Received: 23 November 2020 | Revised: 8 April 2021 | Accepted: 28 May 2021

DOI: 10.1002/qre.2945

SPECIAL ISSUE ARTICLE

WILEY

The Shiryaev–Roberts control chart for Markovian count time series

Sebastian Ottenstreuer

Helmut Schmidt University, Department of Mathematics and Statistics, Hamburg, Germany

Correspondence
Sebastian Ottenstreuer, Department of Mathematics and Statistics, Helmut Schmidt University, Hamburg, Germany.
Email: ottensts@hsu-hh.de

Abstract

The present article examines the performance of the Shiryaev–Roberts (SR) procedure for Markov-dependent count time series, using the Poisson INARCH(1) model as the representative data-generating count process. For the purpose of easier performance evaluation, a comparative analysis with existing cumulative sum (CUSUM) results from the literature is provided. In particular, the zero-state and the steady-state behavior of the two control schemes is considered with regard to the average run length (ARL), the median run length, and the extra quadratic loss as performance measures. The comparison shows that SR performs at least as well as its more popular competitor in detecting changes in the process distribution. In terms of usability, however, the SR procedure has a practical advantage, which is illustrated by an application to a real data set. Moreover, a parametric bootstrap study based on a second data example investigates the effects of parameter estimation on the chart's true ARL to false alarm. In sum, the research reveals the SR chart to be the better tool for monitoring Markov-dependent counts.

KEYWORDS

count time series, INARCH(1) model, run length performance, Shiryaev–Roberts, statistical process control

1 | INTRODUCTION

The application of control charts enables the user to statistically monitor processes, such as the (machine) temperature within a manufacturing process or the number of items defectively produced, in order to detect changes therein as fast as possible. Besides manufacturing industries, change point detection schemes are also used, among others, in service industries or health surveillance. Probably the most famous of its kind is the Shewhart control scheme, named after Walter A. Shewhart, who basically laid the foundation of the today's statistical process control (SPC) with his pioneering works in the 1920s and 1930s. Thanks to their ease of use, Shewhart charts still enjoy great popularity, even though it is known that they have significant weaknesses in detecting smaller changes. To enhance the sensitivity toward small shifts in the process distribution, control procedures with an inherent memory have been proposed over the following decades, such as the cumulative sum (CUSUM) or the exponentially weighted moving average (EWMA) charts by Page[1] and Roberts,[2] respectively. Another chart of this sort is the Shiryaev–Roberts (SR) control chart, on which this article focuses in conjunction with a specific count data model. First introduced by Shiryaev[3] in a Brownian motion context,

the method was independently studied by Roberts[4] for discrete time, resulting in the today's name. Like CUSUM, the SR procedure employs the likelihood of the monitored process, see Section 2 below. For a more detailed overview of SR, the reader is referred to Pollak;[5] an extensive application example of the method is presented by Kenett and Pollak,[6] who consider the surveillance of power failure times in a computer center using a nonhomogeneous Poisson process. In terms of popularity, SR and CUSUM clearly differ, though. The latter is almost as popular as Shewhart charts in the field of SPC, whereas the SR chart has received much less attention so far. The present article, however, shows that this lack of interest is not justified, at least not when monitoring count time series with a Markovian dependence structure.

In general, count-type data are highly relevant in our everyday lives; for instance, think of the number of spam mails one receives per day/week or—omnipresent these days—the daily number of persons newly infected with a certain virus in a given area/country. As with any data type, time series of counts may exhibit some level of autocorrelation. However, this is often neglected in applications of control charts and serial independence is implicitly assumed instead. If the data are indeed autocorrelated, this false assumption can have a severe effect on a chart's performance, see Alwan,[7] Harris and Ross,[8] and others. Change point detection schemes based on the likelihoods, such as CUSUM or SR, can take the data's autocorrelation structure into account, given an appropriate model has been identified to describe the (prechange) observations. In terms of count data, suitable models including serial dependence are, for instance, the first-order integer-valued autoregressive (INAR(1)) process or the binomial autoregressive (AR(1)) process, both introduced by McKenzie[9] along with other AR(1)-like models for count time series. Further approaches for modeling time series of counts in an SPC context are summarized in Weiß.[10] When additionally faced with overdispersion in the data, popular alternatives are INGARCH processes, short for integer-valued generalized autoregressive conditional heteroskedasticity processes. The original names in Rydberg and Shephard[11] and Heinen[12] differ from the term INGARCH used by Ferland et al.,[13] but the latter one seems to be more common in the literature.[14] A special case of the INGARCH family are INARCH(1) models, two of which are presented in Section 3 below. In several applications these models, especially the Poisson (Poi) INARCH(1) process, have proven to be an excellent choice for modeling autocorrelated and overdispersed counts; see Weiß[15] and references therein. In addition, the Markov property simplifies the implementation of the likelihood, which in turn increases the usability of likelihood-based control charts. For these reasons, the (Poisson) INARCH(1) process is an ideal representative to investigate the SR procedure for the class of count time series models with Markovian dependence.

Specifically for Poi-INARCH(1) processes, Weiß and Testik[16] already proposed various CUSUM charts. Since SR can be seen as the natural competitor of CUSUM due to its similar structure, the results of Weiß and Testik[16] can perfectly be used as benchmarks, facilitating a better classification of the SR chart's performance. Comparisons between the two methods have a long tradition in the SPC literature; see, among others, Moustakides et al.,[17] Pollak and Siegmund,[18] and Roberts.[4] More recently, Polunchenko and Raghavan[19] compared CUSUM and SR for the standard Gaussian AR(1) model. To the best of the author's knowledge, the SR control scheme has not yet been studied in a count time series framework. This is now done in Section 4, where the performance of the SR chart is investigated for an underlying Poi-INARCH(1) process in comparison to the results of Weiß and Testik.[16] By considering different performance measures, an extensive comparative analysis is provided in order to determine whether the SR procedure can be a promising alternative for monitoring autocorrelated counts. The practical features of SR are demonstrated in Section 5, where the process monitoring is illustrated using a time series of monthly claim counts of burn injured workers in the heavy manufacturing industry from Freeland.[20] In addition, Section 5 briefly discusses the effects of estimation uncertainty on the SR chart's design and performance by means of a second data example, consisting of the weekly numbers of German districts with new cases of hantavirus infections in 2011. The binomial (Bin) INARCH(1) model, as it was proposed by Weiß and Pollett[21] for this time series, is presented in Section 3. The conclusions of this research are summarized in Section 6.

