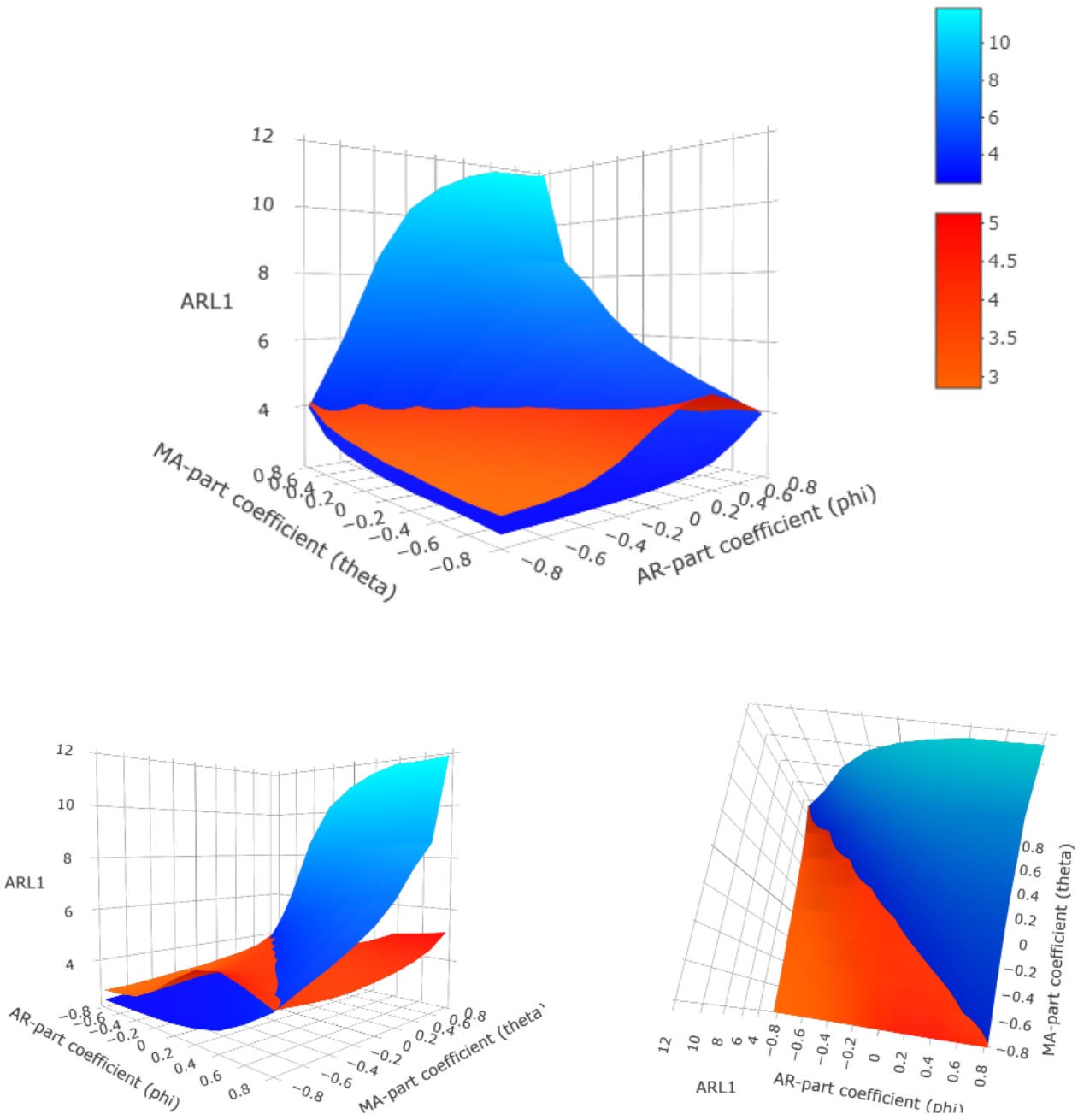

Figure 18: Three 3d plots representing the ARL_1 surfaces depending on ϕ and θ for the two described methods. The blue surface corresponds to the ARL_1 of the CUSUM chart evaluated by block-bootstrap. The red surface corresponds to the ARL_1 evaluated by the parametric approach described in algorithm 3. Parameters used are $k = 0.75$, $\delta = 1.5$ and h is computed such that the reference ARL_0 value is 200. For the block-bootstrap, block of length $l = 10$ are used. The shade in the legend is present to provide a better 3d visualisation.

