Computing the Effect of Runs Rules on Time-Between-Event Control Charts

Corneille, Noé

Award date:
2021

Computing the Effect of Runs Rules on Time-Between-Event Control Charts

Bachelor final project

N. Corneille
Student ID: 1223165

June 8, 2021

Contents

1 Problem Description

Statistical process control (SPC) plays a critical role in manufacturing industry, as it provides a set of tools for process monitoring and quality improvement. Among these techniques, the control chart is the most widely used tool. In industrial settings, an increasing number of applications of SPC involve measuring non-conforming items (i.e. defect items), this is particularly relevant in high-yield or high-purity processes. These processes are characterized by a low defect rate. This defect rate is low enough that the central limit theorem cannot be used effectively. Traditional SPC techniques like Shewhart $\bar{X}$-charts (explained in Chapter 3) or Individual charts are known to perform sub par in high-yield processes. Therefore dedicated control charts such as the Times Between Events (TBE) control charts have been developed for both discrete time (see Calvin (1983), Goh (1987)) and continuous time (see Xie et al. (2002a)) to detect any shifts in the process defect rate. A detailed overview on TBE charts can be found in Ali et al. (2016).

The most commonly used performance metric is the average run length (ARL), which is explained in Chapter 2. However, the ARL has two shortcomings: First, since the run length distributions are skewed, means are not sufficient and additional summary statistics like the variance (see e.g., Di Bucchianico et al. (2005)) or conditional expectations such as the conditional expected delay (CED)) (see Kenett and Pollak (2012) and Frisén (2009)) should be utilized. Secondly, run length distributions for TBE charts ignore that we have two time scales (the number of items and the number of control chart decisions). For TBE charts it is therefore common to consider inspection lengths. Shewhart-$\bar{X}$-charts are often slow in detecting when a process is deemed out of control. To combat this slowness, Western Electric Company (1956) determined some extra rules for determining whether a process is out of control, these rules are called the run rules. Rizzo et al. (2020) reviewed and extended the performance measures of Kenett and Pollak (2012) and Frisén (2009) to continuous and discrete TBE Shewhart-type charts with run rules via Monte Carlo simulations. In this study, a remarkable phenomenon is the non-monotone distributional behavior of the CED. The application of different run rules influences the shape of the run length distribution. Therefore, it is important to evaluate and compare the effect of different run rules and to suggest a practical guideline on the choice of run rules.

The goals of this research project are:

1. Implementation of Markov Chains approach to correctly compute the control limits of TBE charts with runs rules. This is required in order to make fair comparisons between control charts.

2. Numerical simulations to compare the effects of different run rules on the distribution of the run length.

2 Background in matrix algebra and real analysis

Before diving into SPC, it is necessary to build up a foundation of theorems and lemmas from matrix algebra and real analysis. This chapter provides a solid toolbox from which the necessary formulae can be derived in order to efficiently proceed with SPC.

2.1 Matrix calculus

Due to the high level of complexity as a result of the so called runs rules to be introduced later, many calculations involving SPC involve matrices. In particular the product- and chain rule for taking derivatives need to be extended to matrices (see Chapter 4 of Gentle (2007)). Define $\mathbb{R}^{n\times m} := \{\mathbf{M} \text{ an } n\times m \text{ matrix with entries in } \mathbb{R}\}$, note that $n\times m$ does not denote the standard integer product but should rather be a notational convenience for denoting the set of all matrices (similar to how $n\times m$ and nm have different meanings when discussing the dimensions of a matrix); the isomorphism $\mathbb{R}^{n\times m} \cong \mathbb{R}^{nm}$ does however hold through considering a bijection from an $n\times m$ matrix to an nm-dimensional vector. Suppose $\mathbf{A} : \mathbb{R} \to \mathbb{R}^{\mathrm{a}\times\mathrm{m}}, \mathbf{B} : \mathbb{R} \to \mathbb{R}^{\mathrm{m}\times\mathrm{b}}$ are functions which map scalars to matrices, where $a, b, m \in \mathbb{N}$ are constants such that the product $\mathbf{AB}$ exists and is equal to some function $\mathbf{AB} : \mathbb{R} \to \mathbb{R}^{\mathrm{a}\times\mathrm{b}}$. It then holds, analogous to the product rule on functions which map scalars to scalars, that

$$(\mathbf{AB})' = \mathbf{B}\ \mathbf{A}' + \mathbf{A}\ \mathbf{B}', \tag{1}$$

where $\mathbf{M}'$ denotes the derivative of a matrix-valued function $\mathbf{M}$ defined on $\mathbb{R}$. This derivative is defined component wise i.e. $(M')_{ij} = (M_{ij})'$. A particularly useful result of this is when a matrix function is multiplied by vectors to create a scalar function. Let $v \in \mathbb{R}^n$, and let $\mathbf{M} : \mathbb{R} \to \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$, in particular suppose $v' = 0$. Then as $\mathbb{R}^n \cong \mathbb{R}^{n\times 1}$ through the identity function, formula (1) can be applied twice to yield the following identity

$$\begin{aligned}(v^T\mathbf{M}v)' &= (v^T\mathbf{M})'v + (v^T\mathbf{M})v' \\ &= ((v^T)'\mathbf{M} + v^T\mathbf{M}')v + 0 \\ &= (0 + v^T\mathbf{M}')v = v^T\mathbf{M}'v.\end{aligned} \tag{2}$$

Another useful tool is the ability to express derivatives of inverse matrices in terms of derivatives of non-inverted matrices, this can be achieved using the following lemma.

Lemma 2.1. *Suppose* $\mathbf{M} : \mathbb{R} \to \mathrm{GL}_n(\mathbb{R}) := \{\mathbf{M}$ *an* $n\times n$ *matrix of real entries* $\mid \mathbf{M}$ *invertible*$\}$*, i.e.* $\mathbf{M}$ *is an n×n invertible matrix function of 1 real variable. Then the following holds for all* $x \in \mathbb{R}$*:*

$$(\mathbf{M}(x)^{-1})' = -\mathbf{M}(x)^{-1}\mathbf{M}'(x)\mathbf{M}(x)^{-1}. \tag{3}$$

We use $\mathbf{M}$, $\mathbf{M}^{-1}$ to denote $\mathbf{M}(x), (\mathbf{M}(x))^{-1}$ for arbitrary $x \in \mathbb{R}$; note that in particular $\mathbf{M}^{-1}$ does not denote the inverse function from $\mathrm{GL}_n(\mathbb{R})$ to $\mathbb{R}$, but rather the function from $\mathbb{R}$ to $\mathrm{GL}_n(\mathbb{R})$ mapping x to $(\mathbf{M}(x))^{-1}$.

Proof. Let I_n denote the $n\times n$ identity matrix and let $\mathbf{0}_{n\times m}$ denote an $n\times m$ matrix of zero entries. Suppose $\mathbf{M} : \mathbb{R} \to \mathrm{GL}_n(\mathbb{R})$, then $I_n = \mathbf{M}^{-1}\mathbf{M}$. Note that $I_n' = \mathbf{0}_{n\times n}$.
Using Formula (1) it then follows that

$$\begin{gathered}\mathbf{0}_{n\times n} = (\mathbf{M}^{-1})'\mathbf{M} + \mathbf{M}^{-1}\mathbf{M}' \\ \Longleftrightarrow \\ (\mathbf{M}^{-1})'\mathbf{M} = -\mathbf{M}^{-1}\mathbf{M}' \\ \Longleftrightarrow \\ (\mathbf{M}^{-1})' = -\mathbf{M}^{-1}\mathbf{M}'\mathbf{M}^{-1}\end{gathered}$$

Therefore it is indeed true that for all $x \in \mathbb{R}$:

$$(\mathbf{M}(x)^{-1})' = -\mathbf{M}(x)^{-1}\mathbf{M}'(x)\mathbf{M}(x)^{-1}.$$

□

In order to prove some important theorems in Section 4.1 we will need Equation 8 of Gerschgorin (1931), it is also referred to as the Gershgorin Circle Theorem.

Theorem 2.2 (Gershgorin Circle Theorem)**.** *Let $\mathbf{M}$ be a general $n \times n$ matrix with components $M_{ij} \in \mathbb{C}$. Then the eigenvalues of $\mathbf{M}$ are located in the union of circles*

$$|z - M_{ii}| \leq -|M_{ii}| + \sum_{j=1}^{n} |M_{ij}| = S_i - |M_{ii}|,$$

where $z \in \mathbb{C}$. Here we define the ith absolute row sum of $\mathbf{M}$ as $S_i := \sum_{j=1}^{n} |M_{ij}|$. Equivalently: Let $z \in \mathbb{C}$, and $r \in \mathbb{R}$; define $B(z, r) \subset \mathbb{C}$ to be the open ball on the complex plane of radius r centered at z. Let $\overline{A}$ denote the closure of a set A. Then for all eigenvalues λ of $\mathbf{M}$ it holds

$$\lambda \in \bigcup_{i=1}^{n} \overline{B(M_{ii},\ S_i - |M_{ii}|)}.$$

Proof. See Theorem 6.6.1 of Horn and Johnson (1990), or Theorem 2.12 of Zhang (2011). □

This method of determining bounds for the eigenvalues of a matrix can be used to determine bounds on the determinant.

Lemma 2.3. *Let $\mathbf{M}$ be a complex $n \times n$ matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$. It then holds*

$$\det \mathbf{M} = \prod_{i=1}^{n} \lambda_i.$$

Proof. See Section 3.8 (in particular Formula 3.179) of Gentle (2007). □

When calculating expectations involving Markov chains like we will do in Section 4.1, it is useful to have some theory involving series of matrices. The series we will look at specifically are geometric ones. In order to say something about the convergence of matrix series we first need to define the notion of a norm (see Definition 1.1 in Chapter 3.1 of Conway (1994)) and in particular a matrix norm (see Section 4.1 of Zhang (2011)).

Definition 2.4. *Let V be a vector space over a field $\mathbb{F}$. A norm is a function $\|.\| : V \to [0, \infty)$ with the following properties*

1. *For $\lambda \in \mathbb{F}$, $\mathbf{v} \in V$: $\|\lambda \mathbf{v}\| = |\lambda| \, \|\mathbf{v}\|$.*
2. *For $\mathbf{v}, \mathbf{w} \in V$: $\|\mathbf{v} + \mathbf{w}\| \leq \|\mathbf{v}\| + \|\mathbf{w}\|$.*
3. *For $\mathbf{v} \in V$: $\|\mathbf{v}\| = 0$ if and only if $\mathbf{v} = 0$.*

If property 3 does not hold, $\|.\|$ is called a seminorm. When a basis is chosen for V, a norm on V is also called a vector norm.
A fourth property named the submultiplicative (or consistency) property is sometimes needed on vector spaces that allow multiplication of elements.

4. *For $\mathbf{v}, \mathbf{w} \in V$ (where $\mathbf{vw}$ exists and is an element of V), $\|\mathbf{vw}\| \leq \|\mathbf{v}\| \, \|\mathbf{w}\|$.*

A norm with this property is called a submultiplicative norm.

Let V, W be vector spaces. We denote the set of linear transformations between V and W by $\mathrm{Hom}(V, W)$, as linear transformations are homomorphisms between vector spaces. Denote the set of endomorphisms on V by $\mathrm{End}(V) := \mathrm{Hom}(V, V)$. It is important to note that once we choose a basis for V and W, $\mathrm{Hom}(V, W)$ becomes the set of matrices; for example $\mathrm{End}(\mathbb{C}^n) = \mathbb{C}^{n \times n}$, when $\mathbb{C}^n$ is equipped with the standard basis.

Definition 2.5. *Let V be a vector space for which a basis is chosen such that $\mathrm{End}(V)$ is a set of matrices. A submultiplicative norm $\|.\|$ on $\mathrm{End}(V)$ is then called a matrix norm.*

Definition 2.6. *Let $\|.\|_V$ be a norm on a vector space V. The norm $\|.\| : \mathrm{End}(V) \to [0, \infty)$ induced by $\|.\|_V$ is defined as follows*

$$\|\mathbf{M}\| := \sup \left\{ \frac{\|\mathbf{Mv}\|_V}{\|\mathbf{v}\|_V} \;\middle|\; \mathbf{v} \in V \setminus \{0\} \right\}.$$

Note that this definition implies that $\|\mathbf{Mv}\|_V \leq \|\mathbf{M}\| \, \|\mathbf{v}\|_V$ for all $\mathbf{v} \in V \setminus \{0\}$.

Lemma 2.7. *Let $\|.\|_V$ be a norm on a vector space V, the induced norm $\|.\|$ on $\mathrm{End(V)}$ is a submultiplicative norm.*

Proof. By Definition 2.5 it holds that

$$\|\mathbf{M}\| = \sup \left\{ \frac{\|\mathbf{Mv}\|_V}{\|\mathbf{v}\|_V} \;\middle|\; \mathbf{v} \in V \setminus \{0\} \right\} = \sup_{\|\mathbf{v}\|_V = 1} \|\mathbf{Mv}\|_V \, .$$

But then

$$\|\mathbf{AB}\| = \sup_{\|\mathbf{v}\|_V = 1} \|\mathbf{ABv}\|_V \leq \sup_{\|\mathbf{v}\|_V = 1} \|\mathbf{A}\| \, \|\mathbf{Bv}\|_V \leq \|\mathbf{A}\| \, \|\mathbf{B}\| \, .$$

So the fourth property of Definition 2.5 is applicable. The 3 properties of Definition 2.4 hold trivially as $\|.\|_V$ is a norm by definition. □

If a basis is chosen for V, this lemma implies that a norm on $\mathrm{End}(V)$ induced by a vector norm is a matrix norm. We can now prove a lemma that helps us prove convergence of series for the specific matrices used in Section 4.1.

Lemma 2.8. *Define the function $\|.\| : \mathbb{C}^{n \times n} \to [0, \infty)$ by*

$$\|\mathbf{M}\| := \max_{1 \leq i \leq n} \sum_{j=1}^{n} |M_{ij}|,$$

where M_{ij} denotes the (i,j)th component of $\mathbf{M}$. Then $\|.\|$ is a matrix norm.