2 | THE SR PROCEDURE: NOTATIONS AND CONCEPTS

Let $(X_t)_{\mathbb{N}}$ be a random sequence of observations, which needs to be monitored. Assume further that $X_1, X_2, \ldots$ have a certain joint distribution that may sustainably shift to another one at an unknown but fixed point of time $\tau \in \mathbb{N}$. We call a process *in-control* as long as $t < \tau$ and *out-of-control* as soon as $t \geq \tau$. Detecting such a possible change in distribution is the objective of control charts. One type of control chart is the SR method, where the statistic

$$R_t = \sum_{k=1}^{t} \frac{f_{\tilde{\tau}=k}(X_1, \ldots, X_t)}{f_{\tilde{\tau}=\infty}(X_1, \ldots, X_t)} \tag{1}$$

is plotted against time $t = 1, 2, \ldots$. Here, $f_{\tilde{\tau}=k}$ is the joint density of the observations with supposed change point $\tilde{\tau} = k$, while $\tilde{\tau} = \infty$ represents no change occurring; in other words, $f_{\tilde{\tau}=\infty}$ denotes the in-control density. If the range of X_t is countable (as it is the case from Section 3 on, where $(X_t)_{\mathbb{N}}$ denotes a count process), f is the joint probability mass function (pmf) instead. But to introduce the SR chart most generally, we shall stick with f as probability density function (pdf) for the moment.

In SPC, $f_{\tilde{\tau}=\infty}$ is usually assumed to be known, whereas the true out-of-control distribution as well as the true change point are not. Thus, $f_{\tilde{\tau}=k}$ commonly represents the distribution that is most likely to be expected after a change, or which is of particular interest. If the observations are independent and identically distributed (i. i. d.) before and after the change point, the statistic simplifies to

$$R_t = \sum_{k=1}^{t} \prod_{i=k}^{t} \frac{f_1(X_i)}{f_0(X_i)}, \tag{2}$$

where f_0 (f_1) is the prechange (presumed postchange) pdf. Equation (2) enables the recursion

$$R_t = \frac{f_1(X_t)}{f_0(X_t)} (R_{t-1} + 1), \quad R_0 = 0. \tag{3}$$

In the presence of a first-order (homogeneous) Markovian dependence structure within the observations, analogous representations can be obtained:

$$R_t = \sum_{k=2}^{t} \prod_{i=k}^{t} \frac{f_1(X_i \mid X_{i-1})}{f_0(X_i \mid X_{i-1})} \tag{4}$$

and

$$R_t = \frac{f_1(X_t \mid X_{t-1})}{f_0(X_t \mid X_{t-1})} (R_{t-1} + 1), \quad R_1 = 0, \tag{5}$$

for $t = 2, 3, \ldots$ and conditional densities f_0 (prechange) and f_1 (postchange). Because τ is not observable, the process is considered to be out of control only after R_t exceeds a predefined threshold h for the first time. The corresponding stopping time

$$L = \min\{t \mid R_t > h\} \tag{6}$$

is also known as the run length. Its expectation under the probability measure with no change, $E_{\tau=\infty}[L]$, is called average run length (ARL) to false alarm or (in-control) zero-state ARL. The designation "zero-state ARL" is more general in that it also includes the (out-of-control) expectation of the run length, $E_{\tau=1}[L]$, when the process is disturbed right from the beginning of surveillance. In Section 4, we shall apply the zero-state ARL as a performance measure for the analysis. In addition, the conditional steady-state ARL (a notion introduced by Crosier[22]) is used, which is defined as $\lim_{k\to\infty} E_{\tau=k}[L - \tau + 1 \mid L \geq \tau]$. Both ARL concepts can reasonably be justified: The zero-state ARL provides information about the average time until a chart signals an alarm when the underlying process is either in control (then it is a false alarm) or out of control right from the beginning (that is sort of a worst-case scenario). If someone is, however, interested in the mean delay of detection when the process is not out of control initially but changes after a long time, the conditional steady-state ARL should be looked at. The latter measure is perhaps more suited to evaluate the typical out-of-control behavior of a chart, whereas the zero-state ARL seems to be more plausible (and is also more common in the SPC community) as an in-control measure. The specific charts in Section 4 are therefore adjusted with regard to their ARL to false alarm, abbreviated as ARL_0. Besides the expectations of these two run length concepts, their medians $MED_{\tau=1}[L]$ and $\lim_{k\to\infty} MED_{\tau=k}[L - \tau + 1 \mid L \geq \tau]$, as well as a weighted average ARL (more on this later) are considered in Section 4.