With the run lengths used to compute the ARL_1, we can also evaluate the SDRL.

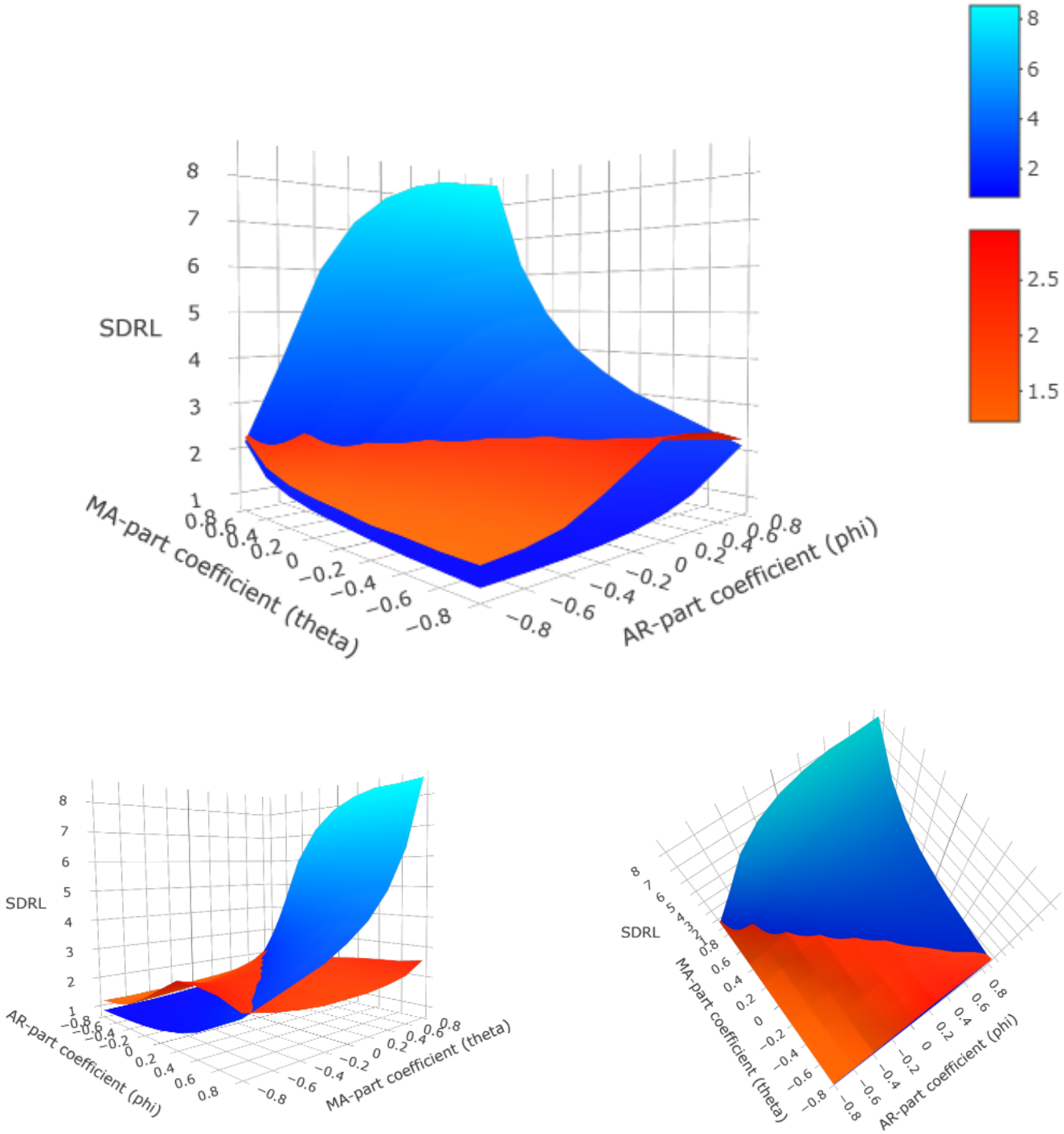

Figure 19: Three 3dplots representing values of the SDRL depending on ϕ and θ. The blue surface corresponds to the SDRL of the CUSUM chart evaluated by block-bootstrap. The red surface corresponds to the $SDRL_1$ evaluated by the parametric approach described in algorithm 3.