Proof. The proof of this lemma is based on Example 5.6.5 of Horn and Johnson (1990). Consider the matrix norm $\|.\|_\infty$ induced by the vector norm $\|\mathbf{v}\|_\infty := \max_i |v_i|$ on $\mathbb{C}^n$. Then

$$\begin{aligned} \|\mathbf{Mv}\|_\infty &= \max_{1 \leq i \leq n} \left| \sum_{j=1}^{n} M_{ij} v_j \right| \\ &\leq \max_{1 \leq i \leq n} \sum_{j=1}^{n} |M_{ij}||v_j| \\ &\leq \|\mathbf{v}\|_\infty \max_{1 \leq i \leq n} \sum_{j=1}^{n} |M_{ij}| = \|\mathbf{v}\|_\infty \, \|\mathbf{M}\| \, . \end{aligned}$$

This implies that $\|\mathbf{M}\|_\infty \leq \|\mathbf{M}\|$. Let $\mathbf{w} \in \mathbb{C}^n$ such that $\|\mathbf{w}\|_\infty = 1$. Then there exists a $\theta_i \in [0, 2\pi)$ such that for each component of $\mathbf{w}$ it holds $w_i = e^{\theta_i \iota}$, where ι denotes the imaginary unit. This θ_i is also referred to as $\mathrm{Arg}(w_i)$. Let k be the index where $\mathbf{M}$ has a maximum absolute row sum, then we can choose $w_i = e^{-\mathrm{Arg}(M_{kj})\iota} = M_{kj}^* / |M_{kj}|$ (where * denotes complex conjugation). This definition implies that $|M_{ij}| = M_{ij} w_i$. Then as $\|\mathbf{w}\|_\infty = 1$, it holds

$$\begin{aligned} \|\mathbf{M}\|_\infty &\geq \frac{\|\mathbf{Mw}\|_\infty}{\|\mathbf{w}\|_\infty} \\ &= \max_{1 \leq i \leq n} \left| \sum_{j=1}^{n} M_{ij} w_j \right| \\ &\geq \left| \sum_{j=1}^{n} M_{kj} w_j \right| \\ &= \left| \sum_{j=1}^{n} |M_{kj}| \right| \\ &= \sum_{j=1}^{n} |M_{kj}| = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |M_{ij}|. \end{aligned}$$

We conclude that for all $\mathrm{M} \in \mathbb{C}^{n \times n}$, $\|\mathbf{M}\| = \|\mathbf{M}\|_\infty$. This in turn implies that $\|.\|$ must be a matrix norm as a result of Lemma 2.7 since it is equal to a norm that is induced by a vector norm. □

Theorem 2.9. *Let $\mathbf{M}$ be a complex square matrix such that $\mathbf{I}-\mathbf{M}$ is invertible, then*

$$\sum_{i=0}^{n}\mathbf{M}^i=(\mathbf{I}-\mathbf{M})^{-1}(\mathbf{I}-\mathbf{M}^{n+1}).$$

Proof. Define $\mathbf{S}_n:=\sum_{i=0}^{n}\mathbf{M}^i$, then $\mathbf{M}\mathbf{S}_n=\sum_{i=0}^{n}\mathbf{M}^{i+1}=\sum_{i=1}^{n+1}\mathbf{M}^i$. We conclude that

$$(\mathbf{I}-\mathbf{M})\mathbf{S}_n=\sum_{i=0}^{n}\mathbf{M}^i-\sum_{i=1}^{n+1}\mathbf{M}^i=\mathbf{I}-\mathbf{M}^{n+1}.$$

We conclude that $\mathbf{S}_n=(\mathbf{I}-\mathbf{M})^{-1}(\mathbf{I}-\mathbf{M}^{n+1})$ if $\mathbf{I}-\mathbf{M}$ is invertible. □

Under certain favourable conditions, this theorem can be applied to infinite series.

Corollary 2.9.1. *Assume $\|\mathbf{M}\|<1$, where $\|.\|$ is any matrix norm. Under the conditions of Theorem 2.9 it then holds that*

$$\sum_{i=0}^{\infty}\mathbf{M}^i=(\mathbf{I}-\mathbf{M})^{-1}.$$

Proof. Suppose the above mentioned conditions hold. The fourth property from Definition 2.5 can be applied n times to yield $\|\mathbf{M}^n\|\leq\|\mathbf{M}\|^n$, for all integers $n>1$. Then $\|\mathbf{M}\|<1$ implies that

$$0\leq\lim_{n\to\infty}\|\mathbf{M}^n\|\leq\lim_{n\to\infty}\|\mathbf{M}\|^n=0.$$

So $\|\mathbf{M}^n\|$ converges to 0 by applying the squeeze theorem (Theorem 2.2.6 of Kosmala (2004)). We conclude that the set of accumulation points of the sequence $(\mathbf{M}^n)_{n\in\mathbb{N}}$ is a nonempty subset of $\{x\in\mathbb{C}^{n\times n}\mid\|x\|=0\}$. As a result of property 3 of Definition 2.4, this set of accumulation points contains only the zero element of $\mathbb{C}^{n\times n}$. Therefore $\lim_{n\to\infty}\mathbf{M}^n$ exists and is identical to the zero matrix. Thus the components of $\mathbf{M}^n$ to all go to 0 as $n\to\infty$. We conclude, as a result of Theorem 2.9, that:

$$\begin{aligned}\sum_{i=0}^{\infty}\mathbf{M}^i&=\lim_{n\to\infty}\sum_{i=1}^{n}\mathbf{M}^i\\&=\lim_{n\to\infty}(\mathbf{I}-\mathbf{M})^{-1}(\mathbf{I}-\mathbf{M}^{n+1})\\&=(\mathbf{I}-\mathbf{M})^{-1}(\mathbf{I}-\lim_{n\to\infty}\mathbf{M}^{n+1})\\&=(\mathbf{I}-\mathbf{M})^{-1}.\end{aligned}$$

□

The condition that $\mathbf{I}-\mathbf{M}$ is invertible can be omitted in Corollary 2.9.1 when the matrix norm $\|.\|$ is induced by a vector norm. This is because the maximum absolute eigenvalue (spectral radius) of $\mathbf{M}$ is less than or equal to $\|\mathbf{M}\|<1$ (see Corollary 1.6 of Varga (1962)). We will give a full proof of this in section 4.1 (see Lemma 4.2).

2.2 Summation by parts

Since many theorems from SPC deal with expectations of discrete random variables, a concise theory of evaluating series is required. In order to prove a theorem in Section 2.3 we need Abel's Lemma, also called summation by parts. This lemma can be seen as the discrete equivalent of integration by parts.

Lemma 2.10. *Let $\{a_n\}_{n\geq0}$, $\{b_n\}_{n\geq0}$ be arbitrary real sequences, and denote the partial sum of a_n by*

$$A_n=\sum_{k=0}^{n}a_k.$$

It then holds that

$$\sum_{k=0}^{n}a_kb_k=A_nb_{n+1}+\sum_{k=0}^{n-1}A_k(b_k-b_{k+1}).\tag{4}$$

Proof. A proof of this lemma can be found as Lemma 3.11.1 of Little et al. (2016). □

It is particularly useful to note that

$$\sum_{k=0}^{\infty} A_k(b_{k+1} - b_k) = \lim_{n\to\infty} A_n b_{n+1} - \sum_{k=0}^{\infty} a_k b_k, \tag{5}$$

if $\sum_{k=0}^{\infty} A_k(b_{k+1} - b_k)$ or $\sum_{k=0}^{\infty} a_k b_k$ converges.

2.3 Probability theory

To be able to efficiently calculate the average run length (ARL) of a Shewhart chart in Chapter 3, we first need the following lemma.

Lemma 2.11. *Let X be a non-negative discrete random variable defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$. It then holds that:*

$$\mathbb{E}[X] = \sum_{i=0}^{\infty} \mathbb{P}(X > i). \tag{6}$$

This lemma can be proven in two ways, the first way involves Tonelli's theorem (see Theorem 3.4.5 of Bogachev (2007)) and is written in a purely measure theoretic way, the second one uses summation by parts (Lemma 2.10).

Proof. Recall the definition of the expectation of X:

$$\mathbb{E}[X] := \int_{\Omega} X(x)\, \mathbb{P}(\mathrm{d}x),$$

as X is integrable w.r.t. $\mathbb{P}$. Note that as $\mathbb{P}$ is a probability measure, $(\Omega, \mathcal{F}, \mathbb{P})$ and $(\mathbb{N}_0, 2^{\mathbb{N}_0}, \#)$ (where $\#$ denotes the counting measure) are σ-finite measure spaces. Furthermore

$$\mathbb{1}_{(i,\infty)}(X) = \mathbb{1}_{(i,\infty)} \circ X,$$

is a composition of measurable functions, therefore it is $\# \otimes \mathbb{P}$-measurable. It also holds

$$\begin{aligned}
\mathbb{P}(X > i) &= \mathbb{P}(X \in (i, \infty)) \\
&= \int_{\Omega} \mathbb{1}_{(i,\infty)}(X(x))\, \mathbb{P}(\mathrm{d}x) \\
&\leq \int_{\Omega} X(x)\, \mathbb{P}(\mathrm{d}x) < \infty.
\end{aligned}$$

We conclude, using the notational convention $\sum_{i=0}^{\infty} f(i) = \int_{\mathbb{N}_0} f(i)\, \#(\mathrm{d}i)$, that:

$$\begin{aligned}
\sum_{i=0}^{\infty} \mathbb{P}(X > i) &= \sum_{i=0}^{\infty} \int_{\Omega} \mathbb{1}_{(i,\infty)}(X(x))\, \mathbb{P}(\mathrm{d}x) \\
&= \int_{\Omega} \sum_{i=0}^{\infty} \mathbb{1}_{(i,\infty)}(X(x))\, \mathbb{P}(\mathrm{d}x) \quad \text{(Tonelli's theorem)} \\
&= \int_{\Omega} X(x)\, \mathbb{P}(\mathrm{d}x) - 0 \cdot \mathbb{P}(x = 0) \\
&= \mathbb{E}[X].
\end{aligned}$$

□

An alternative proof to this lemma can be found using Lemma 2.10.

Proof. Let $n \in \mathbb{N}_0$, and define the sequences $a_n := \mathbb{P}(X = n)$, and $b_n = n$. Note that this implies that $A_n = \mathbb{P}(X \leq n) = 1 - \mathbb{P}(X > n)$, where $n \in \mathbb{N}_0$. The series $\sum_{k=0}^{\infty} k\mathbb{P}(X = k)$ converges as X is defined to have a finite mean. Thus according to (5) it holds that:

$$\sum_{k=0}^{\infty}(1 - \mathbb{P}(X > k))(k + 1 - k) = \lim_{n\to\infty}[\mathbb{P}(X \leq n)(n + 1)] - \sum_{k=0}^{\infty} k\mathbb{P}(X = k)$$

$$\iff$$

$$\lim_{n\to\infty}(n + 1) + \sum_{k=0}^{\infty}\mathbb{P}(X > k) = \lim_{n\to\infty}[\mathbb{P}(X \leq n)(n + 1)] - \sum_{k=0}^{\infty} k\mathbb{P}(X = k)$$

$$\iff$$

$$-\sum_{k=0}^{\infty}\mathbb{P}(X > k) = \lim_{n\to\infty}[(\mathbb{P}(X \leq n) - 1)(n + 1)] - \sum_{k=0}^{\infty} k\mathbb{P}(X = k)$$

$$\iff$$

$$\sum_{k=0}^{\infty}\mathbb{P}(X > k) = \lim_{n\to\infty}[\mathbb{P}(X > n)(n + 1)] + \sum_{k=0}^{\infty} k\mathbb{P}(X = k)$$

Note that

$$0 \leq \mathbb{P}(X > n)(n + 1) = (n + 1)\sum_{k=n+1}^{\infty}\mathbb{P}(X = k) \leq \sum_{k=n+1}^{\infty} k\mathbb{P}(X = k). \tag{7}$$

As X has a finite mean, $\sum_{k=n+1}^{\infty} k\mathbb{P}(X = k)$ approaches 0 as $n \to \infty$. Therefore $\lim_{n\to\infty}\mathbb{P}(X > n)(n + 1) = 0$ by the squeeze theorem. □

Later proofs also require to provide a solid definition for the conditional expectation. Definition 1 of Chapter 9 in Roussas (2004) provides a definition for the conditional expectation:

Definition 2.12. *Let X be a random variable on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ for which $\mathbb{E}[|X|] < \infty$. The conditional expectation $\mathbb{E}[X \mid X \in A]$ is defined as follows:*

$$\begin{aligned}\mathbb{E}[X \mid X \in A] &= \frac{1}{\mathbb{P}(X \in A)}\int_A X(x)\mathbb{P}(\mathrm{d}x)\\ &= \frac{1}{\mathbb{P}(X \in A)}\int_\Omega X(x)\mathbb{1}_A(x)\mathbb{P}(\mathrm{d}x)\\ &= \mathbb{E}[X\mathbb{1}_A].\end{aligned}$$

2.4 Stopping time

In order to effectively monitor using control charts, we need a stopping criterion which does not depend on future observations. The mathematical object that does exactly this is called a stopping time, which is defined formally in the following way (see section 10.3 of Bogachev (2007)):

Definition 2.13. *Let $(\Omega, \mathcal{F}, (\mathcal{F}_n)_{n\in\mathbb{N}_0}, \mathbb{P})$ be a filtered probability space. Then a random variable $N : \Omega \to \mathbb{N}_0$ is called a stopping time if for all $n \geq 0$ it holds that $\{N = n\} \in \mathcal{F}_n$.*

This definition is very abstract and not very useful, however in Rizzo et al. (2020) a more advantageous definition is used. Let $\mathcal{F}_n = \sigma(X_i \mid i \leq n)$, for some iid sequence of random variables X_i. Here $\sigma(X_i \mid i \leq n)$ refers to the sigma-algebra (or event space) generated by $\{X_1, ..., X_n\}$. This conceptually means that the only information available is that which is generated by random variables in the past. Using this definition for $\mathcal{F}_n$, Definition 2.13 can be refactored into the following definition.

Definition 2.14. *Let $X_1, X_2, ...$ be an iid sequence of random variables. A random variable N is called a stopping time if for each $n \geq 0$ the event $\{N = n\}$ depends only on $X_1, ..., X_n$.*

3 Shewhart Control Charts

In Section 3.1 we explain the basics of control charts as used in statistical process control. In Section 3.2 a calculation for the average run length is shown in the case where no runs rules are present. These runs rules are elaborated on in Section 3.3.

3.1 The basic idea of Shewhart control charts

In SPC, control charts are used to detect changes of measurable quantities from some expected behaviour. As these charts monitor the behaviour of these quantities, the term "control chart" is technically the wrong name for such a chart; this is the term that is utilised nevertheless. In order to monitor a measurable quantity, first the distribution of the quantity needs to be estimated through statistical models. When this distribution is known, the theory can be applied. Shewhart charts are the most simple type of control chart. These charts are reliant on the quantities being measured having a normal distribution. Literature often utilises Shewhart-$\bar{X}$ charts, these are a special kind of Shewhart chart where the measurements are first grouped into so called "rational subgroups" of size n. In this text the measurements are not grouped when dealing with Shewhart charts (so $n = 1$).

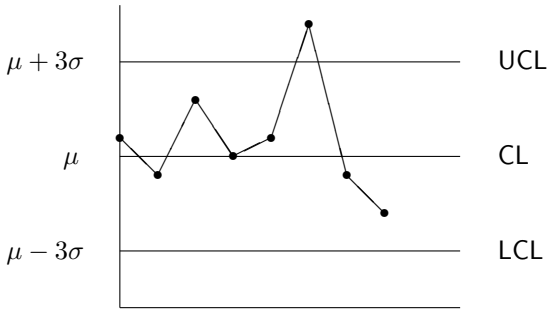

Figure 1: Shewhart chart with control lines.