It is known that the expectation $E_{\tau=\infty}[R_L - L]$ exists and, due to the martingale characteristic of the likelihood ratios in (1), $E_{\tau=\infty}[L] = E_{\tau=\infty}[R_L] > h$ holds, even when the observations are dependent.[6,23] This implies a simple as well as practical rule for the SR chart to apply: In order to guarantee a minimum level of ARL to false alarm, the control limit h only needs to be set to this ARL. Moreover, the difference $R_L - h$ might be expected to be rather small, so that h might serve not only as a lower bound on the acceptable rate of false alarms but—because of $E_{\tau=\infty}[L] = E_{\tau=\infty}[R_L] > h$—as a

TABLE 1 Selection of important count distributions

Distribution	PMF $P(X = x) =$	Support
Poi	$\exp(-\lambda) \cdot \frac{\lambda^x}{x}, \lambda > 0$	$x \in \mathbb{N}_0$
Bin	$\binom{n}{x} \cdot \pi^x \cdot (1-\pi)^{n-x}, n \in \mathbb{N},\ \pi \in (0;1)$	$x \in \{0, \dots, n\}$
Negative Bin	$\binom{r+x-1}{x} \cdot \pi^r \cdot (1-\pi)^x, r > 0,\ \pi \in (0;1)$	$x \in \mathbb{N}_0$
Zero-inflated Poi	$1_{x=0} \cdot \omega + (1-\omega) \cdot \exp-(\lambda) \cdot \frac{\lambda^x}{x}, \omega \in [0;1), \lambda > 0$	$x \in \mathbb{N}_0$

conservative estimator for some desired ARL, too. In consideration of the results in Section 4, where $E_{\tau=\infty}[L] \gg h$ for some charts, it is, however, not advisable to approximate a predetermined ARL to false alarm using the control limit solely. For a more accurate estimation, linear asymptotics of the form $E_{\tau=\infty}[L] = c \cdot h$ as $h \to \infty$ have been established in the literature, where the value of the constant $c > 1$ can be calculated under certain assumptions.[6,23,24] To the best of the author's knowledge, generally applicable asymptotics of this type have not been shown for autocorrelated (count) observations yet. A brief simulation study at the end of Section 4 will instead indicate that the linear relationship is also (approximately) valid for Poi-INARCH(1) SR charts. Concerning the applicability of control charts, this feature is a major advantage of the SR procedure over other popular schemes: Once the constant c is computed, which is at worst done via simulations with large h, various in-control ARLs can be determined much easier and faster, see the application example in Section 5.

3 | INARCH(1) MODELS

The previous section introduced the SR procedure and its implementation for Markov processes. Focusing on Markov count processes, there are several models proposed in the literature, see Weiß[14] for a detailed survey. As one of the most popular instances among them, we consider the INARCH(1) model in some more detail. Generally, INARCH(1) models, or rather the family of INGARCH models can be seen as integer-valued counterparts to the conventional ARMA models, despite their (controversially discussed) designation as integer-valued GARCH models.[14] Given a count process $(X_t)_{\mathbb{N}}$ with range $\mathbb{N}_0$, INGARCH models take a linear regression of the conditional mean $E[X_t \mid X_{t-1}, X_{t-2}, \dots]$ as a basis. The specific model name then depends on the conditional distribution of X_t. Most commonly, the Poisson distribution is used for this, but other count distributions have also been examined in the literature.[14] Table 1 lists a few relevant count distributions, two of which are discussed in the framework of the following two INARCH(1) models.

3.1 | The Poi-INARCH(1) model

As mentioned above, the Poi-INARCH(1) model shall serve as the representative case of count data models with an AR(1)-like autocorrelation in this research. The observations X_t, conditioned on all previous observations, are assumed to be Poisson distributed, that is,

$$X_t \mid X_{t-1}, X_{t-2}, \dots \sim \text{Poi}(\beta + \alpha X_{t-1}),$$

with $\beta > 0$ and $\alpha \in (0;1)$. In this way, $(X_t)_{\mathbb{N}}$ forms a stationary, ergodic Markov chain,[13,25] the marginal distribution of which cannot be expressed in a closed-form formula but in terms of its marginal cumulants; see Weiß[26] for a recursive scheme to determine the cumulants. In particular, the mean and variance of an INARCH(1) process are

$$\mu = \frac{\beta}{1-\alpha} \quad \text{and} \quad \sigma^2 = \frac{\mu}{1-\alpha^2}, \tag{7}$$

indicating that the marginal distribution is obviously overdispersed, that is, $\sigma^2 > \mu$. It follows from Ferland et al.[13] that all moments of the Poi-INARCH(1) process exist and that its autocorrelation function equals the one of a standard AR(1) process, that is, $\rho(k) := \text{Corr}[X_t, X_{t-k}] = \alpha^k$. Further statistical properties are given in Ferland et al.[13] Readers interested in an application to a real data set are referred to Weiß,[15] where, among other count data models, the Poi-INARCH(1) model is fitted to the monthly number of strikes.

The Poisson transition probabilities of the process are

$$P(X_t = x_t \mid X_{t-1} = x_{t-1}) = \exp(-\beta - \alpha \cdot x_{t-1}) \cdot \frac{(\beta + \alpha \cdot x_{t-1})^{x_t}}{x_t!}, \tag{8}$$

see the first row of Table 1 (the remaining formulas are applied analogously for other types of INARCH(1) process). They lead to the conditional likelihood function

$$\begin{aligned} L(\beta, \alpha) &= P(X_T = x_T, \ldots, X_2 = x_2 \mid X_1 = x_1) \\ &= \prod_{t=2}^{T} P(X_t = x_t \mid X_{t-1} = x_{t-1}) \\ &= \exp\left(-(T-1)\beta - \alpha \sum_{t=2}^{T} x_{t-1}\right) \cdot \frac{\prod_{t=2}^{T} (\beta + \alpha \cdot x_{t-1})^{x_t}}{\prod_{t=2}^{T} x_t!}. \end{aligned} \tag{9}$$