Observations

At first glance, we see that the block-bootstrap method doesn't work well when ϕ and θ are close to 1. For instance, the computed ARL_1 is about 12 when $\phi = \theta = 0.8$. Inversely, the ARL_1 computed by block-bootstrap is minimal for ϕ and θ close to -1. The same observation holds for the SDRL (see figure 19). By taking a closer look at the run lengths computed by block-bootstrap when ϕ and θ are close to 1, we realise that most runs are short and some are very long. Actually, the bootstrap series X_b^* are strongly autocorrelated. It can lead to a large number of successive observations below 0. If this happens, the run lengths are large and the ARL_1 is increased. In some sense, what happens here is similar to what happened in the example of figure 13 on page 30 when the autoregressive coefficient is close to 1.

The ARL_1 values computed from the residuals of an Autoregressive model (the red surface) are more stable than the ARL_1 provided by the block-bootstrap method. When $\phi = \theta = 0.8$, the value computed by the parametric method (in red) is smaller than the one provided by block-bootstrap. When the AR(1) model is fitted, it captures a part of the autocorrelation initially present in the data. So, this prevents partially from observing large numbers of successive observations below 0. It implies that the number of long runs is decreased in comparison with block-bootstrap. This explains also why the SDRL computed by the parametric method is smaller than the SDRL provided by block-bootstrap for this value of (ϕ, θ).

In the center of the graph, ϕ and θ are equal to 0. In such a situation, the generated time series are white noise. In this case, extracting the residuals of the fitted AR(1) model doesn't modify significatively the independance structure of the time series. The block-bootstrap procedure samples independent observations drawn from a normal distribution. The bootstrap time series is thus a combination of blocks of white noise. Since we have independent and normally distributed data, the situation is similar to the one described in the "indepent data" part of this chapter (section 7.1). Figure 15 showed that the performances of bootstrap, compared to the reference value computed without bootstrap, are similar for k close to 0.75. Moreover, we had seen that the bias between those two methods decreases when the size of the original sample increases. Here, the size of the original sample is large ($N = 2000$), so it explains also why the difference is small.

The case where ϕ and θ are 0 is not the only point where the ARL_1 provided by the two methods are close. By looking at the last plot of figure 18, we observe the region where the performances are similar is a straight line. When $\phi > -\theta$, the ARL_1 provided by the parametric approach is smaller than the ARL_1 computed by block-bootstrap. The difference gets larger as we get further away from the border. Conversely, when $\phi < -\theta$, the block-bootstrap provides a lower ARL_1 than the parametric approach. Similar conclusions hold for the SDRL.

Important remark

In this first comparison, we compare two ARL_1 values computed by two different methods. In the preceding chapters, the ARL_1 values were compared based on a common ARL_0 value. Here, the control limit h was adjusted such that the ARL_0

value for the parametric method is 200. This control limit value was then reused when we computed the ARL_1 by block-bootstrap. It implies that the ARL_0 computed for block-bootstrap can be very different than the ARL_0 for the parametric method. Since ARL_0 value is different, the rate of false positive alarms is different too.

7.2.2.3 Second comparison

In the last section we compared two ARL_1 values computed by block-bootstrap or by the parametric method (see algorithms 3 and 4). However, this comparison was biased because the two methods didn't have the same rate of false positive alarms. In this section, we introduce another comparison where we fix this problem.

Plan of the comparison

In order to be able to compare two ARL_1 values for a same ARL_0, the following steps are performed.

1. We generate independent data from a standard normal distribution. Assume we are interested in the detection of an upward shift in the mean of size δ. Then, we fix $k = \delta/2$. After that, the control limit h is adjusted by algorithm 1 such that the desired ARL_0 value is obtained (here 200). For the moment, we act like there is no correlation.

2. Now, we generate correlated data from an ARMA(1,1) model. By using k and h previously adjusted, we compute the ARL_0 by using the parametric method described in algorithm 3. The "effective" ARL_0 computed in this step is noted ARL_0^{ef}.