The goal of Shewhart charts is determining whether a process is in control (IC) or out of control (OC). Since this chart detects changes in a normal distribution, the measurements which we will refer to as X_i with $i \in \{1, ..., N^*\}$ are characterized as follows:

$$X_i \sim \begin{cases} \mathcal{N}(\mu, \sigma^2) & \text{if } i < N \\ \mathcal{N}(\mu^*, (\sigma^*)^2) & \text{if } i \geq N \end{cases}, \tag{8}$$

as an OC signal implies that the expected distribution of the measurements has changed. However since we do not want to measure OC signals, the model can be reformulated in a simpler manner. Rather that taking $i \in \{1, ..., N^*\}$, we only consider $i \in \{1, \ldots, N-1\}$ which implies that for all i, $X_i \sim \mathcal{N}(\mu, \sigma^2)$. To determine whether or not the process is in control, we use the so called "control limits". We deem a process OC when a measurement falls outside of the region between the upper control limit (UCL) and lower control limit (LCL), these limits can be seen in Figure 1, and are often defined to be the $\mu \pm 3\sigma$. The random variable N is defined to be the first index of measurements for which the process is OC, in Figure 1 it would hold $N = 6$ assuming the left-most point has index 1. This random variable can be defined similar to definition 2 in Rizzo et al. (2020):

$$N := \begin{cases} \min\{n \geq 1 \mid X_n \text{ is an OC signal}\} \\ N^* \text{ if } X_i \text{ IC for all } i. \end{cases} \tag{9}$$

Due to the minimum function, the event of N being equal to n^* will indeed only depend on $X_1, ..., X_{n^*}$. We can therefore conclude that N is indeed a stopping time according to Definition 2.14. It is assumed that the process will go out of control at some point therefore this definition can be simplified to

$$N := \min\{n \geq 1 \mid X_n \text{ is an OC signal}\}. \tag{10}$$

When no runs rules are present the event $\{X_n \text{ is an OC signal}\}$ is equivalent to X_n being outside of the interval from LCL to UCL. Since runs rules change what this event encapsulates the general definition

above is given. When no runs rules are used it holds:

$$N := \min\{n \geq 1 \mid X_n \in \mathbb{R} \setminus [\mathrm{LCL}, \mathrm{UCL}]\}. \tag{11}$$

When dealing with Shewhart charts, one often works with the so called run length (RL). The run length is defined as the probability distribution of the stopping time N. The average run length (ARL) is defined as $\mathrm{ARL} := \mathbb{E}[N]$. The following section will deal with the calculations involved in determining this expectation.

3.2 Average run length without runs rules

Let $X \sim \mathcal{N}(\mu, \sigma^2)$, and define $\mathrm{UCL} := \mu + n\sigma$, $\mathrm{LCL} := \mu - n\sigma$, $n \in \mathbb{R}_{>0}$, which is more general than the regular definitions. Then (10) becomes

$$N = \min\{n \geq 1 \mid X_n \in \mathbb{R} \setminus [\mu - n\sigma, \mu + n\sigma]\}. \tag{12}$$

Suppose the mean of the sample is shifted by some amount $\delta\sigma$ i.e. $X \sim \mathcal{N}(\mu_0 + \delta\sigma, \sigma^2)$, which also implies that for $X' := (X - \mu_0)/\sigma$ it holds that;

$$X' \sim \mathcal{N}(\delta, 1),$$

furthermore $\delta = \frac{|\mu - \mu_0|}{\sigma}$. Let $X' - \delta =: X^* \sim \mathcal{N}(0, 1)$, then

$$\begin{aligned} \mathbb{P}(X \in (0, \mathrm{UCL})) &= \mathbb{P}(X' \in (0, n)) \\ &= \mathbb{P}(X^* \in (-\delta, n - \delta)) \\ &= \Phi(n - \delta) - \Phi(-\delta). \end{aligned} \tag{13}$$

Similarly:

$$\mathbb{P}(X \in (\mathrm{LCL}, 0)) = \Phi(-\delta) - \Phi(-(n + \delta)) = \Phi(n + \delta) - \Phi(\delta), \tag{14}$$

$$\mathbb{P}(X \in (-\infty, \mathrm{LCL})) = \Phi(-(n + \delta)) = 1 - \Phi(n + \delta), \tag{15}$$

$$\mathbb{P}(X \in (\mathrm{UCL}, \infty)) = 1 - \Phi(n - \delta). \tag{16}$$

Note that since X is normally distributed, $\{0\}$, $\{\mathrm{LCL}\}$, and $\{\mathrm{UCL}\}$ are null sets with respect to the image-measure $\mathbb{P} \circ X^{-1}$, therefore

$$\mathbb{P}(X \in \{0\}) = \mathbb{P}(X \in \{\mathrm{LCL}\}) = \mathbb{P}(X \in \{\mathrm{UCL}\}) = 0,$$

which removes the need to include clopen sets into (13)-(16). Other formulae can be derived using the same method, in general;

$$\mathbb{P}(X \in (\mu + a, \mu + b)) = \mathbb{P}(\sigma(X^* + \delta) \in (a, b)) = \Phi(b/\sigma - \delta) - \Phi(a/\sigma - \delta). \tag{17}$$

For the case without runs-rules, it holds that N is geometrically distributed with parameter

$$p = \mathbb{P}(X \in \mathbb{R} \setminus [\mathrm{LCL}, \mathrm{UCL}]) = 1 - \mathbb{P}(X \in (\mathrm{LCL}, \mathrm{UCL})).$$

Note that this implies that $\mathbb{P}(N = n) = (1 - p)^{n-1}p$, and $\mathbb{P}(N > n) = (1 - p)^n$. It then holds that the ARL for the in control situation is equal to the expectation of N. Using (6) we deduce that

$$\mathrm{ARL} = \sum_{i=0}^{\infty} (1 - p)^i = \frac{1}{p}. \tag{18}$$

An alternative way of finding the ARL is through probability generating functions. Let G_N denote the probability generating function of N, i.e.,

$$G_N(z) = \sum_{i=0}^{\infty} \mathbb{P}(N = i) z^i.$$

As the mean of N is also the 1^{st} factorial moment of N, it holds that

$$\mathrm{ARL} = \lim_{z \uparrow 1} G'_N(z) \tag{19}$$

3.3 Runs rules

Shewhart control charts are good for quickly detecting "large" changes of the mean (order of magnitude 2.5 standard deviations or more) but not good at quickly detecting smaller changes or trends. This is because Shewhart control charts have no memory, i.e., they use only the current observation. In order to increase the detection performance of Shewhart charts, heuristic extra stopping rules on top of the standard rule of points being outside of the interval $[\mathrm{LCL}, \mathrm{UCL}]$ have been proposed. Such stopping rules are known under the name runs rules. A well-known set of runs rules are the so-called Western Electric Rules documented in Western Electric Company (1956).

Suppose now we use the standard definition of a Shewhart chart i.e. X_i are iid samples equal to some distribution $X \sim \mathcal{N}(\mu, \sigma)$. Analogously to the definitions $\mathrm{LCL} = \mu - 3\sigma$, $\mathrm{UCL} = \mu + 3\sigma$, we define several other variables to characterise regions used for the runs rules. Let

$$\begin{aligned} \mathrm{LWL} &= \mu - 2\sigma, & \mathrm{UWL} &= \mu + 2\sigma, \\ \mathrm{LIWL} &= \mu - \sigma, & \mathrm{UIWL} &= \mu + \sigma. \end{aligned}$$

We denote the sets corresponding to each zone with $C,\ W,\ I$, more specifically

$$\begin{aligned} \mathrm{C_L} &= [\mu - 3\sigma, \mu), \quad \mathrm{C_U} = [\mu, \mu + 3\sigma], \\ \mathrm{W_L} &= [\mu - 2\sigma, \mu), \quad \mathrm{W_U} = [\mu, \mu + 2\sigma], \\ \mathrm{I_L} &= [\mu - \sigma, \mu), \quad \mathrm{I_U} = [\mu, \mu + \sigma]. \end{aligned}$$

The original rules introduced in Western Electric Company (1956) are:

1. One point outside of $C_L \cup C_U$;
2. Two consecutive points on either side of $W_L \cup W_U$;
3. Four consecutive points on either side of $I_L \cup I_U$;
4. Eight consecutive points on either side of the center line ($x = \mu$).

These rules are sometimes supplemented with three additional rules,

5. Fifteen consecutive points in $I_L \cup I_U$ (This is called "stratification", it means that the observed process has smaller variance than the control limits. If this is really the case, then it means that an improvement has taken place. However, it may also be due to bad sampling.)
6. Eight consecutive points on either side of the center line, with no points inside $I_L \cup I_U$ (This is a symmetry break, so it may indicate a shift.)
7. A series of out-of-control points in the lower zones followed by a series of out-of-control points in the upper zones or vice versa. A series of points without a change in direction (This may b e an indication of a trend.).

Koutras et al. (2007) suggests a general notation for these runs rules. An out of control signal is given if the k of the last m standardized sample statistics X' fall within the interval (a, b), where $k \leq m$ and $a \leq b$; we then denote this by $X'(k, m, a, b)$. In the special case that $k \neq m$, these rules are called scans rules rather than runs rules. The runs rules as described above are then given by:

1. $C_1 = \{X'(1, 1, -\infty, -3), X'(1, 1, 3, \infty)\}$;
2. $C_2 = \{X'(2, 2, -3, -2), X'(2, 2, 2, 3)\}$;
3. $C_3 = \{X'(4, 4, -3, -1), X'(4, 4, 1, 3)\}$;
4. $C_4 = \{X'(8, 8, -3, 0), X'(8, 8, 0, 3)\}$;
5. $C_5 = \{X'(15, 15, -1, 1)\}$;
6. $C_6 = \{X'(8, 8, -3, -1), X'(8, 8, 1, 3)\}$;
7. C_7 is not expressible in terms of the suggested notation.

These runs rules may be combined, the notation for this is $C_{ij...k} = C_i \cup C_j \cup ... \cup C_k$. An equivalent notation can be defined as follows; let $\mathcal{I} \subset \mathbb{N}_0$ an index set such that for all $i \in \mathcal{I}$ the runs rule C_i exists, then $C_{\mathcal{I}} = \bigcup_{i \in \mathcal{I}} C_i$. It is important to note that the events generated by these runs rules should only depend on already known data points, so that these rules can be made into stopping times. If we view the rules defined above as sets of events, (10) can be generalized by limiting the runs rules to see past events only. Suppose the set of all events generated by the runs rules is given by $C_{\mathcal{I}}$, we can then limit this to past events by considering $\mathcal{S}_n := C_{\mathcal{I}} \cap \sigma(X_i \mid i \leq n)$. The stopping time is then defined as

$$N := \min \left\{ i \;\middle|\; \bigcup_{S \in \mathcal{S}_i} S \right\}. \tag{20}$$

To perform meaningful calculations on this N, more advanced techniques than already described are needed; these techniques are the topic of the next section.

4 Run Lengths for Shewhart Charts with Runs Rules

In Section 3.2 the ARL for Shewhart charts without runs rules is calculated. When runs rules are involved, the difficulty of calculating the ARL increases rapidly. Note that due to their definition, Shewhart charts are inherently memoryless. To deal with this, memory of the previous states observations in a Shewhart chart can be encoded into a Markov chain.

4.1 Theory for general runs rules

Define the discrete-time Markov chain $(Y_t \mid t \in \mathbb{N}_0) \in \{0, \ldots, n\}$, which is designed to reflect the runs rules as defined in section 3.3. The first state of this Markov chain always represents the safe state, and the last state represents a point outside of $[\mathrm{LCL}, \mathrm{UCL}]$ or an OC signal as a result of a runs rule. This last state is thus an absorbing state.

In general the transition probability matrix of this Markov chain has the following form (see Brook and Evans (1972)):

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{R} & \mathbf{c} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{21}$$

since the row sums of this matrix should be equal to 1, it must hold that $\mathbf{c} = (\mathbf{I} - \mathbf{R})\mathbf{1}$. Note that since $\mathbf{\Lambda}$ is a stochastic matrix, the rows of $\mathbf{I} - \mathbf{\Lambda}$ sum to 0. This implies that 0 is an eigenvalue of $\mathbf{I} - \mathbf{\Lambda}$ as $(\mathbf{I} - \mathbf{\Lambda})\mathbf{1} = 0\mathbf{1}$. We conclude that $\mathbf{I} - \mathbf{\Lambda}$ has a nontrivial null-space and is therefore not invertible. Since the matrix $\mathbf{I} - \mathbf{R}$ is invertible in at least some cases, it is more convenient for calculating the ARL. The situation in which a point outside of $[\mathrm{LCL}, \mathrm{UCL}]$ is found will always yield an OC signal, therefore $\mathbf{c}$ is nonzero everywhere as it implies that the absorbing state is reachable from any state. This also implies that the row sums of R are always strictly smaller than 1. Due to the nature of the runs rules, the only states in the Markov chain which are linked to themselves are the first and last states. Since a lot of formulae for the ARL are based on the convergence of geometric series involving $\mathbf{R}$, we need the following lemmas to prove that these series converge.

Lemma 4.1. *Let $\mathbf{M}$ be a real-valued $n \times n$ matrix with components $M_{ij} \in [0, 1)$ such that the row sums of $\mathbf{M}$ are all strictly smaller than 1. For all eigenvalues $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$ of $\mathbf{M}$ it then holds that $|\lambda_i| < 1$ for all $1 \leq i \leq n$.*

Proof. Using Theorem 2.2 we can find bounds on the absolute values of the individual eigenvalues of $\mathbf{R}$. Let $z \in \mathbb{C}$, and $r \in \mathbb{R}$; define $B(z, r) \subset \mathbb{C}$ to be the open ball on the complex plane of radius r centered at z. Let $\overline{A}$ denote the closure of a set A. By assumption the components M_{ij} of $\mathbf{M}$ are real and positive. This implies that the absolute values used in Theorem 2.2 can be ignored i.e. $|M_{ij}| = M_{ij}$. Denote the i-th row sum of $\mathbf{M}$ by S_i, we know then that $S_i < 1$ for all $1 \leq i \leq n$.