(Various approaches for estimating the two model parameters, such as maximum likelihood [ML], can be found in Weiß.[15]) According to (5) and (8), the recursive version of the statistic of the Poi-INARCH(1) SR chart is given as

$$R_t = \exp(\beta_0 - \beta_1 + (\alpha_0 - \alpha_1)x_{t-1}) \cdot \left(\frac{\beta_1 + \alpha_1 \cdot x_{t-1}}{\beta_0 + \alpha_0 \cdot x_{t-1}}\right)^{x_t} \cdot (R_{t-1} + 1), \tag{10}$$

where, in line with the notation of Section 2, the subscript "1" in the model parameters refers to the out-of-control scenario that is conjectured to happen. Of course, (β_1, α_1) can differ from the true postchange parameters. The in-control parameters (β_0, α_0) are, however, assumed to be known a priori below (for instance, suppose they can fairly precisely be estimated from a long historical time series). For performance analysis or comparisons, this is a common and reasonable assumption, whereas in practical applications the user is often confronted with estimation uncertainty when designing a control chart. The effects estimation errors can have on the true ARL to false alarm of the SR chart are indicated in the context of a data example in Section 5.2. A general review of the SPC literature on this subject is given by Jensen et al.[27]

3.2 | The Bin-INARCH(1) model

For modeling a time series of overdispersed but finite-valued counts, Weiß and Pollett[28] proposed the Bin-INARCH(1) model, in which the observations X_t are assumed to be conditionally binomial distributed such that

$$X_t \mid X_{t-1}, X_{t-2}, \ldots \sim \text{Bin}\left(n, b + a\frac{X_{t-1}}{n}\right),$$

with $n \in \mathbb{N}$ and $b, b + a \in (0; 1)$. Like the Poi-INARCH(1) model, the Bin-INARCH(1) process is a stationary, ergodic Markov chain.[14] Due to the finite range of the process, its marginal distribution can readily be computed by utilizing the Markov property. Model mean and variance are given by

$$\mu = \frac{nb}{1-a} \quad \text{and} \quad \sigma^2 = \frac{\mu(1-\mu/n)}{1-(1-1/n)a^2}. \tag{11}$$

The autocorrelation function is again $\rho(k) = a^k$, but unlike in the above model, it can also take negative values. With binomial transition probabilities, recall Table 1, the conditional likelihood function and the statistic of the SR chart for Bin-INARCH(1) processes equal

$$L(b, a) = \prod_{t=2}^{T} P(X_t = x_t \mid X_{t-1} = x_{t-1})$$

$$= \prod_{t=2}^{T} \binom{n}{x_t} \left(b + a\frac{x_{t-1}}{n}\right)^{x_t} \left(1 - b - a\frac{x_{t-1}}{n}\right)^{n-x_t} \tag{12}$$

and

$$R_t = \left(\frac{b_1 + a_1 x_{t-1}/n}{b_0 + a_0 x_{t-1}/n}\right)^{x_t} \left(\frac{1 - b_1 - a_1 x_{t-1}/n}{1 - b_0 - a_0 x_{t-1}/n}\right)^{n-x_t} (R_{t-1} + 1). \tag{13}$$

Weiß and Pollett[28] fitted the Bin-INARCH(1) model to a time series of weekly counts of German districts with new cases of hantavirus infections. We shall pick up their data example in Section 5.2, to illustrate the estimation uncertainty that directly affects the design parameters of the SR chart and, thus, its performance. Yet for the following performance evaluation, the Poi-INARCH(1) model is used as the more popular and, therefore, more representative model for Markovian count time series.

4 | PERFORMANCE EVALUATION

As previously mentioned, the performance of the SR procedure is compared to the one of the CUSUM chart for the purpose of an easier evaluation. The results of the Poi-INARCH(1) CUSUM charts displayed below are taken from Weiß and Testik,[16] who used CUSUM statistics that have been likewise derived from the conditional likelihood function. They concentrated on the following three scenarios of conjectured changes:

1) uniform shift in both parameters, that is, $(\beta_1, \alpha_1) = \delta_1 \cdot (\beta_0, \alpha_0)$;
2) shift in β solely, that is, $(\beta_1, \alpha_1) = (\delta_1 \cdot \beta_0, \alpha_0)$;
3) shift in α solely, that is, $(\beta_1, \alpha_1) = (\beta_0, \delta_1 \cdot \alpha_0)$;

for $\delta_1 = 1.1, 1.25, 1.5, 2.0$. Hence, 12 different Poi-INARCH(1) CUSUM charts were analyzed in total. To assess their effectiveness, Weiß and Testik[16] considered both the zero-state and the steady-state ARL for various kinds of parameter shifts. The true out-of-control parameters, $(\beta, \alpha) \neq (\beta_0, \alpha_0)$, were basically generated in the same way as the chart parameters (β_1, α_1) above, but with a wider range of δ-values, namely, $\delta = 1.025, 1.05, 1.1, 1.25, 1.5, 2, 2.5$. So, only substitute δ_1 with δ above to obtain the true out-of-control parameters. The tabular presentation on the following pages provides a more comprehensible overview of all possible chart and shift type combinations studied. As in-control parameters, Weiß and Testik[16] chose two different pairs $(\beta_0, \alpha_0) = (3.5, 0.3)$ and $(\beta_0, \alpha_0) = (2, 0.6)$, both yielding the same model mean $\mu_0 = 5$. To be able to compare the performances of SR and CUSUM, the parameter and scenario selection of Weiß and Testik[16] will be adopted in the sequel. For the same reason, the thresholds of the different Poi-INARCH(1) SR charts are chosen so that each ARL to false alarm, ARL_0, approximately meets its counterpart from Weiß and Testik,[16] that is within the interval of 366 to 370. The conditional steady-state run lengths are determined with the same thresholds, as it is common practice in SPC to adjust a control chart in terms of its zero-state ARL and not of its steady-state ARL. The results in the following tables are obtained via simulations (so are the values of Weiß and Testik[16]), which means they are only approximations of the true values, though quite accurate ones due to 10^6 replications. The steady-state run lengths, $\lim_{\tau\to\infty}(L - \tau + 1 \mid L \geq \tau)$, are approximated with $\tau = 200$; a detailed description about the computation of the results in this research is provided by Appendix B.