3. Compute the ARL_1 by the parametric method. This value is noted ARL_1^{Par}

4. Calibrate the control limit of the block-bootstrap chart such that the ARL_0 (of the block-bootstrap chart) is close to ARL_0^{ef}. The control limit satisfying this condition is called h^{BB}. This is done by using algorithm 1 and by fixing $A_0 = ARL_0^{ef}$.

5. Compute the ARL_1 by the block-bootstrap method. The ARL_1 is evaluated by applying the CUSUM chart with parameters (k, h^{BB}) on ARMA(1,1) data. This value is noted ARL_1^{BB}.

The only step where the block-bootstrap is used is when we're adjusting the control limit of the block-bootstrap chart. We can now compare ARL_1^{BB} and ARL_1^{Par} on the basis of a common ARL_0 value.

Effect of ϕ and θ

The result section is divided in two parts. In the first part, we discuss how ARL_1^{BB} and $\text{ARL}_1^{\text{Par}}$ depend on ϕ and θ. In the second part, we focus on the effect of δ on ARL_1^{BB} and $\text{ARL}_1^{\text{Par}}$.

ϕ	θ	ARL_0^{Ef}	h^{BB}	$\text{ARL}_1^{\text{Par}}$	ARL_1^{BB}
0.8	0.8	80.44	3.81	4.05	7.84
0.8	0	199.35	6.29	4.00	11.77
0.8	-0.8	194.07	2.42	4.05	3.93
0	0.8	170.49	3.43	3.85	5.64
0	0	197.69	2.43	4.03	3.95
0	-0.8	587.85	2.23	3.60	3.14
-0.8	0.8	193.98	2.43	4.02	3.96
-0.8	0	195.92	1.78	4.05	2.57
-0.8	-0.8	1071.14	2.42	3.50	3.40

Table 4: Comparison of ARL_1^{BB} and $\text{ARL}_1^{\text{Par}}$ for different values of ϕ and θ. The shift size δ is 1.5. k, h and ARL_0 are given by 0.75, 2.48 and 200, respectively. ARL_0^{Ef} is the effective ARL_0 computed by the parametric method. h^{BB} is the control limit adjusted at step 4. Each ARL value was calculated by $M = 10000$ Monte-Carlo simulations. For the block-bootstrap, blocks of length $l = 10$ have been used.

We observe on table 4 that the ARL_0^{Ef} can take values other than 200. We can also see that ARL_0^{Ef} is close to 200 when $\theta = 0$. When $\theta = 0$, the fitted model corresponds with the data generating process. So it is logical to observe $\text{ARL}_0^{\text{Ef}} \approx 200$ for these values. Moreover, we remark that table 4 shows a high correlation between ARL_1^{BB} and h^{BB}. For example, for $\phi = 0.8$ and $\theta = 0$, the block-bootstrap is slow to detect a shift because of the high control limit. The reason why h^{BB} is high for $\phi = 0.8$ and $\theta = 0$ is the sign of the correlation. For an autoregressive process, the i-th lag autocorrelation is given by ϕ^i. So, the autocorrelations are all positive for this value of ϕ and θ. We have seen in exemple 14 page 32 that a positive autocorrelation leads to an higher alarm number, and then a smaller ARL_0 value. When $\phi = -0.8$ and $\theta = 0$, then the i-th lag autocorrelation is given by $(-1)^i(0.8)^i$. In this case, the ARL_0 will be larger than 200 and the control limit h^{BB} will be small.

Performance according to δ

Results shown in table 4 are obtained for the particular value of $\delta = 1.5$. The main objective of this part is to see if the results shown previously can change depending on δ.

We construct a range of values of δ going from 0 to 2. For each value in this range, we begin by fixing $k = \delta/2$. In order to establish a fair comparison, we compute the ARL_0^{Ef}, and then, h^{BB} is adjusted such that the ARL_0 value of the block-bootstrap chart is the same as ARL_0^{Ef}. Finally, ARL_1^{BB} and $\text{ARL}_1^{\text{Par}}$ are computed. Figure 20 shows how ARL_1^{BB} and $\text{ARL}_1^{\text{Par}}$ depend on δ when $\phi = 0.8$ and $\theta = 0.8$.