Let $1 \leq i \leq n$, and suppose $z \in \overline{B(M_{ii}, S_i - M_{ii})}$. Then $|z - M_{ii}| \leq S_i - M_{ii}$, but then also $|z - M_{ii}|^2 \leq (S_i - M_{ii})^2$. Note that $|z - M_{ii}|^2 = (\mathrm{Re}(z) - M_{ii})^2 + \mathrm{Im}(z)^2 = |z|^2 - 2\mathrm{Re}(z)M_{ii} + M_{ii}^2$. Therefore:

$$|z|^2 - 2\mathrm{Re}(z)M_{ii} + M_{ii}^2 \leq (S_i - M_{ii})^2 = S_i^2 - 2S_iM_{ii} + M_{ii}^2,$$

equivalently:

$$|z|^2 \leq S_i^2 - 2(S_i - \mathrm{Re}(z))M_{ii}.$$

As $\mathrm{Re}(z) \leq M_{ii} + S_i - M_{ii} = S_i$, it holds that $S_i - \mathrm{Re}(z) \geq 0$; therefore $S_i^2 - 2(S_i - \mathrm{Re}(z))M_{ii} \leq S_i^2$. We conclude that $|z|^2 \leq S_i^2$. Because $S_i < 1$ the following inclusion holds:

$$\overline{B(M_{ii}, S_i - M_{ii})} \subseteq \overline{B(0, S_i)} \subset B(0, 1).$$

As this holds for every $1 \leq i \leq n$, the following inclusion must also hold

$$\bigcup_{i=1}^{n} \overline{B(M_{ii}, S_i - M_{ii})} \subseteq \bigcup_{i=1}^{n} \overline{B(0, S_i)} \subset B(0, 1)$$

We conclude that for all $1 \leq i \leq n$, $\lambda_i \in B(0, 1)$ as a result of Theorem 2.2. Equivalently $|\lambda_i| < 1$, for all $1 \leq i \leq n$.

□

Lemma 4.2. *Let $\mathbf{M}$ be a complex $n \times n$ matrix of which every eigenvalue is not equal to $\frac{1}{z}$, with $z \in \mathbb{C} \setminus \{0\}$. Then $\mathbf{I} - z\mathbf{M}$ is invertible.*

Proof. As $\mathbf{M}$ is a square matrix, it has eigenvalues $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$. These eigenvalues are the roots of $\det(\mathbf{M} - \lambda\mathbf{I})$ by definition. It is assumed that $|\lambda_i| \neq \frac{1}{z}$ for all $1 \leq i \leq n$. As $z\mathbf{M} - \mathbf{I} = z\left(\mathbf{M} - \frac{1}{z}\mathbf{I}\right)$, the eigenvalues of $z\mathbf{M} - \mathbf{I}$ are z times those of $\mathbf{M} - \frac{1}{z}\mathbf{I}$. The eigenvalues of the latter matrix are the roots of $\det\left(\mathbf{M} - \frac{1}{z}\mathbf{I} - \lambda\mathbf{I}\right) = \det\left(\mathbf{M} - (\lambda + \frac{1}{z})\mathbf{I}\right)$. Since z is assumed to be nonzero, it can be seen using the variable transformation $\lambda \mapsto \lambda - \frac{1}{z}$ that the roots of these determinants are equal to $\lambda_1 - \frac{1}{z}, \ldots, \lambda_n - \frac{1}{z}$. But as $\lambda_i \neq \frac{1}{z}$ it must hold that $\lambda_i - \frac{1}{z} \neq 0$ for all $1 \leq i \leq n$. Equivalently the eigenvalues of $z\mathbf{M} - \mathbf{I}$, given by $z\lambda_i - 1$ for $1 \leq i \leq n$, are also nonzero. We can conclude using Lemma 2.3 that

$$\begin{aligned}\det(\mathbf{I} - z\mathbf{M}) &= -\det(z\mathbf{M} - \mathbf{I}) \\ &= -\prod_{i=1}^{n}(z\lambda_i - 1) \neq 0.\end{aligned}$$

□

Using these lemmas it is possible to prove that $\mathbf{I} - \mathbf{R}$ is invertible.

Theorem 4.3. *Let $\mathbf{M}$ be a matrix with row-sums smaller than 1, and components $M_{ij} \in [0, 1)$. It then holds that $\mathbf{I} - \mathbf{M}$ is invertible.*

Proof. Because of the conditions set on $\mathbf{M}$, Lemma 4.1 implies that for each eigenvalue λ of $\mathbf{M}$ it holds that $|\lambda| < 1$. Hence using Lemma 4.2 with $z = 1$ we conclude that $\mathbf{I} - \mathbf{M}$ is invertible. □

Note that the matrix $\mathbf{R}$ derived from our Markov chain has the properties required for Theorem 4.3, therefore $\mathbf{I} - \mathbf{R}$ is guaranteed to be invertible.

Define the vectors $\boldsymbol{\pi}_0 \in [0, 1]^{n+1}$, $\boldsymbol{\rho}_0 \in [0, 1]^n$ by

$$\boldsymbol{\pi}_0^T := (\mathbb{P}(Y_0 = 0), \mathbb{P}(Y_0 = 1), \ldots, \mathbb{P}(Y_0 = n)), \tag{22}$$

$$\boldsymbol{\rho}_0^T := (\mathbb{P}(Y_0 = 0), \mathbb{P}(Y_0 = 1), \ldots, \mathbb{P}(Y_0 = n - 1)), \tag{23}$$

and let $\{\mathbf{e}_0, \ldots \mathbf{e}_n\}$ denote the standard basis of $\mathbb{R}^{n+1}$ i.e.

$$\mathbf{e}_n = (0, \ldots, 0, 1)^T.$$

For Shewhart charts without runs rules, we start in the safe region, so that $P(Y_0 = 0) = 1$. However, for charts with memory like Shewhart charts with runs rules, it might be convenient to start in another state in order to increase the detection speed of the procedure. The idea is that when the process is in control, the chart will return by itself to the safe state, while in an out-of-control situation the chart is already closer to a state that will signal. This idea has been introduced in the context of CUSUM control charts in Lucas and Crosier (2000). In Koutras et al. (2007) it is derived that

$$\mathbb{P}(N > k) = 1 - \mathbb{P}(Y_k = n) = 1 - \boldsymbol{\pi}_0^T \boldsymbol{\Lambda}^k \mathbf{e}_n, \tag{24}$$

and therefore

$$\begin{aligned}\mathbb{P}(N = k) &= \mathbb{P}(N > k - 1) - \mathbb{P}(N > k) \\ &= \mathbb{P}(Y_k = n) - \mathbb{P}(Y_k = n - 1) \\ &= \boldsymbol{\pi}_0^T \left(\boldsymbol{\Lambda}^k - \boldsymbol{\Lambda}^{k-1}\right) \mathbf{e}_n \\ &= \boldsymbol{\pi}_0^T \boldsymbol{\Lambda}^{k-1} \left(\boldsymbol{\Lambda} - I\right) \mathbf{e}_n.\end{aligned} \tag{25}$$

In Brook and Evans (1972) calculations are done using the submatrix $\mathbf{R}$. As a result of Theorem 4.3 formulae with geometric series involving an $\mathbf{R}^k$ term simplify to taking the inverse, this is not possible with formulae involving a $\boldsymbol{\Lambda}^k$ term since $\mathbf{I} - \boldsymbol{\Lambda}$ is guaranteed to have an eigenvalue equal to 0. The following proposition let us do calculations using this more convenient submatrix. Theorem 4.3 in combination with the following propositions provide a proof for the mathematical validity of the formulae used in Brook and Evans (1972).

Proposition 4.4. *The right hand side of Formula (25) is equal to $\boldsymbol{\rho}_0^T \mathbf{R}^{k-1}(\mathbf{I} - \mathbf{R})\mathbf{1}$.*

Proof. According to Section 2.1 of Zhang (2011) block matrices can be multiplied as they were regular matrices (in Zhang (2011) this is called a type III operation). We can deduce that

$$\mathbf{\Lambda}^2 = \begin{bmatrix} \mathbf{R}^2 + 0 & \mathbf{R}\mathbf{c} + \mathbf{c} \\ \mathbf{0}^T\mathbf{R} + 1 \cdot \mathbf{0}^T & \mathbf{0}^T\mathbf{c} + 1 \cdot 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}^2 & \sum_{i=0}^{1} \mathbf{R}^i\mathbf{c} \\ \mathbf{0}^T & 1 \end{bmatrix}.$$

This will be used for the base case of an inductive argument. Suppose now that

$$\mathbf{\Lambda}^k = \begin{bmatrix} \mathbf{R}^k & \sum_{i=0}^{k-1} \mathbf{R}^i\mathbf{c} \\ \mathbf{0}^T & 1 \end{bmatrix}.$$

Then it holds that:

$$\begin{aligned} \mathbf{\Lambda}^{k+1} &= \mathbf{\Lambda}\mathbf{\Lambda}^k \\ &= \begin{bmatrix} \mathbf{R}\mathbf{R}^k + 0 & \mathbf{R}\sum_{i=0}^{k-1} \mathbf{R}^i\mathbf{c} + \mathbf{c} \\ \mathbf{0}^T\mathbf{R}^k + 1 \cdot \mathbf{0}^T & \mathbf{0}^T\sum_{i=0}^{k-1} \mathbf{R}^i\mathbf{c} + 1 \cdot 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}^{k+1} & \sum_{i=0}^{k} \mathbf{R}^i\mathbf{c} \\ \mathbf{0}^T & 1 \end{bmatrix}, \end{aligned}$$

therefore the assumption holds for all $k \in \mathbb{N}_0$. Using the fact that $\mathbf{c} = (\mathbf{I} - \mathbf{R})\mathbf{1}$ it can be concluded that

$$\begin{aligned} \mathbf{\Lambda}^k - \mathbf{\Lambda}^{k-1} &= \begin{bmatrix} \mathbf{R}^k - \mathbf{R}^{k-1} & \mathbf{R}^{k-1}(\mathbf{I} - \mathbf{R})\mathbf{1} \\ \mathbf{0}^T & 0 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}^k - \mathbf{R}^{k-1} & -(\mathbf{R}^k - \mathbf{R}^{k-1})\mathbf{1} \\ \mathbf{0}^T & 0 \end{bmatrix} \end{aligned}$$

Therefore it holds that

$$\left(\mathbf{\Lambda}^k - \mathbf{\Lambda}^{k-1}\right)\mathbf{e}_n = \begin{pmatrix} \mathbf{R}^{k-1}(\mathbf{I} - \mathbf{R})\mathbf{1} \\ 0 \end{pmatrix}$$

Hence we can conclude that

$$\boldsymbol{\pi}_0^T\left(\mathbf{\Lambda}^k - \mathbf{\Lambda}^{k-1}\right)\mathbf{e}_n = \boldsymbol{\rho}_0^T\mathbf{R}^{k-1}(\mathbf{I} - \mathbf{R})\mathbf{1}$$

□

From these equations the ARL can be calculated using Lemma 2.11. Following the proof of Proposition 4.4, it can be derived that

$$\mathbf{\Lambda}^k\mathbf{e}_n = \sum_{i=0}^{k-1} \mathbf{R}^i(\mathbf{I} - \mathbf{R})\mathbf{1}. \tag{26}$$

Intuitively we do not want our process to start with an OC signal. This implies that the last component of $\boldsymbol{\pi}_0$ is equal to 0. Using this information the following can be derived about the case where our process is not allowed to start OC,

$$\begin{aligned} 1 &= \boldsymbol{\pi}_0^T\mathbf{1} \\ &= \boldsymbol{\rho}_0^T\mathbf{1} + 0 = \boldsymbol{\rho}_0^T\mathbf{1}. \end{aligned}$$

Using this equality, a convenient formula for the ARL exists because $(\mathbf{I} - \mathbf{R})$ is invertible.

Proposition 4.5. *When $\boldsymbol{\rho}_0^T\mathbf{1} = 1$, it holds that*

$$\mathrm{ARL} = \boldsymbol{\rho}_0^T(\mathbf{I} - \mathbf{R})^{-1}\mathbf{1}.$$

Proof. As $(\mathbf{I} - \mathbf{R})$ is invertible by Theorem 4.3, according to Theorem 2.9 it holds that (26) is equivalent to

$$\mathbf{\Lambda}^k\mathbf{e}_n = (\mathbf{I} - \mathbf{R})^{-1}(\mathbf{I} - \mathbf{R}^k)(\mathbf{I} - \mathbf{R})\mathbf{1}$$

Let $\|\mathbf{R}\|_\infty := \max_j \sum_i |R_{ij}|$ denote the row-sum norm from Lemma 2.8. We know that this norm is a matrix norm. As $\mathbf{R}$ only has nonnegative entries this norm is equal to the largest row sum of $\mathbf{R}$.

These row sums are strictly smaller than 1, therefore $\|\mathbf{R}\|_\infty < 1$. Hence Corollary 2.9.1 implies that $\sum_{k=0}^{\infty} \mathbf{R}^k = (\mathbf{I}-\mathbf{R})^{-1}$. Using Lemma 2.11 we can then conclude that

$$\begin{aligned}
\mathrm{ARL} &= \sum_{k=0}^{\infty} \mathbb{P}(N > k) \\
&= \sum_{k=0}^{\infty} 1 - \boldsymbol{\rho}_0^T \boldsymbol{\Lambda}^n \mathbf{e}_0 \\
&= \sum_{k=0}^{\infty} 1 - \boldsymbol{\rho}_0^T (\mathbf{I}-\mathbf{R})^{-1}(\mathbf{I}-\mathbf{R}^k)(\mathbf{I}-\mathbf{R})\mathbf{1} \\
&= \sum_{k=0}^{\infty} 1 - \boldsymbol{\rho}_0^T \mathbf{1} + \boldsymbol{\rho}_0^T (\mathbf{I}-\mathbf{R})^{-1}\mathbf{R}^k(\mathbf{I}-\mathbf{R}) \\
&= \sum_{k=0}^{\infty} \boldsymbol{\rho}_0^T (\mathbf{I}-\mathbf{R})^{-1}\mathbf{R}^k(\mathbf{I}-\mathbf{R}) \\
&= \boldsymbol{\rho}_0^T (\mathbf{I}-\mathbf{R})^{-1} \left(\sum_{k=0}^{\infty} \mathbf{R}^k\right)(\mathbf{I}-\mathbf{R}) \\
&= \boldsymbol{\rho}_0^T (\mathbf{I}-\mathbf{R})^{-1}\mathbf{1}.
\end{aligned}$$

□

This proposition can be seen as an extension of Formula 3.3 of Brook and Evans (1972), since it shows how to calculate the ARL from the vector $(\mathbf{I}-\mathbf{R})^{-1}\mathbf{1}$.
In Koutras et al. (2007) the ARL is found in a different manner by taking the first derivative of the probability generating function of N evaluated at 1. This probability generating function is equal to

$$\sum_{k=0}^{\infty} \mathbb{P}(N=k) z^k = z\boldsymbol{\pi}_0^T (\mathbf{I} - z\boldsymbol{\Lambda})^{-1}(\boldsymbol{\Lambda}-\mathbf{I})\mathbf{e}_n, \tag{27}$$

under the condition that Λ has no eigenvalues equal to z (see Lemma 4.2). Let $f : \mathbb{R} \to \mathbb{R}$ be defined by $f(z) := \boldsymbol{\pi}_0^T (\mathbf{I} - z\boldsymbol{\Lambda})^{-1}(\boldsymbol{\Lambda}-\mathbf{I})\mathbf{e}_n$. The power series can then be rewritten in the form

$$\sum_{k=0}^{\infty} \mathbb{P}(N=k) z^k = zf(z). \tag{28}$$

The product rule then implies that the derivative of this series is equal to

$$\frac{\mathrm{d}}{\mathrm{d}z} \sum_{k=0}^{\infty} \mathbb{P}(N=k) z^k = f(z) + zf'(z).$$