For a thorough performance analysis, different measures are considered for both the zero-state and the steady-state run length concepts, namely the ARL, the median run length (MDRL), and the so-called extra quadratic loss (EQL). The latter is a popular measure to assess the overall (out-of-control) effectiveness of a control chart, see, for example, Ahmad et al.[29] or Ahmad et al.[30] Mathematically, the EQL reads as follows:

$$\mathrm{EQL} = \frac{1}{\delta_{\max} - \delta_{\min}} \int_{\delta_{\min}}^{\delta_{\max}} \delta^2 \cdot \mathrm{ARL}(\delta)\, d\delta$$

with $\mathrm{ARL}(\delta)$ denoting either the zero-state or the steady-state ARL for a specific shift δ. So the EQL is basically a weighted average of ARLs over the whole shift domain $\delta_{\min} < \delta < \delta_{\max}$ using δ^2 as the weight[31]; its computation is made with numerical integration in this research. More details about the EQL can be found in Wu et al.[32]

TABLE 2 Zero-state ARLs of SR and CUSUM[16] for Poi-INARCH(1) processes with $(\beta_0, \alpha_0) = (3.5, 0.3)$. If $\text{ARL}^{\text{SR}} \leq \text{ARL}^{\text{CUSUM}}$ (italic values), ARL^{SR} in bold. Table cells in gray where $(\beta, \alpha) = (\beta_1, \alpha_1)$

(β, α)	(β_1, α_1)	$\delta_1 \cdot (\beta_0, \alpha_0)$				$(\delta_1 \cdot \beta_0, \alpha_0)$				$(\beta_0, \delta_1 \cdot \alpha_0)$			
	δ_1	**1.1**	**1.25**	**1.5**	**2**	**1.1**	**1.25**	**1.5**	**2**	**1.1**	**1.25**	**1.5**	**2**
	δ h	**313**	**250.5**	**175.5**	**95.5**	**333**	**292**	**238**	**166.5**	**343.5**	**310.5**	**261.5**	**185.5**
	ARL_0	366.3	366.7	366.7	369.9	367.5	367.9	367.6	367.4	367.8	367.4	367.0	367.5
		366.2	*366.9*	*366.9*	*369.4*	*367.7*	*367.5*	*367.6*	*367.4*	*367.7*	*367.4*	*367.5*	*367.7*
$\delta \cdot (\beta_0, \alpha_0)$	1.025	206.2	**216.7**	**240.7**	**273.4**	213.4	**207.6**	**222.7**	**248.1**	246.9	214.3	**215.0**	**236.8**
		205.0	*225.6*	*248.9*	*276.6*	*202.1*	*212.9*	*231.2*	*256.1*	*202.4*	*209.7*	*222.2*	*243.3*
	1.05	135.4	**137.8**	**163.8**	**203.7**	146.4	**131.6**	**144.0**	**172.1**	185.9	144.5	**138.8**	**158.6**
		127.3	*145.4*	*173.0*	*210.9*	*126.2*	*134.1*	*152.3*	*182.4*	*127.4*	*131.0*	*142.1*	*166.9*
	1.1	77.4	**69.0**	**83.5**	**118.1**	90.4	69.1	**71.2**	**90.2**	125.1	84.5	72.4	**80.3**
		63.8	*71.0*	*90.6*	*125.4*	*66.1*	*66.0*	*75.5*	*98.3*	*66.9*	*65.2*	*69.4*	*85.2*
	1.25	34.1	22.9	**22.3**	**31.7**	44.7	27.2	21.8	**23.3**	61.7	36.0	26.0	22.9
		22.5	*20.3*	*22.9*	*34.5*	*25.8*	*21.7*	*20.9*	*24.9*	*24.0*	*21.6*	*20.3*	*21.6*
	1.5	17.8	10.6	8.5	**8.9**	26.0	14.4	10.0	8.2	30.5	17.0	11.7	9.0
		10.4	*8.5*	*7.9*	*9.2*	*13.1*	*10.3*	*8.6*	*8.1*	*10.3*	*9.2*	*8.2*	*7.7*
	2	8.7	5.3	4.1	3.5	15.2	8.0	5.3	4.0	12.4	7.4	5.3	4.2
		5.1	*4.2*	*3.6*	*3.4*	*7.1*	*5.5*	*4.4*	*3.7*	*4.7*	*4.3*	*3.9*	*3.5*
	2.5	5.7	3.8	3.0	2.5	11.1	5.9	4.0	3.0	7.2	4.8	3.7	3.0
		3.6	*3.1*	*2.7*	*2.4*	*5.1*	*4.0*	*3.3*	*2.8*	*3.3*	*3.1*	*2.8*	*2.5*
$(\delta \cdot \beta_0, \alpha_0)$	1.025	**243.5**	**258.0**	**281.5**	**305.5**	246.3	**246.7**	**263.9**	**286.9**	278.2	254.0	**257.9**	**278.8**
		246.3	*266.7*	*288.8*	*306.1*	*240.7*	*253.8*	*271.8*	*292.7*	*245.1*	*253.0*	*265.4*	*284.8*
	1.05	175.3	**187.6**	**218.6**	**253.5**	181.7	**175.0**	**194.7**	**226.6**	223.5	187.7	**189.0**	**214.6**
		173.3	*198.7*	*227.7*	*257.2*	*168.1*	*181.5*	*205.4*	*235.0*	*174.1*	*181.5*	*196.8*	*223.4*
	1.1	108.1	**109.2**	**137.2**	**178.7**	118.3	**101.8**	**114.2**	**145.8**	161.9	120.0	**112.7**	**133.3**
		98.5	*116.9*	*147.9*	*184.4*	*96.6*	*103.4*	*122.9*	*155.6*	*102.1*	*104.5*	*115.7*	*141.9*
	1.25	50.3	39.0	**46.1**	**71.4**	61.5	41.2	**38.6**	**49.9**	90.5	56.2	44.0	**45.6**
		36.8	*38.0*	*50.4*	*76.2*	*39.3*	*36.1*	*39.9*	*55.3*	*40.5*	*38.0*	*38.4*	*47.6*
	1.5	28.3	18.0	**16.5**	**23.2**	37.6	22.0	16.7	**16.8**	52.5	29.9	21.0	17.6
		17.6	*15.3*	*16.6*	*25.2*	*20.6*	*16.8*	*15.4*	*17.8*	*19.0*	*17.0*	*15.5*	*16.0*
	2	16.2	9.4	7.2	**7.2**	24.0	13.0	8.8	6.9	28.2	15.4	10.4	7.8
		9.1	*7.3*	*6.5*	*7.3*	*11.8*	*9.0*	*7.4*	*6.7*	*9.0*	*7.9*	*7.0*	*6.4*
	2.5	11.8	6.8	5.0	4.4	19.0	10.1	6.6	4.8	19.2	10.4	7.0	5.2
		6.5	*5.1*	*4.4*	*4.3*	*8.9*	*6.8*	*5.4*	*4.5*	*6.0*	*5.3*	*4.7*	*4.2*
$(\beta_0, \delta \cdot \alpha_0)$	1.025	298.5	**302.3**	**311.7**	**330.1**	304.2	**300.8**	**305.9**	**315.2**	315.5	299.1	**299.1**	**309.3**
		296.8	*304.6*	*315.7*	*332.0*	*299.5*	*302.1*	*308.8*	*319.7*	*293.6*	*297.0*	*302.1*	*312.7*
	1.05	247.2	**250.6**	**265.4**	**293.7**	257.8	**250.0**	**257.0**	**272.6**	276.1	248.2	**247.2**	**261.5**
		244.1	*255.5*	*272.0*	*297.2*	*246.5*	*252.3*	*261.9*	*278.8*	*238.2*	*242.0*	*250.9*	*265.4*
	1.1	179.1	**178.9**	**196.7**	**234.2**	193.8	**178.8**	**185.2**	**205.0**	220.3	181.1	**175.3**	**190.5**
		171.7	*183.3*	*204.2*	*241.9*	*175.5*	*178.8*	*191.1*	*213.9*	*166.1*	*168.9*	*177.5*	*196.7*
	1.25	90.8	**79.8**	**89.5**	**122.7**	107.5	84.1	**82.9**	**96.0**	135.9	93.5	80.4	**85.2**
		76.2	*80.8*	*95.1*	*131.5*	*81.9*	*79.9*	*85.4*	*103.2*	*75.7*	*73.6*	*76.5*	*88.6*
	0.5	46.9	34.0	**34.2**	**47.5**	60.6	39.5	33.7	**36.3**	79.2	48.2	36.3	33.6
		33.5	*31.7*	*35.7*	*51.8*	*38.4*	*33.6*	*33.1*	*38.9*	*34.5*	*31.6*	*30.4*	*33.1*
	2	21.5	13.6	11.5	**12.8**	30.8	17.7	13.0	**11.5**	35.8	20.6	14.7	11.9
		13.6	*11.5*	*11.1*	*13.4*	*16.7*	*13.5*	*11.7*	*11.6*	*13.6*	*12.2*	*11.1*	*10.7*
	2.5	12.4	8.0	6.5	**6.3**	19.1	10.9	7.8	6.4	18.2	11.2	8.4	6.8
		8.0	*6.7*	*6.1*	*6.3*	*10.1*	*8.1*	*6.8*	*6.2*	*7.7*	*7.0*	*6.4*	*5.9*