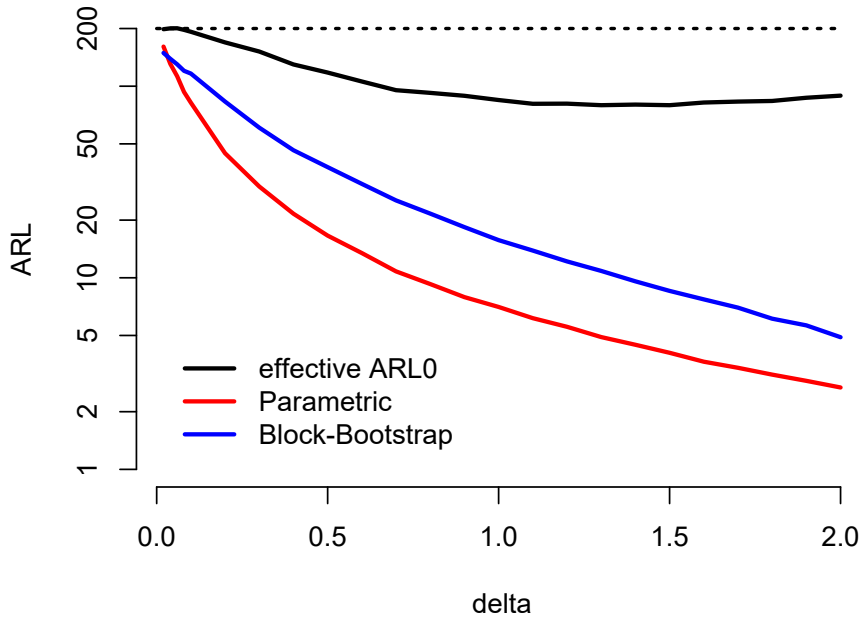

Figure 20: ARL_1^{BB} and $\text{ARL}_1^{\text{Par}}$ for different values of δ when $\phi = 0.8$ and $\theta = 0.8$. The solid black curve represents the effective ARL_0 obtained. The dashed line represents an ARL value of 200. The red curve represents $\text{ARL}_1^{\text{Par}}$, the ARL_1 value computed by the parametric method. The blue curve represents ARL_1^{BB}. The y-axis is in log-scale.

We can see that the parametric method provides better performance than the block-bootstrap regardless of the value of δ. Figure 21 represents also ARL_1^{BB} and $\text{ARL}_1^{\text{Par}}$ for different values of δ. However the ARL values are computed for $\phi = -0.8$ and $\theta = 0$ this time.

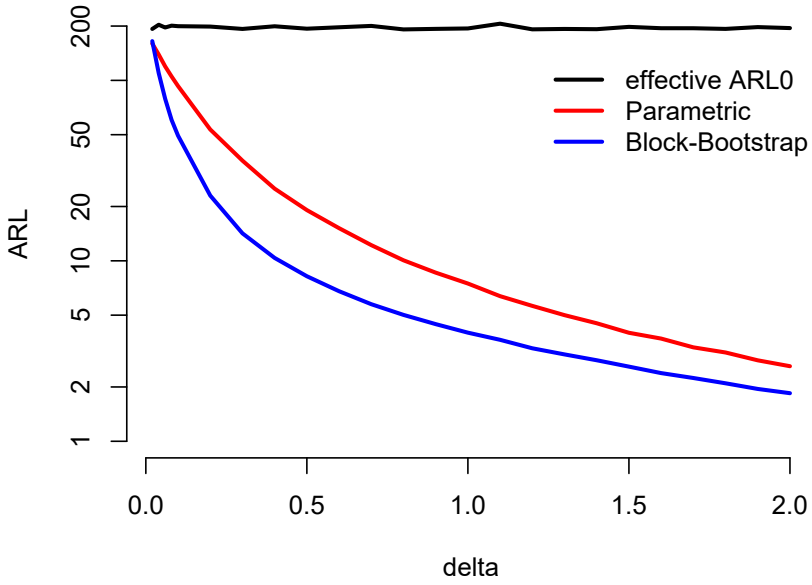

Figure 21: ARL_1^{BB} and ARL_1^{Par} for different values of δ when $\phi = -0.8$ and $\theta = 0$.The solid black curve represents the effective ARL_0 obtained. The red curve represents ARL_1^{Par}, the ARL_1 value computed by the parametric method. The blue curve represents ARL_1^{BB}.