Using (1),(2), lemma (2.1), and the fact that $\frac{\mathrm{d}}{\mathrm{d}z}(\mathbf{I} - z\boldsymbol{\Lambda}) = -\boldsymbol{\Lambda}$, we can derive:

$$f'(z) = -\boldsymbol{\pi}_0^T (\mathbf{I} - z\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}(\mathbf{I} - z\boldsymbol{\Lambda})^{-1}(\boldsymbol{\Lambda}-\mathbf{I})\mathbf{e}_n.$$

Hence we conclude from Theorem 8.5.15 of Kosmala (2004) that:

$$\begin{aligned}
\sum_{k=0}^{\infty} k\mathbb{P}(N=k) z^{k-1} &= \frac{\mathrm{d}}{\mathrm{d}z} \sum_{k=0}^{\infty} \mathbb{P}(N=k) z^k \\
&= \boldsymbol{\pi}_0^T (\mathbf{I} - z\boldsymbol{\Lambda})^{-1}(\boldsymbol{\Lambda}-\mathbf{I})\mathbf{e}_n - z\boldsymbol{\pi}_0^T (\mathbf{I} - z\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}(\mathbf{I} - z\boldsymbol{\Lambda})^{-1}(\boldsymbol{\Lambda}-\mathbf{I})\mathbf{e}_n \\
&= \boldsymbol{\pi}_0^T (\mathbf{I} - z\boldsymbol{\Lambda})^{-1}\left[I - z\boldsymbol{\Lambda}(\mathbf{I} - z\boldsymbol{\Lambda})^{-1}\right](\boldsymbol{\Lambda}-\mathbf{I})\mathbf{e}_n.
\end{aligned}$$

As $\mathbf{I}-\boldsymbol{\Lambda}$ is not invertible, is not possible to simply evaluate the probability generating function derived above at $z = 1$. Thus, in contrast with Proposition 4.5, there is no closed form expression for the ARL in terms of $\boldsymbol{\Lambda}$. Since the geometric series for the probability generating function converges for $|z| < 1$, the closest we can get to a closed formula is

$$\mathrm{ARL} = \lim_{z \uparrow 1} \boldsymbol{\pi}_0^T (\mathbf{I} - z\boldsymbol{\Lambda})^{-1}\left[I - z\boldsymbol{\Lambda}(\mathbf{I} - z\boldsymbol{\Lambda})^{-1}\right](\boldsymbol{\Lambda}-\mathbf{I})\mathbf{e}_n. \tag{29}$$

As $\mathbf{\Lambda}$ is a stochastic matrix, its eigenvalues lie inside the circle $\overline{B(0,1)}$ as a result of Theorem 2.2. As the set of eigenvalues of $\mathbf{\Lambda}$ is finite, there exists an $\varepsilon > 0$ such that $\mathbf{I} - z\mathbf{\Lambda}$ is invertible for z in the punctured half circle $(B(1,\varepsilon) \setminus \{1\}) \cap \{z \in \mathbb{C} \mid \mathrm{Re}(z) < 1\}$. Hence the $(\mathbf{I} - z\mathbf{\Lambda})^{-1}$ terms are valid inside the limit in (29). Note that calculating the ARL using the general form derived through this method is significantly more computationally intensive than using Proposition 4.5.

4.2 ARL for run rule C_2

Suppose we are looking at a Shewhart chart using only the 2nd run rule C_2 as defined in Section 3.3, we use the model for X_i from Formula (8). This implies that the process is deemed out of control when: for some $n \in \mathbb{N}$, $X_n \in (-\infty, \mathrm{LCL}) \cup (\mathrm{UCL}, \infty)$; or there are two consecutive points in the upper or lower warning areas $(\mathrm{LCL}, \mu - 2\sigma), (\mu + 2\sigma, \mathrm{UCL})$. The safe area is characterised by $(\mu - 2\sigma, \mu + 2\sigma)$.

We define the Markov chain $(Y_t \mid t \in \mathbb{N}_0) \in \{0, 1, 2, 3\}$. The first state $(Y_i = 0)$ represents a point in the safe area, the second and third states ($Y_i = 1$ or $Y_i = 2$) represent the first point in the upper or lower warning area respectively, and the last state $(Y_i = 3)$ represents an OC process.

Let $p_k^{(i)} := \mathbb{P}(Y_i = k)$, as X_i are iid, so are Y_i therefore the i in this definition can be omitted, and Y, X are used to refer to the distributions.

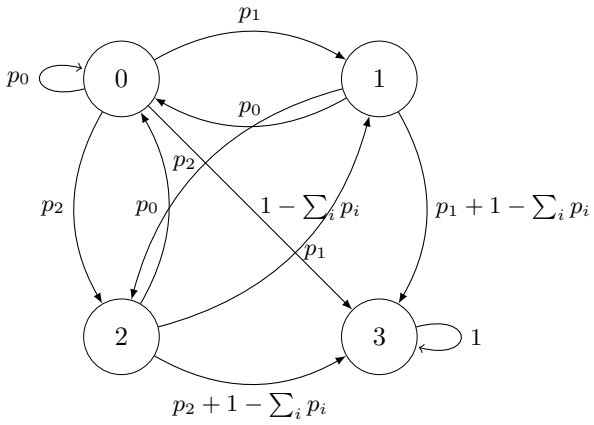

Figure 2: State diagram for the Markov chain when using run rule C_2

The transition probability matrix for the Markov chain can be derived from Figure 2 and is equal to:

$$\mathbf{\Lambda} = \begin{bmatrix} p_0 & p_1 & p_2 & 1 - p_0 - p_1 - p_2 \\ p_0 & 0 & p_2 & 1 - p_0 - p_2 \\ p_0 & p_1 & 0 & 1 - p_0 - p_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{30}$$

Using the technique from Formula (17), the probabilities p_i can be calculated numerically by considering what the event $\{Y = i\}$ looks like in terms of X; for example $\{Y = 0\} = \{X \in (\mu - 2\sigma, \mu + 2\sigma)\}$. In particular, $p_0 = \Phi(2) - \Phi(-2)$, $p_1 = \Phi(3) - \Phi(2)$, $p_2 = \Phi(-2) - \Phi(-3)$, where Φ is the cdf of a standard normal distribution. It can be noted using (21) that

$$\mathbf{I} - \mathbf{R} = \begin{bmatrix} 1 - p_0 & -p_1 & -p_2 \\ -p_0 & 1 & -p_2 \\ -p_0 & -p_1 & 1 \end{bmatrix}.$$

In this section, we will calculate the ARL through Proposition 4.5. Using software like Mathematica, it can be determined that $\det(\mathbf{I} - \mathbf{R}) \approx 0.986$, which verifies Theorem 4.3. Furthermore, we assume that our process starts in the safe area which implies that $\boldsymbol{\rho}_0 = \mathbf{e}_0 := (1, 0, \ldots)^T$. The inverse of $\mathbf{I} - \mathbf{R}$ can be calculated using the following definition for a general invertible matrix $\mathbf{M}$ (Formula 3.25 of Gentle (2007)):

$$\mathbf{M}^{-1} = \frac{\mathrm{adj}(\mathbf{M})}{\det(\mathbf{M})},$$

where $\mathrm{adj}(\mathbf{M})$ denotes the adjugate matrix of $\mathbf{M}$ which is equal to the transpose of the cofactor matrix $\mathrm{cof}(\mathbf{M})$. The cofactor matrix can be defined recursively (Formula 3.20 Gentle (2007)), for our matrix

specifically it is equal to

$$\begin{aligned}\operatorname{adj}(\mathbf{I}-\mathbf{R}) &= \operatorname{cof}(\mathbf{I}-\mathbf{R})^T \\ &= \begin{pmatrix} \det\begin{pmatrix}1 & -p_2\\ -p_1 & 1\end{pmatrix} & -\det\begin{pmatrix}-p_0 & -p_2\\ -p_0 & 1\end{pmatrix} & \det\begin{pmatrix}-p_0 & 1\\ -p_0 & -p_1\end{pmatrix} \\ -\det\begin{pmatrix}-p_1 & -p_2\\ -p_1 & 1\end{pmatrix} & \det\begin{pmatrix}1-p_0 & -p_2\\ -p_0 & 1\end{pmatrix} & -\det\begin{pmatrix}1-p_0 & -p_1\\ -p_0 & -p_1\end{pmatrix} \\ \det\begin{pmatrix}-p_1 & -p_2\\ 1 & -p_2\end{pmatrix} & -\det\begin{pmatrix}1-p_0 & -p_2\\ -p_0 & -p_2\end{pmatrix} & \det\begin{pmatrix}1-p_0 & -p_1\\ -p_0 & 1\end{pmatrix}\end{pmatrix}^T \\ &= \begin{pmatrix} 1-p_1p_2 & p_2p_0+p_0 & p_1p_0+p_0 \\ p_2p_1+p_1 & -p_2p_0-p_0+1 & p_1 \\ p_1p_2+p_2 & p_2 & -p_1p_0-p_0+1\end{pmatrix}^T.\end{aligned}$$

From this matrix it can readily be seen (sum over the first column of the cofactor matrix) that

$$\mathbf{e}_0^T\operatorname{adj}(\mathbf{I}-\mathbf{R})\mathbf{1} = 1+p_1+p_2+p_1p_2 = (1+p_1)(1+p_2).$$

From the cofactor matrix it can be derived that

$$\det(\mathbf{I}-\mathbf{R}) = (1-p_0)(1-p_1p_2)-p_1(p_0+p_0p_2)-p_2(p_0+p_0p_1) = 1-p_1p_2-p_0(1+p_1)(1+p_2),$$

We conclude that

$$\begin{aligned}\mathbf{e}_0^T(\mathbf{I}-\mathbf{R})^{-1}\mathbf{1} &= \frac{\mathbf{e}_0^T\operatorname{adj}(\mathbf{I}-\mathbf{R})\mathbf{1}}{\det(\mathbf{I}-\mathbf{R})} \\ &= \frac{(1+p_1)(1+p_2)}{1-p_1p_2-p_0(1+p_1)(1+p_2)}.\end{aligned}$$

Koutras et al. (2007) uses a method similar to the derivation of (29) to conclude that

$$\mathrm{ARL} = \frac{(1+p_1)(1+p_2)}{1-p_1p_2-p_0(1+p_1)(1+p_2)},$$

this can be seen as a verification that both methods of deriving the ARL indeed lead to the same quantity.

4.3 Conclusion

In order to calculate the ARL with runs rules, models based on Markov chains are used. The transition probability matrix of this model is called $\mathbf{\Lambda}$. Since $\mathbf{I}-\mathbf{\Lambda}$ is never invertible, the submatrix $\mathbf{R}$ is used for calculations. The eigenvalues of $\mathbf{R}$ all have an absolute value smaller than 1, which implies that $\mathbf{I}-\mathbf{R}$ is invertible. These facts are combined with the assumption that our process can never start out of control, this leads to easily computable expressions for the ARL. A general formula for the ARL can also be derived from the probability generating function of N, however this is computationally more intensive than calculating the inverse of $\mathbf{I}-\mathbf{R}$.

5 TBE control charts

In the previous section we looked at control charts for continuous data, i.e. data that can assume real values or a subinterval of the real line. However, in many industrial applications it is not possible to obtain data on a continuous scale or data is intrinsically discrete. An example of the latter is the case when one can only count objects. We will see that for industrial processes based on count data in which the counts are very rare, one has to resort to other types of control charts. In this chapter we will study the so-called TBE (Time Between Event) control chart and obtain generalizations of run length properties for this type of control charts. The mathematical complication that we have to deal with is that sums with a fixed upper bound become sums with a random upper bound which is highly dependent on the summands.

5.1 Introduction

In order to deal with count data, a straightforward idea is to develop a Shewhart chart based on a binomial distribution rather than a normal distribution, i.e. $X \sim \mathrm{Bin}(m, \pi)$ (see Section 3.3 of Qiu (2013)). When the sample size m is large, the Central Limit Theorem implies that X can be approximated by $\mathcal{N}(m\pi, m\pi(1-\pi))$. A problem with this approximation is that it requires π to be decently far away from 0 for m to be able to be considered large. In our application to high-yield processes π is too small for m to be considered large enough in a reasonable time frame for this approximation to yields satisfying results. Therefore an approximation based on the Central Limit Theorem cannot be used. Section 1.2 of Xie et al. (2002b) points out in more detail the issues with these charts.

In order to overcome these limitations, two main approaches have been suggested in literature. The traditional way of doing this is by transforming the data to an approximately normal distribution (see e.g., Nelson (1994)). Transformation of this data leads to an additional problem where the people operating the machines lose their feel for what the correct values should be. It has shown to yield better results to focus on the number of events between non-conforming events rather than the counts themselves. Therefore, it seems more appropriate to base control charts in high-purity process settings on conforming run length (CRL) (see e.g., Calvin (1983), Goh (1987), Woodall (1997), and Woodall and Driscoll (2015)). TBE control charts belong to the latter category.

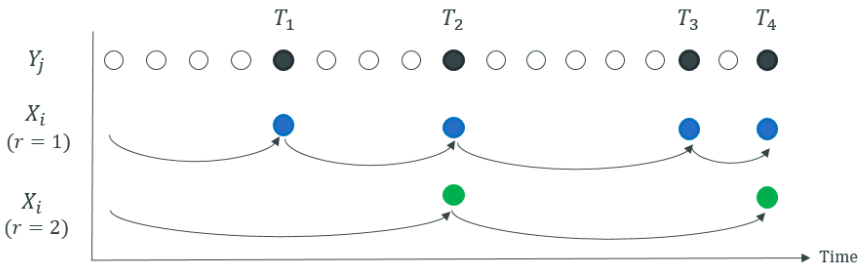

Figure 3: Number of points plotted for the Bernoulli process (Y_i) and the conforming run lengths $(X_i\ (r = 1), X_i\ (r = 2))$.

We will look at a special type of TBE chart, the CCC$_r$-chart, sometimes just called the CCC-chart. This chart relies on a Bernoulli process Y_i, which records conforming and nonconforming events. The control chart then monitors the number of conforming events between each nonconforming event X_i. In Figure 3 it can be seen that we can choose a value of r when deciding X_i. This $r \in \mathbb{N} \setminus \{0\}$ decides how many nonconforming events can happen before we restart our count, therefore X_i is equal to the number of conforming events until r nonconforming events have happened plus $r - 1$. Therefore

$$X_i = \min\{j > 0 \mid \#\{k \leq j \mid Y_k = 1\} = r \cdot i\}.$$

In particular, when the process is in control these X_i's are negative binomially distributed with parameters r, and p. When the process is out of control, the probability of the underlying Bernoulli distribution is changed, making the parameters of the negative binomial distribution r and $p^* := p + \delta$. The most used metric for control charts is the run length (RL), however Rizzo et al. (2020) illustrates some shortcoming of the RL in TBE control charts:

- It expresses the performances in control chart clock time rather than process clock time.
- It cannot be used to compare control charts.