TABLE 3 Zero-state EQLs of SR and CUSUM charts for Poi-INARCH(1) processes with $(\beta_0, \alpha_0) = (3.5, 0.3)$. If $\text{EQL}^{\text{SR}} \leq \text{EQL}^{\text{CUSUM}}$ (italic values), EQL^{SR} in bold

(β_1, α_1)	$\delta_1 \cdot (\beta_0, \alpha_0)$				$(\delta_1 \cdot \beta_0, \alpha_0)$				$(\beta_0, \delta_1 \cdot \alpha_0)$			
δ_1	**1.1**	**1.25**	**1.5**	**2**	**1.1**	**1.25**	**1.5**	**2**	**1.1**	**1.25**	**1.5**	**2**
(β, α) h	**313**	**250.5**	**175.5**	**95.5**	**333**	**292**	**238**	**166.5**	**343.5**	**310.5**	**261.5**	**185.5**
ARL_0	366.3	366.7	366.7	369.9	367.5	367.9	367.6	367.4	367.8	367.4	367.0	367.5
	366.2	*366.9*	*366.9*	*369.4*	*367.7*	*367.5*	*367.6*	*367.4*	*367.7*	*367.4*	*367.5*	*367.7*
$\delta \cdot (\beta_0, \alpha_0)$	47.2	33.6	31.9	**37.0**	70.1	42.3	33.6	**32.7**	72.6	45.4	35.1	32.0
	32.1	*30.1*	*31.6*	*38.4*	*38.7*	*33.3*	*31.3*	*33.3*	*31.7*	*29.8*	*28.9*	*30.2*
$(\delta \cdot \beta_0, \alpha_0)$	77.3	55.0	**54.2**	**67.3**	104.2	65.1	53.8	**55.5**	129.8	78.3	59.7	55.2
	52.2	*49.6*	*54.7*	*70.1*	*60.3*	*52.5*	*51.3*	*57.7*	*53.4*	*50.0*	*48.8*	*52.8*
$(\beta_0, \delta \cdot \alpha_0)$	112.6	87.7	**87.4**	**107.6**	145.3	100.3	87.9	**90.7**	170.5	111.7	90.6	86.3
	85.8	*82.9*	*89.3*	*113.5*	*96.6*	*87.5*	*85.9*	*94.6*	*85.3*	*80.9*	*79.7*	*84.8*

To ensure a better overview, the tables of the zero-state and the steady-state MDRLs, Tables A.1 and A.2, are displayed in Appendix A. The discussion below is furthermore confined to the case where $(\beta_0, \alpha_0) = (3.5, 0.3)$. Since the qualitative conclusion of a comparison for $(\beta_0, \alpha_0) = (2, 0.6)$ is nearly the same (only very few combinations of shift and chart types lead to a different outcome of the comparison, without affecting the overall picture though), the corresponding results are omitted here, but are available from the author upon request.