This time the block-bootstrap provides better performance than the parametric method whatever the value of δ. The effective ARL_0 value is 200 since data were generated from an autoregressive process. These two graphs don't show any evidence that the shift size δ influences the choice of the best performing method.

7.2.2.4 Conclusion

In this chapter we tried to compare two possible methods to deal with correlated data. The first method described use residuals of an ARMA model to compute the CUSUM chart. The second method evaluates run length by block-bootstrap. We begin by presenting how to compute the ARL_1 values for those two methods in algorithm 3 and 4. After, that 2 different comparisons of these methods were performed.

The main goal of the first comparison was to evaluate the effect of ϕ and θ on the results provided by the two methods. However, this first comparison has a major drawback : it compares two ARL_1 values for charts who have two different ARL_0. The second comparison is developped to tackle this ARL_0 problem. This second comparison has shown that, for a particular value of δ, the method that works best depends on the sign of the correlation. If the correlation sign is positive, then the parametric method is the best of the two methods. Conversely, if the correlation has a negative sign, then the ARL_1 computed by block-bootstrap is smaller than the ARL_1 computed by the parametric method. The stronger the correlation, the larger the difference in performance. For the methods used in the second comparison, we have also seen that the size of the shift δ doesn't influence which of the 2 methods provides the best performance.

8 Conclusions and discussion

The main goal of this master thesis was to say more about the use of the CUSUM charts in cases where the quality characteristic is non-normally distributed. In this work, we have conducted several different simulation studies which were designed to determine how to calibrate the CUSUM chart in various situations.

Chapter 3 described the CUSUM chart for monitoring the mean of an univariate normal process. Section 3.3 focussed on how to calibrate the chart parameters. We have seen that the first thing to do is to specify a desired ARL_0 value. After that, we need to specify a target shift size δ and then fix the reference value k of the chart to $\delta/2$. Finally, the searching algorithm described in algorithm 1 should be used to compute the control limit h such that the chart with parameters k and h has the desired ARL_0 value. To go further, in many cases, a value for the target shift size δ is not available or cannot be specified accurately. Instead, it might be adequate to define an interval or an union of intervals for the plausible values of δ. In his Ph.D. thesis, Liu [5] constructed a CUSUM chart by accommodating a prior distribution on the potential shift size δ. This prior distribution and the interval bounds should be chosen based on past experiences or scientific knowledge about the production process.

Chapter 4 described the impact of non-normality on CUSUM charts. In this chapter, we assumed that the independence assumption holds. The main goal of this section was to evaluate whether the choice $k = \delta/2$ is still adequate for non-normal continuous distributions. We began by studying the case of a symmetric distribution. Figure 6 showed that the choice $k = \delta/2$ provides very slighty sub-optimal performances. The difference between $k = \delta/2$ and the optimal k is small and then $k = \delta/2$ is an adequate choice for a symmetric distribution. The value for k that minimizes ARL_1 is slightly smaller than $\delta/2$. In the case of an asymmetric distribution, two scenarios are possible. If the mean shift direction is on the low mass side, then the conclusions are similar to the case of a symmetric distribution. The optimal value for k is slightly smaller than $\delta/2$. If the mean shift direction is on the high mass side, then figure 9 showed that $k = \delta/2$ is no longer an appropriate choice. In this case, k should be fixed manually depending on the false positive alarm rate. In a future work, it would be interesting to repeat the simulations carried out for other distributions (uniform, multimodal). In cases when the process distribution is not parametric, Qiu [9] presents nonparametric control charts. For grouped data, Bakir and Reynolds [1] described a nonparametric CUSUM chart with the following charting statistic :

$$C_n^+ = \max\left(0, C_{n-1}^+ + \Psi_n - k\right), \qquad \text{for } n \geq 1. \tag{39}$$

In this statistic, Ψ_n corresponds to the sum of the Wilcoxon signed-ranks within the n-th group of observations.