For TBE control charts another metric is used, namely the average length of inspection (ALI). Define N to be the stopping time from definition (10). The LI (length of inspection) is then defined as follows:

$$\mathrm{LI} := \sum_{i \leq N} X_i. \tag{31}$$

In practice we look at two distributions based on the LI, namely $\mathrm{LI}_{\mathsf{in}} := \mathrm{LI}|\text{process is IC}$, and $\mathrm{LI}_{\mathsf{out}} := \mathrm{LI}|\text{process is OC}$. The difference between these distributions is the choice of parameters for X_i. The next section provides calculations for these distributions.

5.2 Average Length of Inspection

The ALI can be seen as the TBE chart analogue of the ARL. Using Definition (31) we find

$$\mathrm{ALI} := \mathbb{E}\left[\sum_{i \leq N} X_i\right]. \tag{32}$$

In order to calculate the ALI we first need to make use of Wald's Formula (see Lemma 10.2.9 of Grimmett and Stirzaker (2001), originally described in Wald (1945) for independent N and generalized in Blackwell (1946)) to the case of stopping times. Simple proofs of this formula use independence of N and X_i, this is however not required in general. To avoid the need for this independence a proof based on the so called optional stopping theorem will be used.

Lemma 5.1. *Let $X, X_1, \ldots, X_n$, be i.i.d. random variables with finite mean and variance. Let N be a stopping time with respect to the filtration $\mathcal{F}_n = \sigma(X_i \mid i \leq n)$ such that $\mathbb{E}[N] < \infty$. Then,*

$$\mathbb{E}\left[\sum_{i=1}^{N} X_i\right] = \mathbb{E}[N]\mathbb{E}[X]. \tag{33}$$

Proof. Because $\mathcal{F}_n$ is the sigma algebra generated by all X_i with $i \leq n$, for each n the random variable X_n is trivially $\mathcal{F}_n$ measurable. Furthermore as the X_i are i.i.d. X_n is independent from $\mathcal{F}_{n-1}$, this implies that $\mathbb{E}[X_{n+1} \mid \mathcal{F}_n] = \mathbb{E}[X_n \mid \mathcal{F}_n] = X_n$. Therefore X_n is a Martingale with respect to $\mathcal{F}_n$ according to Definition 7.7.3 of Grimmett and Stirzaker (2001), as X_n is assumed to be integrable. But then the process $M_n := \sum_{i=1}^{n} X_i - \mathbb{E}[X_i]$ is also a Martingale because

$$\begin{aligned}\mathbb{E}[M_{n+1} \mid \mathcal{F}_n] &= \mathbb{E}[X_{n+1} - \mathbb{E}[X_{n+1}] \mid \mathcal{F}_n] + \mathbb{E}[M_n \mid \mathcal{F}_n], \\ &= X_{n+1} - \mathbb{E}[X_{n+1}] + 0.\end{aligned}$$

Since each X_n has finite variance, it must hold that $\mathbb{E}[X_n^2]$ is a bounded sequence. Therefore we can conclude that there exists some constant C such that $\mathbb{E}[|X_n|] \leq C$ for each n. Note that this implies that X_n is bounded almost surely (if X_n is unbounded on some set of nonzero measure then $\mathbb{E}[|X_n|] = \infty$), which furthermore implies that surely $\mathbb{E}[X_N] < \infty$.
Finally it holds X_n is integrable (has finite mean), and $\{N > n\}$ approaches the empty set as $n \to \infty$ because $\mathbb{E}[N]$ is finite. Therefore X_n is finite almost everywhere which implies that $\mathbb{E}[X_n \mathbb{1}_{\{N>n\}}] \to 0$ as $n \to \infty$.

Therefore we can apply Theorem 12.5.1 of Grimmett and Stirzaker (2001) (the optional stopping theorem). Applying this theorem yields $\mathbb{E}[M_N] = \mathbb{E}[M_0]$, or equivalently:

$$\mathbb{E}\left[\sum_{i=1}^{N} X_i - \sum_{i=1}^{N} \mathbb{E}[X_i]\right] = 0.$$

Furthermore using the triangle inequality

$$\left|\sum_{i=1}^{N} \mathbb{E}[X_i]\right| \leq \sum_{i=1}^{N} \mathbb{E}[|X_i|] \leq CN.$$

Therefore we can add this series to both sides of the equation derived by the optional stopping theorem. Using linearity of the expectation operator we can then conclude that

$$\mathbb{E}\left[\sum_{i=1}^{N} X_i\right] = \mathbb{E}\left[\sum_{i=1}^{N} \mathbb{E}[X_i]\right] = \mathbb{E}[N\mathbb{E}[X]] = \mathbb{E}[N]\mathbb{E}[X].$$

□

Using Wald's formula we can then derive a relation between the ALI and the ARL. Using the definition of the ARL, it holds

$$\mathrm{ALI} = \mathbb{E}[X] \cdot \mathrm{ARL}.$$

Since the random variable X is defined to be the number of items until r nonconforming items are found it is negative binomially distributed with parameters r, p. The parameter p is the probability of an item being nonconforming i.e. the parameter of the Bernoulli distribution on which X is based. In particular it holds that $\mathrm{ALI}_{\mathsf{in}} = \frac{r}{p}\mathbb{E}[N]$, and $\mathrm{ALI}_{\mathsf{out}} = \frac{r}{p+\delta}\mathbb{E}[N]$.

Let us now consider the situation without runs rules and a two-sided control limit, i.e. N is defined equivalent to (11). We use the notational convention $\mathrm{UCL} = u$, $\mathrm{LCL} = \ell$. In the case where no runs rules are present, it holds true that N is geometrically distributed with parameter $p = 1 - F_{X_i}(\mathrm{u}) + F_{X_i}(\ell)$, equivalently $1 - p = \mathbb{P}(X \in [\ell, u])$. Theorem 2 of Di Bucchianico et al. (2005) then states the following.

Theorem 5.2. *Let $X, X_1, \ldots, X_n$, be i.i.d. random variables with finite mean μ and variance σ. Let N be a stopping time with respect to the filtration $\sigma(X_i \mid i \leq n)$ such that $\mathbb{E}[N] < \infty$. Then,*

$$\mathrm{Var}\left(\sum_{i=1}^{N} X_i\right) = \frac{\sigma^2}{p} - \mu^2\left(\frac{3-p}{p^2}\right) + 2\mu\mathbb{E}\left[N\sum_{i=1}^{N} X_i\right], \tag{34}$$

where p is the geometric distribution parameter of N.

This theorem may be difficult to work with as a result of the $\mathbb{E}\left[N\sum_{i=1}^{N} X_i\right]$ term. This can be worked around by imposing the extra condition that X is non-negative. This restriction is valid in the context of TBE control charts since the values of X represent real-life measurements (which cannot be negative). In Di Bucchianico et al. (2005) the following theorem has the assumption that X must have a discrete support, this however is not needed for the proof to be valid.

Theorem 5.3. *Let $X, X_1, \ldots, X_n$, be i.i.d. non-negative random variables with finite mean μ and variance σ. Let N be a stopping time with respect to the filtration $\sigma(X_i \mid i \leq n)$ such that $\mathbb{E}[N] < \infty$. Then,*

$$\mathrm{Var}\left(\sum_{i=1}^{N} X_i\right) = \frac{\sigma^2}{p} - \mu^2\left(\frac{1-p}{p^2}\right) + \frac{2\mu}{p^2}\mathbb{E}[X\mathbb{1}_{[\ell,u]}], \tag{35}$$

where p is the geometric distribution parameter of N.

Proof. Because X is non-negative, $\mathbb{E}[|X|] = \mathbb{E}[X] < \infty$. Definition 2.12 then implies that for each set A, $\mathbb{E}[X \mid X \in A] = \mathbb{E}[X\mathbb{1}_A]/\mathbb{P}(X \in A)$. We can therefore simplify $\mathbb{E}\left[N\sum_{i=1}^{N} X_i\right]$ by conditioning on values of N.

$$\begin{aligned}
\mathbb{E}\left[N\sum_{i=1}^{N} X_i\right] &= \sum_{n=1}^{\infty} \mathbb{E}\left[N\sum_{i=1}^{N} X_i \mid N = n\right]\mathbb{P}(N = n) \\
&= \sum_{n=1}^{\infty} n\sum_{i=1}^{n} \mathbb{E}\left[X_i \mid N = n\right]\mathbb{P}(N = n) \\
&= \sum_{n=1}^{\infty} n((n-1)\mathbb{E}[X \mid X \in [\ell, u]] + \mathbb{E}[X \mid X \in \mathbb{R} \setminus [\ell, u]])\mathbb{P}(N = n) \\
&= p\mathbb{E}[X \mid X \in [\ell, u]]\sum_{n=1}^{\infty} n(n-1)(1-p)^{n-1} + p\mathbb{E}[X \mid X \in \mathbb{R} \setminus [\ell, u]]\sum_{n=1}^{\infty} n(1-p)^{n-1}
\end{aligned}$$

The arithmetico-geometric series can be calculated by considering the derivatives of the power series $f(q) := \sum_{n=1}^{\infty} q^n = \frac{1}{1-q}$ for $0 \leq q < 1$. As $0 \leq 1-p < 1$ by definition, $f(1-p) = \frac{1}{p}$, and as power series are termwise differentiable within the radius of convergence (see Theorem 8.5.15 of Kosmala (2004)):

$$\sum_{n=1}^{\infty} n(1-p)^{n-1} = f'(1-p) = \frac{1}{p^2}$$
$$\sum_{n=1}^{\infty} n(n-1)(1-p)^{n-1} = (1-p)f''(1-p) = \frac{2(1-p)}{p^3}.$$

Therefore:

$$\begin{aligned}
\mathbb{E}\left[N\sum_{i=1}^{N} X_i\right] &= p\mathbb{E}[X \mid X \in [\ell,u]]\sum_{n=1}^{\infty} n(n-1)(1-p)^{n-1} + p\mathbb{E}[X \mid X \in \mathbb{R}\setminus[\ell,u]]\sum_{n=1}^{\infty} n(1-p)^{n-1} \\
&= \frac{2(1-p)}{p^2}\mathbb{E}[X \mid X \in [\ell,u]] + \frac{1}{p}\mathbb{E}[X \mid X \in \mathbb{R}\setminus[\ell,u]] \\
&= \frac{2(1-p)}{p^2}\frac{\mathbb{E}[X\mathbb{1}_{[\ell,u]}]}{\mathbb{P}(X \in [\ell,u])} + \frac{1}{p}\frac{\mathbb{E}[X(1-\mathbb{1}_{[\ell,u]})]}{1-\mathbb{P}(X \in [\ell,u])} \\
&= \frac{2(1-p)}{p^2}\frac{\mathbb{E}[X\mathbb{1}_{[\ell,u]}]}{1-p} + \frac{1}{p}\frac{\mathbb{E}[X] - \mathbb{E}[X\mathbb{1}_{[\ell,u]}])}{p} \\
&= \frac{2}{p^2}(\mathbb{E}[X\mathbb{1}_{[\ell,u]}]) + \frac{1}{p^2}(\mu - \mathbb{E}[X\mathbb{1}_{[\ell,u]}]) \\
&= \frac{\mu}{p^2} + \frac{1}{p^2}\mathbb{E}[X\mathbb{1}_{[\ell,u]}].
\end{aligned}$$

As a result, Theorem 5.2 implies that:

$$\begin{aligned}
\mathrm{Var}\left(\sum_{i=1}^{N} X_i\right) &= \frac{\sigma^2}{p} - \mu^2\left(\frac{3-p}{p^2}\right) + 2\mu\mathbb{E}\left[N\sum_{i=1}^{N} X_i\right] \\
&= \frac{\sigma^2}{p} - \mu^2\left(\frac{3-p}{p^2}\right) + 2\mu\left(\frac{\mu}{p^2} + \frac{1}{p^2}\mathbb{E}[X\mathbb{1}_{[\ell,u]}]\right) \\
&= \frac{\sigma^2}{p} - \frac{3\mu^2}{p^2} + \frac{p\mu^2}{p^2} + \frac{2\mu^2}{p^2} + \frac{2\mu}{p^2}\mathbb{E}[X\mathbb{1}_{[\ell,u]}] \\
&= \frac{\sigma^2}{p} - \mu^2\left(\frac{1-p}{p^2}\right) + \frac{2\mu}{p^2}\mathbb{E}[X\mathbb{1}_{[\ell,u]}].
\end{aligned}$$

□

This formula can be generalized to one sided intervals by letting $\ell = 0$, or $u \to \infty$.

The square root of the above variance is also called the Standard Deviation of the Length of Inspection (SDLI). The entire LI distribution can be determined through simulation.

5.3 Runs rules on CCC-charts

Similar to Shewhart charts, the runs rules are based on the values that X_i takes on. Since not every X_i corresponds to a single measurement, it is important to note that the stopping time N is now in a different time scale compared to Shewhart charts. The main difference between CCC-charts and Shewhart charts is that rather than X_i being normally distributed, X_i is negative binomially distribution. This change does not affect the setup of the Markov-Chains which are used to model this, but rather it changes the transition probabilities of these Markov-Chains. The Shewhart-type runs rules described in Section 3.3 are usually not used on CCC-charts, there are however ways to describe them.

Suppose we want to calculate the probability of the event $E_{a,b} := \{X_i \in (\mu + a\sigma, \mu + b\sigma)\}$, where $\mu = \mathbb{E}[X_i] = \frac{r}{p}$, and $\sigma^2 = \mathrm{Var}(X_i) = \frac{(1-p)r}{p^2}$. Then this event is equivalent to

$$E_{a,b} = \left\{X_i \in \frac{r}{p}\left(1 + a\sqrt{\frac{1-p}{r}}, 1 + b\sqrt{\frac{1-p}{r}}\right)\right\}.$$

Since the bounds of this interval are not guaranteed to be integers, ceiling and floor functions can be used as required. For simplicity define $\ell_a := \left\lceil \frac{r}{p}\left(1 + a\sqrt{\frac{1-p}{r}}\right)\right\rceil$, $u_b := \left\lfloor \frac{r}{p}\left(1 + b\sqrt{\frac{1-p}{r}}\right)\right\rfloor$. Then the probability of this event is equal to

$$\mathbb{P}(E_{a,b}) = \sum_{\ell_a \leq k \leq u_b} p_{X_i}(k), \tag{36}$$

where p_{X_i} is the probability mass function of X_i. Note that in calculations where the process is assumed to be OC, p is replaced by $p^* = p + \delta$. In this case the probability of the event where X_i is outside of the standard $(\mathrm{LCL}, \mathrm{UCL}) \supseteq [\ell_{-3}, u_3]$ interval is equal to $1 - \mathbb{P}(E_{-3,3})$. In the example provided in section 4.2 the following would hold in the CCC-chart analogue:

$$p_0 = \mathbb{P}(E_{-2,2}), \quad p_1 = \mathbb{P}(E_{2,3}), \quad p_2 = \mathbb{P}(E_{-3,-2}).$$

Since X_i is now a discrete random variable, singletons are no longer null-sets with respect to the image-measure of X_i. Therefore we have to be very careful about whether

$$E_{-3,-2} \cup E_{-2,2} \cup E_{2,3} = E_{-3,3}.$$

The following theorem will prove that these sets are indeed well defined.