Let us start with Table 2, displaying the zero-state ARLs. The performance of each of the 12 charts is read from top to bottom, starting with the ARL to false alarm below the horizontal double line, where $\delta = 1$ (equivalent to $(\beta, \alpha) = (\beta_0, \alpha_0) = (3.5, 0.3)$). The corresponding design parameters, including the control limits, can be gathered from the first three rows. The subdivision through vertical dashed lines shall help to more easily distinguish between the three different kinds of Poi-INARCH(1) SR charts according to the types (1), (2), (3) of presumed changes described above. The listed postchange parameters $(\beta, \alpha) \neq (\beta_0, \alpha_0)$ are sorted and subdivided in the same way, that is, uniform shift in both parameters (first row block), shift in β solely (second row block) and shift in α solely (third row block).

When looking at the out-of-control ARLs in Table 2, an interesting pattern becomes evident: Given an arbitrary chart type (i.e., with a certain kind of presumed change) and an arbitrary change scenario (e. g., shift only in β), the table cells in which SR outperforms CUSUM (values in bold) visually form a sort of triangle within the (dashed) boundary lines. In other words, the SR charts are clearly inferior to their CUSUM competitors for $\delta_1 = 1.1$ (first column of each column block) but successively improve toward them for increasing δ_1. The performance of the SR charts with $\delta_1 = 2$ (last column of each column block) is, in total, even better than the one of the corresponding CUSUM charts.

Nevertheless, the CUSUM method overall proves to be superior in terms of the zero-state ARL. This impression is also confirmed by Table 3, where the zero-state EQLs of the considered charts are displayed for the different out-of-control scenarios. The dominance of CUSUM is, however, less surprising in the gray-shaded table cells of Table 2, that is when the true process parameters and the design parameters of the charts meet each other. Under the assumption of a known postchange distribution and serially independent observations, Moustakides[33] showed that the CUSUM method is optimal with regard to the detection measure of Lorden,[34] which corresponds to the zero-state ARL in the case of CUSUM and SR. Despite the serial dependence in the count data, it is therefore reasonable that the SR procedure is inferior to CUSUM whenever (β, α) equals (β_1, α_1). But if the true parameters deviate from the design parameters downward such that $(\beta, \alpha) \leq (\beta_1, \alpha_1)$ with at least one parameter being less-than, SR appears to be the better testing method in fact.

Comparing the two methods in terms of their steady-state performance provides a different picture. Now it is the SR procedure outperforming its competitor in most table cells, see Tables 4, 5, and A.2. Only for large shifts, the CUSUM charts perform better in Table 4. As in the zero-state case, we can again see a kind of echelon form of table cells in which SR underbids the CUSUM values (Table 4), but less pronounced than before. Especially the SR chart with $(\beta_1, \alpha_1) = (\beta_0, 1.1 \times \alpha_0) = (3.5, 0.33)$ violates this form, which might be due to a different accuracy of the steady-state approximation used for this particular chart, though; see Appendix B for clarification. One also realizes that the SR charts now dominate in the gray-shaded table cells of Table 4. To some extent, this can be motivated by a proposition of Pollak and Tartakovsky[35] for i. i. d. observations: Under the constraint that the ARL to false alarm satisfies a certain level, Pollak and Tartakovsky[35]

TABLE 4 Steady-state ARLs of SR and CUSUM charts[16] for Poi-INARCH(1) processes with $(\beta_0, \alpha_0) = (3.5, 0.3)$. If $ARL^{SR} \leq ARL^{CUSUM}$ (italic values), ARL^{SR} in bold. Table cells in gray where $(\beta, \alpha) = (\beta_1, \alpha_1)$