From chapter 6 on, we focus on cases where the independence assumption doesn't hold. Figure 13 in chapter 6 showed that autocorrelated observations can seriously alter the results given by the CUSUM chart. In order to deal with autocorrelation, we proposed to fit an ARMA model to the autocorrelated observations and then, apply the CUSUM chart to the residuals. If the ARMA adequately models the process observations, then the residuals are close to independent normal data (white noise) and we know that the CUSUM chart will work properly in this case. However, this won't work if the process observations are not appropriate to be modeled by an ARMA. To overcome this constraint, Chapter 7 attempts another approach based on bootstrap.

Before using bootstrap on autocorrelated observations, section 7.1 analysed the influence of bootstrap on normal and independent data. Table 3 showed simulation results where different sizes of bootstrap sample n_B and different number of bootstrap replications B were used. It appeared that the ARL_1 and the SDRL computed by bootstrap are slightly larger than the reference values computed without bootstrap. The size of the overestimation decreases when the size of the bootstrap sample n_B increases. The number of bootstrap replicates B doesn't influence the estimation of the ARL_1 or SDRL. Figure 15 showed that the bias caused by bootstrap reduces when δ increases. If small sample sizes are used, then the bootstrap should not be used when δ is small.

The second part of the chapter presented how the bootstrap can be used on autocorrelated observations. The indepedent bootstrap used in section 7.1 can no longer be used and is replaced by a block-bootstrap method. The block-bootstrap has the advantage to be more polyvalent and adapts to any correlation structure. Also, this section 7.2 compared the results provided by the bootstrap-based method versus the parametric method presented in Chapter 6. Two comparisons were made using results provided by simulations. In order to simulate unperfectly decorrelated residuals, a partially correct model is fitted to the data. Otherwise, the model fitted would be exactly the same as the data generating process and the comparison would be unfair. Both comparisons showed that the sign of the lag 1 autocorrelation of the observations dictates which of the two methods is the best (see figure 18 and table 4). If the lag 1 autocorrelation sign is positive, then the parametric method is the best of the two methods. Conversely, if the correlation has a negative sign, then the ARL_1 computed by block-bootstrap is smaller than the ARL_1 computed by the parametric method. The stronger the first autocorrelation, the larger the difference in ARL_1 between the two methods. In the second comparison, we have also seen that the size of the shift δ doesn't influence which of the two methods provides the best performance.

The major issue of the first comparison is that it compares two ARL_1 values for charts which have different ARL_0 references. The difference is even huge for some

combinations of (ϕ, θ). Since the ARL_0 values are different, the rate of false positive alarms is different too and the comparison is unfair. Also, the arbitrary choice was made to generate data from an ARMA(1,1) model and to decorrelate them using an AR(1) model. The idea behind doing this was to simulate unperfectly decorrelated residuals. In a future work, it can be interesting to see the results for data simulated from an ARMA model with more parameters. Along with that, it would also be interesting to use a more developed model to decorrelate the generated data. Also, more elaborate block-bootstrap techniques exist such as moving block-bootstrap or stationary block-bootstrap. It might be interesting to add them in the comparison to see if the results obtained for the block-bootstrap can be improved.

In order to detect variation caused by special causes, CUSUM charts use all available data that are observed at both the current and earlier time points. *Exponentially Weighted Moving Average* (EWMA) charts work with the same idea. The fundamental difference between CUSUM and EWMA is the way of giving weight to the previous informations. On the one hand, the weights are evenly distributed for CUSUM charts. On the other hand, the weights of the observations decay exponentially for EWMA charts. In a future work, it can be interesting to study to what extent the results obtained in this master thesis can evolve if CUSUM charts are replaced by EWMA charts.

9 Appendix

n_b	B	ARL_1	Bias	SDRL
50	200	8.061	0.297	4.103
	500	8.041	0.276	4.120
	1500	8.037	0.272	4.113
	5000	8.031	0.267	4.118
100	200	7.909	0.145	4.014
	500	7.895	0.130	4.026
	1500	7.899	0.134	4.031
	5000	7.894	0.130	4.030
200	200	7.824	0.060	3.964
	500	7.839	0.074	3.988
	1500	7.821	0.057	3.985
	5000	7.831	0.067	3.987
500	200	7.786	0.022	3.937
	500	7.788	0.024	3.960
	1500	7.795	0.031	3.970
	5000	7.802	0.038	3.971
1000	200	7.771	0.007	3.926
	500	7.778	0.014	3.954
	1500	7.780	0.016	3.958
	5000	7.774	0.009	3.955

Table 5: Effect of bootstrap on the CUSUM chart when, $k = 0.5$, $\delta = 1$ and data are genereted from a t_3 distribution. h is fixed such that the ARL_0 value is 200. The bias is computed as the difference between ARL_1 and 7.764.