Theorem 5.4. *Let $a, b, c \in \mathbb{Z}$ such that $a < b < c$. Then:*

$$E_{a,b} \cup E_{b,c} = E_{a,c}.$$

Proof. Let $x < y$, and define $U_{x,y} := [\ell_x, u_y]$. This implies that $E_{x,y} = \{X_i \in U_{x,y}\}$. Then note: $U_{a,b} \cup U_{b,c} = U_{a,c}$ implies

$$\begin{aligned} E_{a,b} \cup E_{b,c} &= \{X_i \in U_{a,b}\} \cup \{X_i \in U_{b,c}\} \\ &= \{X_i \in U_{a,b} \cup U_{b,c}\} \\ &= \{X_i \in U_{a,c}\} = E_{a,c}. \end{aligned}$$

Therefore it sufficed to show that $U_{a,b} \cup U_{b,c} = U_{a,c}$.

By definition, for all x it holds that u_k and ℓ_k are either the same integer or successive integers. Therefore $U_{a,b} \cup U_{b,c} = \{\ell_a, \ldots, u_b, \ell_b, \ldots, u_c\}$ is the set of successive integers between ℓ_a and u_c. However this set of successive integers between ℓ_a and u_c is precisely the definition of $U_{a,c}$. We conclude therefore that indeed $U_{a,b} \cup U_{b,c} = U_{a,c}$ and hence also $E_{a,b} \cup E_{b,c} = E_{a,c}$. □

The identity supplied above can then be proven by applying the theorem twice.

This theorem however, does however leave one additional problem. If $\frac{r}{p}\left(1 + k\sqrt{\frac{1-p}{r}}\right)$ is an integer for some k, Formula (36) will double count these probabilities. To avoid this we have to use the following convention. Let $n \in \mathbb{Z}$, then

$$\lfloor n \rfloor = n, \quad \lceil n \rceil = n + 1.$$

This is somewhat analogous to fixing the double counting problem by adding a very small ε to n. However the need to specify ε such that it is small enough is removed as it is needed to prevent cases in which $\lceil n + \varepsilon \rceil = n + 2$. Therefore the more efficient route is to 'define' $\lceil n \rceil = n + 1$. This convention also simplifies the proof of Theorem 5.4 somewhat since it implies that u_n and ℓ_n are guaranteed to be successive integer for all n. Note that most programming languages do not use this convention, therefore we might still need to use the ε trick in algorithms. This method might still however introduce an unwanted endpoint which needs to be removed manually.

In practice the analogues of Shewhart-type runs rules are not used on CCC-charts. The rules that are used are the following (as described in Section 7 of Rizzo et al. (2020)) inspired by Kenett and Pollak (2012):

- 1 observation below the LCL (ℓ). Note that the UCL is not used since observations that are too large are beneficial to the measured performance;
- 6 observations in a row on either side of the center line;
- 9 observations form a monotone sequence.

Since our observations X_i are bounded below by 0 but not bounded above, the distribution is skewed. As a result of the skew of this distribution, the center line is chosen to be the median rather than the mean. Since the assumptions about the distribution of the in-control process are known, this median is derived from our null hypothesis. In the case where this is not known, a rolling median can be used.

5.4 Code implementation details

The simulations for the CCC-charts are based on Monte-Carlo type simulations. We use the NumPy library in Python to generate a list $L := [L_0, \dots, L_{n-1}]$ of observations (i.e. $L_i \in \mathbb{R}$ for each $i \in \{0, \dots n-1\}$) from some specified probability distribution. In order to calculate which index of the list triggers an OC signal from one of the rules, the elements of need to be conditioned to some binary relation. Suppose $\sim$ is some binary relation on real numbers, then, for $a \in \mathbb{R}$, we define $L^* := (L \sim a)$ to be the list of boolean values $L^* = [L_0 \sim a, \dots, L_{n-1} \sim a]$. Note that in this case binary relations are treated like functions from real numbers to booleans, i.e. $\sim$: $\mathbb{R} \times \mathbb{R} \to \{\text{True}, \text{False}\}$, which is in line with how comparison operators are treated in programming languages. All the runs rules can be defined in terms of the following general algorithm. This algorithm is based on Shackleton (2020), and uses 'short-circuiting' which is a technical way of stating that the algorithm should end when the matching condition is found. This 'short-circuiting' avoids unnecessarily checking the whole array when it could have been ended earlier in the calculation.

Algorithm 1: The general array checking function, $\text{get\_first\_sequence}(L^*,\ n)$

input : A list L^* of boolean values, the number of consecutive elements $n \in \mathbb{N}$ that need to be true
output: The smallest $o \in \{0, \dots, \#L - 1\}$ such that $L_o, \dots L_{o+n}$ are all True, or $\#L$ if such an o does not exist

```
Let o = #L*
for i ∈ {0, ... #L* − n} do
    f = True
    for j ∈ {0, ..., n − 1} do
        if not L_{i+j} then
            f = False
            Break
        end
    end
    if f then
        o = i
        Break
    end
end
return o
```

This algorithm is optimized using the Numba package. Adding the @njit decorator to this function in the python implementation makes the runtime approximately 20 times faster on a list of 10^6 elements. For runs rule 2, Algorithm 1 can be readily executed with $n = 9$ on $L < m$ and $L > m$, where m is the median of our null-hypothesis distribution. For runs rule 3 a different approach is required. Let $D := [L_1 - L_0, \dots, L_{n-1} - L_{n-2}]$, this can be done efficiently using the *numpy.diff* function. Then n increasing points in L is equivalent to $n-1$ points in D being larger than 0, and smaller than 0 in the case of decreasing points. So rule 3 is equivalent to applying Algorithm 1 with $n = 5$ on $D > 0$ and $D < 0$. For the first runs rule, Algorithm 1 is not required. As only 1 point needs to be checked on $L < \ell \vee L > u$ (where $\vee$ represents the logical or operator, in NumPy one uses the *numpy.logical_or* function for this), it is faster to use the *numpy.argmax* function. The code is structured into classes using Object-Oriented programming. The code is structured in such a way that the sampling distribution and runs rules are passed into the arguments of the simulation function (this can be done using lambda bodies). This allows any user of the program to add and remove runs rules at will, and also makes it possible to change the sampling distribution.

For some distributions it may not be possible to analytically find a median. In order to still use such medians, we use a loop to find the median numerically. Let X be a discrete distribution with CDF $F_X : \mathbb{N}_0 \to [0, 1]$. Then if the preimage evaluated at $\frac{1}{2}$ is empty, i.e. $F_X^{-1}(\frac{1}{2}) = \varnothing$, the median is approximately equal to the following quantity:

$$\frac{1}{2}\left(\max\left\{n \;\middle|\; F_X(n) < \frac{1}{2}\right\} + \min\left\{n \;\middle|\; F_X(n) > \frac{1}{2}\right\}\right) = \min\left\{n \;\middle|\; F_X(n) > \frac{1}{2}\right\} - \frac{1}{2},$$

where the simplification can be made as $\max\left\{n \mid F_X(n) < \frac{1}{2}\right\}$ and $\min\left\{n \mid F_X(n) > \frac{1}{2}\right\}$ must be consecutive integers. If there exists an n such that $F_X(n) = \frac{1}{2}$, then that is the median. From these numerical versions of the runs rules,

In order to effectively compare performance of the control charts, the ALI or ARL values need to be set to the same value by adjusting the LCL. This can be done in two main ways. The first way assumes that the transition-probability matrix of the runs rule is known, and solve the equation derived from Proposition 4.5 for the control limit values. However this is not always possible, specifically in the case of runs rule 3 it is not possible to create a Markov model which corresponds to the runs rule. The other way is to apply a root-finding algorithm on the ARL derived from Algorithm 1. The downsides of this method are that an extremely large amount of iterations is required, and that the simulated values of the ARL are inherently random. The second problem can be combated by calculating multiple ARLs for each control limit and applying the root finding on the mean of those observations, this however makes the first problem worse by introducing even more iterations. This large amount of iterations shows why it is highly advantageous to utilise the optimizations from the Numba package (or write an optimized simulation in a low-level programming language).

Algorithm 2: One iteration of the simulation that finds the run length distribution.

input : A list L of observations (floating point numbers), the LCL ℓ, the median m
output: The first observation which yields an OC signal (numbered starting at 1), or $\#L+1$ if this never happens

Let $D = [L_1 - L_0, \ldots, L_{\#L-1} - L_{\#L-2}]$
$n_1 = \text{get\_first\_sequence}(L < \ell,\ 1)$
$n_2 = \min\{\text{get\_first\_sequence}(L < m,\ 9),\ \text{get\_first\_sequence}(L > m,\ 9)\}$
$n_3 = \min\{\text{get\_first\_sequence}(D < 0,\ 5),\ \text{get\_first\_sequence}(D > 0,\ 5)\}$
return $\min\{n_1 + 1,\ n_2 + 9,\ n_3 + 5\}$

Algorithm 2 is iterated approximately 10^3 times in order to find the run length distribution for a given median and LCL, the list of observations is generated by sampling from a probability distribution using an external function each iteration. For finding a control limit which has a certain ARL, this simulation is done for different LCL values chosen efficiently using a root finding algorithm. This root finding is done until we find a run length distribution for which the ARL is within some tolerance ε of our target ARL. Algorithm 2 can be modified to calculate the ALI rather than the ARL, this is done by instead returning

$$\sum_{n=0}^{n^*-1} L_n,$$

where $n^* = \min\{n_1 + 1,\ n_2 + 9,\ n_3 + 5\}$. This can be done using the *np.sum* function.

6 Results

In this section, control limit values and inspection length distributions will be calculated using the algorithms described in section 5.4. All calculations are done with a maximum run length of 10^4. This upper limit might skew the data slightly, however 10^4 is enough standard deviations away from the mean run length to where the effect is minimal.

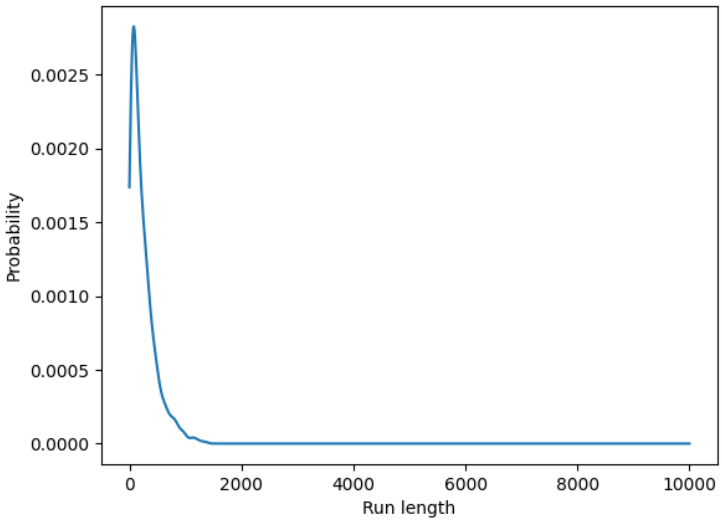

Figure 4: Run length distribution for $p = 0.05,\ r = 4$.

In Figure 4 it can be seen that indeed $\mathbb{P}(N \geq 10^4) \approx 0$, this verifies that the choice of maximum run length will have minimal effect on the RL and LI distributions.

6.1 LI distributions

In this section, inspection length distributions will be found through Monte-Carlo simulations of 10^4 iterations. From the graphs of these distributions, values for the ALI can be derived such that applying a root-finding algorithm on the LCL will yield reasonable results. Some ALI values may yield results such as $\mathrm{LCL} < 0$, which can never happen. To avoid this problem we perform some exploratory data analysis on the LI distributions for a small set of values.

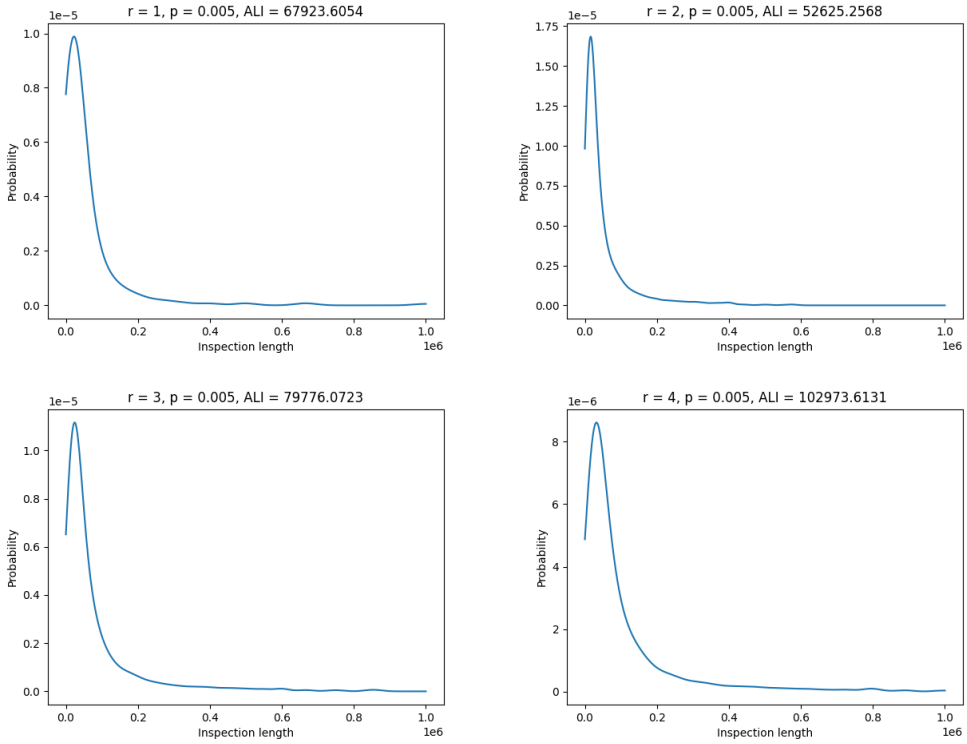

Figure 5: LI distributions for $r \in \{1, 2, 3, 4\}$ with $p = 0.05$, and $\mathrm{LCL} = \mu - 2\sigma$

All of the plots in Figure 5 contain similar distributions. The large peak closest to $LI = 0$, likely corresponds to the first runs rule (refered to as $L < \ell$ in section 5.4). For larger values of LI, small bumps can be found. These small bumps likely correspond to the other runs rules.