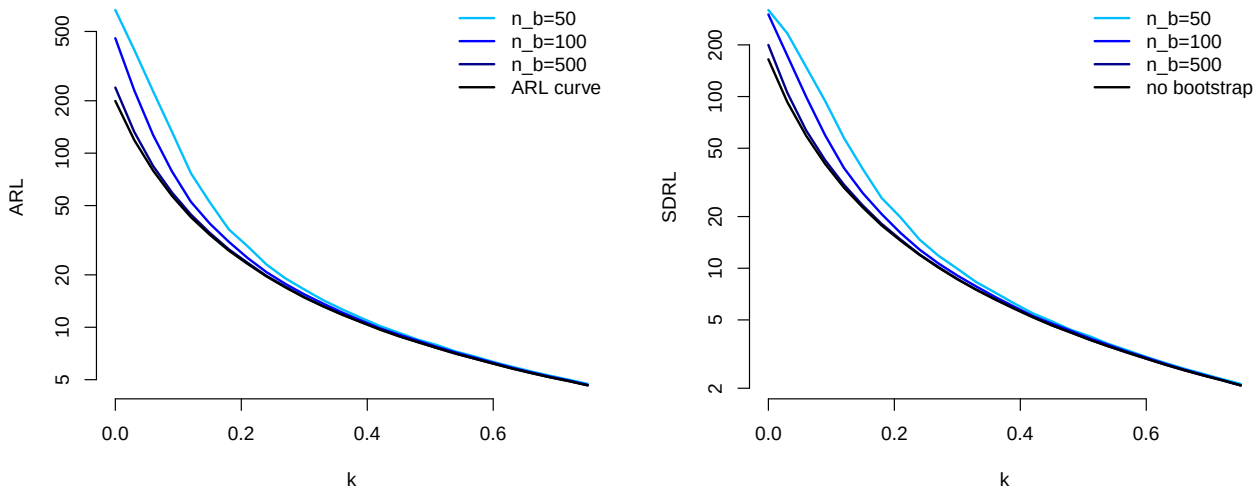

Figure 22: ARL_1 (left) and SDRL (right) estimated by bootstrap for different values of δ and for different sample size n_B when the generated data come from a t_3 distribution. The black curve represents SDRL when no bootstrap is performed. The graph is separated in two parts for a better visualisation. The y-axis is in log-scale.

References

[1] Bakir, S. and Reynolds, M. (1979). A nonparametric procedure for process control based on within group ranking, *Technometrics*, 21, 175-183.

[2] Govaerts, B. (2005). Statistical Process Control, *Syllabus : LSTAT2510*, Louvain-la-Neuve, Belgique.

[3] Kenney, J. and Keeping E. (1951). *Mathematics of Statistics, Part Two*, 2d ed. New York: Van Nostrand. 65-68.

[4] Lahiri, S. (2003). *Resampling Methods for Dependent Data*, Springer, 1 edition.

[5] Liu, W. (2010). *Some Charting Methodologies in MSPC (Ph.D. Thesis)*, School of Statistics, University of Minnesota, Minneapolis.

[6] Lorden, G. (1971). Procedures for reacting to a change in distribution. *The Annals of Mathematical Statistics*, 42, 1897-1908.

[7] Moustakides, G. (1986). Optimal stopping times for detecting changes in distribution, *The Annals of Statistics*, 14 (4), 1379-1387.

[8] Page, E.S. (1954). Continuous inspection scheme, *Biometrika* 41, 100-115.

[9] Qiu, P. (2013). *Introduction to Statistical Process Control*, Florida : CRC Press.

List of Figures

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Faculté des sciences
Place des sciences, 2 bte L6.06.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/sc**