6.2 Control limits

In order to effectively compare control charts, their ALI must be the same. In order to change the ALI, we modify the value of k where $\mathrm{LCL} = \mu - k\sigma := \mathbb{E}[X] - k\sqrt{\mathrm{Var}(X)}$. Here $X \sim \mathrm{NB}(p, r)$ as is common for CCC-charts. The root finding algorithm is performed 100 times, the chosen value of k is the mean of these 100 values.

p	r	target ALI	$\bar{k}$
0.005	1	80000	0.78
0.005	2	80000	1.12
0.005	3	80000	1.31
0.005	4	80000	1.46
0.008	1	80000	0.43
0.008	2	80000	0.96
0.008	3	80000	1.18
0.008	4	80000	1.33

Table 1: The determined values for k for chosen ALI values.

It is important to note that these values for k are not exact, but are averages of sequences of values for k. If one were to run the simulation with these values of k, the resulting ALI values will not be exactly 80000. In order to get closer to an exact value of k, an enormous amount of computation time would be required.

In the case where the third runs rule is not present, an exact value for k can be found for each target ALI by creating a Markov-chain model corresponding to the runs rules. Formula (33) and Proposition 4.5 can then be used in a root-finding algorithm to find a value for k which is not probabilistic. The thirds runs rule prevents such a method however as it requires knowledge of the previous observation which cannot be simplified to a single matrix.

6.3 Numba optimizations

To deal with the large amount of iterations that need to be performed in order to get meaningful data from the simulations, the software doing the simulations must be very optimized. These optimizations are done with Numba. In order to measure the effect of the Numba optimizations, the timeit package is used on a test function. The test function makes a list of observations sampled from a standard normal distribution. The Numba optimized function is then instructed to find the first occurrence of 9 consecutive points being larger 10.5. The first occurence of 9 consecutive points smaller than -10.5 is then also found. The probabilitiues of these events are essentially 0, this guarantees that the test function will not terminate prematurely. If the test function does terminate prematurely, the Numba optimzation will not work as it cannot further optimize built-in NumPy functions.

Number of iterations	Time with Numba (ms)	Time without Numba (ms)
10^4	0.4	9
10^5	5	98
10^6	39	977
10^7	426	9646
10^8	4630	92364

Table 2: The timeit results from the test function with and without Numba for certain amounts of iterations.

From Table 2 it can be seen that Numba improves the runtime of the test function by approximately 2000%.

7 Conclusion

At the start of this project, two main goals were set to be completed.

1. Implementation of Markov Chains approach to correctly compute the control limits of TBE charts with runs rules. This is required in order to make fair comparisons between control charts.

2. Numerical simulations to compare the effects of different run rules on the distribution of the run length.

The Markov-chain approach for runs rules on control charts turned out to have some holes in its mathematical basis. The main problems with the literature on this topic were the following: Inverses of matrices were used which were not proven to be nonsingular, and two different approaches for obtaining an analytic expression for the ARL which were not proven to be equivalent. In this paper, these mathematical deficiencies have been solved by proving that: $\mathbf{I} - \mathbf{R}$ is nonsingular, $\mathbf{I} - z\mathbf{\Lambda}$ is nonsingular for relevant z, the formulae for ARL in terms of $\mathbf{\Lambda}$ are equivalent to those in terms of $\mathbf{R}$, and for runs rule C_2 the probability generating function approach yields the same formula for ARL as the matrix algebra approach.

Some numerical simulations to determine the effects of the runs rules on RL and LI distributions have been done. The main contribution to that area is that this paper supplies exceptionally fast code for these calculations by including optimizations from the Numba package in Python. This code is also designed to be modular, therefore runs rules can be added or removed as needed and distributions can also be easily swapped out. The numerical simulation is based on one efficient function which find the first sequence of n consecutive True values inside an array of booleans. Every runs rule can be expressed in terms of this efficient function, and therefore a large amount of time can be saved by using this program as opposed to an equivalent implementation in, for example, R or Python without Numba optimizations.

References

S. Ali, A. Pievatolo, and R. Göb. An overview of control charts for high-quality processes. *Qual. Reliab. Engng. Int.*, 32:2171–2189, 2016.

D. Blackwell. On an equation of Wald. *Ann. Math. Statistics*, 17(1):84–87, 1946.

V.I. Bogachev. *Measure Theory*. Springer, New York, 2007.

D. Brook and D.A. Evans. An approach to the probability distribution of CUSUM run length. *Biometrika*, 59(3):539–549, 1972.

T.W. Calvin. Quality control techniques for "zero defects". *IEEE Trans. Comp. Hybr. Man. Techn.*, 6 (3):323–328, 1983.

J.B. Conway. *A Course in Functional Analysis*. Graduate Texts in Mathematics. Springer, New York, 1994.

A. Di Bucchianico, G.D. Mooiweer, and E.J.G. Moonen. Monitoring infrequent failures of high-volume production processes. *Qual. Reliab. Engng. Int.*, 21:521–528, 2005.

M. Frisén. Optimal sequential surveillance for finance, public health, and other areas. *Sequential Analysis*, 28(3):310–337, 2009.

J.E. Gentle. *Matrix Algebra*. Springer, New York, 2007.

S. Gerschgorin. Über die Abgrenzung der Eigenwerte einer Matrix. *Izvestija Akademii Nauk SSSR, Serija Matematika*, 7(3):749–754, 1931.

T.N. Goh. A control chart for very high yield processes. *Qual. Assurance*, 13:18–22, 1987.

G.R. Grimmett and D.R. Stirzaker. *Probability and Random Processes*. Oxford University Press, Oxford, 2001.

R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1990.

R. Kenett and M. Pollak. On assessing the performance of sequential procedures for detecting a change. *Qual. Reliab. Eng. Int.*, 28(5):500–507, 2012.

W.A.J. Kosmala. *A Friendly Introduction to Analysis: Single and Multivariable*. Featured Titles for Real Analysis. Pearson Prentice Hall, Upper Saddle River, New Jersey, 2004.

M. V. Koutras, S. Bersimis, and P. E. Maravelakis. Statistical process control using Shewhart control charts with supplementary runs rules. *Methodology and Computing in Applied Probability*, 9(2): 207–224, 2007.

C.H. Little, T.L. Kee, and B. Bruce van Brunt. *Real Analysis via Sequences and Series*. Springer, New York, 2016.

J.M. Lucas and R.B. Crosier. Fast initial response for CUSUM quality-control schemes: give your CUSUM a head start. *Technometrics*, 42(1):102–107, 2000.

L.S. Nelson. A control chart for parts-per-million nonconforming items. *J. Qual. Technol.*, 26(3): 239–240, 1994.

P. Qiu. *Introduction to Statistical Process Control*. Chapman and Hall/CRC, Boca Raton, Florida, 2013.

C. Rizzo, S. Chin, E. R. Van den Heuvel, and A. Di Bucchianico. Performance measures of discrete and continuous time-between-events control charts. *Qual. Reliab. Eng. Int.*, 2020.

G.G. Roussas. *An Introduction to Measure-Theoretic Probability*. Elsevier, Burlington, MA, 2004.

H. Shackleton. Numpy array: First occurence of n consecutive values smaller than threshold. https://www.javaer101.com/en/article/1008280.html, 2020.

R.S. Varga. *Matrix Iterative Analysis*. Springer, Berlin, 1962.

A. Wald. Sequential tests of statistical hypotheses. *Ann. Math. Statistics*, 16(2):117–186, 1945.

Western Electric Company. Statistical quality control handbook, 1956.

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

W.H. Woodall and A. Driscoll. Some recent results on monitoring the rate of a rare event. In S. Knoth and W. Schmid, editors, *Frontiers in Statistical Quality Control 11*, pages 15–27. Springer, 2015.

M. Xie, T. Goh, and P. Ranjan. Some effective control chart procedures for reliability monitoring. *Reliab. Eng. Syst. Safety*, 77:143–150, 2002a.

M. Xie, T.N. Goh, and V. Kuralmani. *Statistical Models and Control Charts for High Quality Processes*. Kluwer, Boston, 2002b.

F. Zhang. *Matrix Theory: Basic Results and Techniques*. Universitext. Springer, New York, 2011.

A Source code

A version of the following source code with docstrings can be found on the following GitHub page: https://github.com/nCorneille/SPC_calculations

```python
import numpy as np

class SimulationHandler:

    def __init__(self, sampling_distribution, stopping_rules,
                 LCL: float = -3, UCL: float = 3, CL: float = 0):
        self.sampling_distribution = sampling_distribution
        self.LCL = LCL
        self.UCL = UCL
        self.CL = CL
        self.stopping_rules = stopping_rules

    def simulate_run_length(self, changepoint: int = -1,
                            changepoint_distribution = lambda x: x,
                            max_iterations: int = 1000, LCL=0):

        return self.simulate_run_length_variable_rules(self.stopping_rules, changepoint,
                            changepoint_distribution, max_iterations, LCL)

    def simulate_inspection_length(self, changepoint: int = -1,
                            changepoint_distribution = lambda x: x,
                            max_iterations: int = 1000, LCL=0):

        return self.simulate_inspection_length_variable_rules(self.stopping_rules,
                            changepoint,
                            changepoint_distribution, max_iterations, LCL)

    def simulate_run_length_variable_rules(self, stopping_rules, changepoint: int = -1,
                                           changepoint_distribution = lambda x: x,
                                           max_iterations: int = 1000, LCL=0):
        if changepoint == -1:
            data: np.array(float) = self.sampling_distribution(max_iterations)
        else:
            changed_data = changepoint_distribution(max_iterations - changepoint)
            sample_data = self.sampling_distribution(changepoint)
            data = np.concatenate((sample_data, changed_data))

        n = []
        for rule in stopping_rules:
            n.append(rule(data))

        return np.min(n) + 1

    def simulate_inspection_length_variable_rules(self, stopping_rules,
                                          changepoint: int = -1,
                                          changepoint_distribution=lambda x: x,
                                          max_iterations: int = 1000, LCL=0):
        if changepoint == -1:
            data: np.array(float) = self.sampling_distribution(max_iterations)
        else:
            changed_data = changepoint_distribution(max_iterations - changepoint)
            sample_data = self.sampling_distribution(changepoint)
            data = np.concatenate((sample_data, changed_data))

        n = []
        for rule in stopping_rules:
            n.append(rule(data))

        return np.sum(data[0::np.min(n) + 1])
```

```python
import numpy as np
from numba import njit

class RunsRules:
    @staticmethod
    def get_first_element(conditioned_data: np.array(bool)) -> int:
        if not np.any(conditioned_data):
            return len(conditioned_data)
        else:
            return np.argmax(conditioned_data)

    @staticmethod
    @njit
    def get_first_sequence(conditioned_data: np.array(bool), n: int) -> int:
        out = len(conditioned_data)
        for i in range(out - n + 1):
            found = True
            for j in range(n):
                if not conditioned_data[i + j]:
                    found = False
                    break
            if found:
                out = i
                break

        return out

    @staticmethod
    def upper_CL_rule(data, UCL):
        return RunsRules.get_first_element(data >= UCL)

    @staticmethod
    def lower_CL_rule(data, LCL):
        return RunsRules.get_first_element(data <= LCL)

    @staticmethod
    def double_sided_CI_rule(data, LCL, UCL):
        return RunsRules.get_first_element(np.logical_or(data <= LCL, data >= UCL))

    @staticmethod
    def n_points_above_CL(data: np.array(float), n: int, CL: float):
        out = RunsRules.get_first_sequence(data > CL, n)
        return out + (0 if out == len(data) else n)

    @staticmethod
    def n_points_below_CL(data: np.array(float), n: int, CL: float):
        out = RunsRules.get_first_sequence(data < CL, n)
        return out + (0 if out == len(data) else n)

    @staticmethod
    def n_points_increasing(data: np.array(float), n: int):
        diff = np.diff(data)
        out = RunsRules.get_first_sequence(diff > 0, n - 1)
        return out + (1 if out == len(diff) else n)

    @staticmethod
    def n_points_decreasing(data: np.array(float), n: int):
        diff = np.diff(data)
        out = RunsRules.get_first_sequence(diff < 0, n - 1)
        return out + (1 if out == len(diff) else n)
```

```
import os
import timeit

import matplotlib.pyplot as plt

from scipy import stats
from scipy import optimize

from src.simulation_handler import *
from src.runs_rules import *

def test_function():
    n = 1000000
    m = 1
    for _ in range(m):
        data: np.array(float) = np.random.normal(0, 1, n)

        n_u = RunsRules.n_points_above_CL(data, 9, 10.5)
        n_l = RunsRules.n_points_below_CL(data, 9, -10.5)

        return min(n_u, n_l)


def simulation_CCC_charts():
    r = 4
    p = 0.005
    p_tilde = 0.008
    changepoint = -1

    median = 0
    while stats.nbinom.cdf(median, r, p) - 1 / 2 < 0:
        median += 1

    sd = np.sqrt((1-p)*r)/p
    #print(median)
    #print(sd)

    def sampling_distr(x,p_): return np.random.negative_binomial(r, p_, x)

    LCL = max(0, median - 2 * sd)
    sim = SimulationHandler(lambda x: sampling_distr(x, p),
                            [lambda x: RunsRules.n_points_above_CL(x, 9, median),
                             lambda x: RunsRules.n_points_below_CL(x, 9, median),
                             lambda x: RunsRules.n_points_increasing(x, 6),
                             lambda x: RunsRules.n_points_decreasing(x, 6),
                             lambda x: RunsRules.lower_CL_rule(x, LCL)])


    def simulate(_LCL):
        out = []
        for i in range(1000):
            out.append(sim.simulate_run_length_variable_rules(
              [lambda x: RunsRules.n_points_above_CL(x, 9, median),
               lambda x: RunsRules.n_points_below_CL(x, 9, median),
               lambda x: RunsRules.n_points_increasing(x, 6),
               lambda x: RunsRules.n_points_decreasing(x, 6),
               lambda x: RunsRules.lower_CL_rule(x, _LCL)],
              changepoint, lambda x: sampling_distr(x, p_tilde)))

        return np.mean(np.array(out))

    print(simulate(LCL))
    output = optimize.root_scalar(lambda k: simulate(median - k * sd) - 230,
```

```
64                              bracket=[0, 3], x0=1.5, xtol=0.01)
65
66      print(output.root)
67      print(simulate(output.root))
68      #density = stats.gaussian_kde(out)
69      #x = np.linspace(0, 2000, 5000)
70
71      #plt.plot(x, density(x))
72      #plt.show()
73
74
75  if __name__ == "__main__":
76      #print(timeit.timeit(test_function, number=1))
77      simulation_CCC_charts()
```