(β, α)	(β_1, α_1)	$\delta_1 \cdot (\beta_0, \alpha_0)$				$(\delta_1 \cdot \beta_0, \alpha_0)$				$(\beta_0, \delta_1 \cdot \alpha_0)$			
	δ_1	1.1	1.25	1.5	2	1.1	1.25	1.5	2	1.1	1.25	1.5	2
	δ h	313	250.5	175.5	95.5	333	292	238	166.5	343.5	310.5	261.5	185.5
	ARL_0	366.3	366.7	366.7	369.9	367.5	367.9	367.6	367.4	367.8	367.4	367.0	367.5
		366.2	*366.9*	*366.9*	*369.4*	*367.7*	*367.5*	*367.6*	*367.4*	*367.7*	*367.4*	*367.5*	*367.7*
$\delta \cdot (\beta_0, \alpha_0)$	1.025	**162.6**	**202.7**	**235.8**	**271.3**	**147.8**	**184.2**	**213.7**	**244.6**	**114.2**	**160.6**	**193.3**	**229.4**
		189.6	*218.4*	*245.6*	*275.8*	*179.6*	*202.6*	*225.8*	*254.7*	*178.5*	*196.2*	*215.2*	*241.1*
	1.05	**99.9**	**126.5**	**159.7**	**202.3**	**93.4**	**112.4**	**136.1**	**169.3**	**77.2**	**100.5**	**120.0**	**151.6**
		114.7	*139.9*	*170.0*	*209.6*	*108.3*	*124.7*	*147.4*	*180.6*	*109.7*	*119.7*	*135.9*	*163.3*
	1.1	**51.4**	**60.4**	**80.3**	**117.5**	**51.0**	**54.8**	**65.4**	**88.0**	**44.8**	**52.2**	**58.3**	**75.0**
		55.7	*67.2*	*88.4*	*125.5*	*54.6*	*59.6*	*72.3*	*97.2*	*55.4*	*58.2*	*65.6*	*83.3*
	1.25	19.3	**18.1**	**20.4**	**31.1**	21.2	18.9	**18.4**	**22.1**	**18.3**	18.9	**18.4**	**19.8**
		18.9	*18.6*	*22.2*	*34.2*	*20.3*	*18.7*	*19.2*	*24.2*	*19.6*	*18.8*	*18.7*	*21.1*
	1.5	9.4	8.0	**7.5**	**8.7**	11.1	9.2	8.0	**7.6**	**8.7**	8.7	8.1	**7.6**
		8.9	*7.9*	*7.7*	*9.2*	*10.3*	*8.7*	*7.9*	*7.9*	*8.9*	*8.4*	*7.9*	*7.7*
	2	5.1	4.3	3.9	**3.6**	6.2	5.1	4.3	3.8	**4.6**	4.6	4.3	4.0
		4.8	*4.2*	*3.8*	*3.6*	*5.7*	*4.8*	*4.2*	*3.7*	*4.7*	*4.4*	*4.2*	*3.9*
	2.5	3.8	3.3	3.0	**2.7**	4.6	3.9	3.3	2.9	**3.6**	3.6	3.4	**3.1**
		3.6	*3.2*	*2.9*	*2.7*	*4.2*	*3.6*	*3.2*	*2.8*	*3.6*	*3.5*	*3.3*	*3.1*
$(\delta \cdot \beta_0, \alpha_0)$	1.025	**196.6**	**243.3**	**276.0**	**304.4**	**175.8**	**221.6**	**253.9**	**282.8**	**134.6**	**195.1**	**233.9**	**270.9**
		229.7	*260.4*	*285.1*	*305.7*	*215.4*	*242.0*	*266.5*	*291.0*	*218.0*	*237.2*	*258.1*	*282.2*
	1.05	**134.7**	**174.8**	**213.7**	**252.8**	**121.6**	**153.3**	**185.7**	**223.3**	**99.4**	**137.2**	**167.4**	**207.0**
		159.4	*190.9*	*224.6*	*256.1*	*146.6*	*171.6*	*199.6*	*232.5*	*151.8*	*169.0*	*189.9*	*220.7*
	1.1	**76.2**	**98.7**	**133.6**	**177.6**	**71.4**	**84.5**	**107.1**	**143.1**	**63.4**	**79.8**	**95.5**	**126.7**
		87.3	*110.9*	*144.9*	*182.8*	*81.3*	*94.9*	*118.0*	*155.0*	*86.4*	*94.5*	*110.3*	*138.6*
	1.25	**30.3**	**32.2**	**43.4**	**70.4**	**31.1**	**30.1**	**33.9**	**48.1**	**28.8**	**31.5**	**32.9**	**41.1**
		31.0	*35.2*	*49.1*	*76.0*	*31.2*	*31.5*	*37.3*	*54.1*	*32.9*	*32.9*	*35.6*	*46.3*
	1.5	15.2	**13.6**	**14.7**	**22.5**	16.8	14.5	**13.5**	**15.6**	**14.6**	14.9	**14.2**	**14.7**
		14.6	*13.7*	*15.8*	*24.8*	*15.8*	*14.1*	*13.9*	*17.0*	*15.4*	*14.7*	*14.3*	*15.5*
	2	8.1	6.7	**6.1**	**6.7**	9.7	7.9	6.6	**6.0**	**7.5**	7.4	6.8	**6.3**
		7.6	*6.5*	*6.1*	*7.0*	*8.8*	*7.4*	*6.4*	*6.1*	*7.6*	*7.1*	*6.6*	*6.3*
	2.5	5.8	4.8	4.2	**3.9**	7.2	5.8	4.7	**4.0**	**5.3**	5.2	4.8	4.4
		5.4	*4.6*	*4.1*	*4.0*	*6.5*	*5.3*	*4.5*	*4.0*	*5.4*	*5.0*	*4.7*	*4.3*
$(\beta_0, \delta \cdot \alpha_0)$	1.025	**246.7**	**286.1**	**305.9**	**327.9**	**226.4**	**274.5**	**294.7**	**311.6**	**160.8**	**236.6**	**273.5**	**300.9**
		278.1	*297.4*	*313.1*	*331.2*	*271.8*	*289.0*	*303.2*	*319.1*	*264.3*	*281.0*	*294.8*	*309.3*
	1.05	**200.1**	**236.5**	**260.5**	**292.7**	**185.4**	**224.5**	**246.5**	**268.7**	**133.9**	**190.9**	**223.6**	**253.2**
		227.3	*248.7*	*269.0*	*297.5*	*221.5*	*240.3*	*256.9*	*276.6*	*211.7*	*228.4*	*243.9*	*263.7*
	1.1	**138.8**	**166.6**	**191.9**	**233.2**	**132.0**	**157.0**	**176.3**	**202.3**	**98.2**	**132.0**	**155.2**	**183.6**
		157.6	*177.8*	*201.5*	*241.5*	*154.6*	*170.0*	*186.6*	*211.5*	*144.6*	*157.1*	*171.9*	*194.1*
	1.25	**62.8**	**71.0**	**86.3**	**121.6**	**63.9**	**68.8**	**76.5**	**93.8**	**51.0**	**59.9**	**66.1**	**80.1**
		67.2	*76.2*	*93.2*	*131.0*	*68.9*	*72.8*	*81.8*	*102.0*	*63.7*	*66.4*	*72.7*	*86.9*
	1.5	**28.9**	**28.4**	**32.2**	**46.9**	31.5	**29.5**	**29.8**	**34.9**	**25.7**	**27.4**	**27.4**	**30.2**
		28.9	*29.4*	*34.6*	*51.4*	*31.2*	*29.8*	*31.1*	*38.0*	*28.5*	*28.0*	*28.4*	*32.2*
	2	12.5	11.0	**10.8**	**12.9**	14.5	12.4	**11.1**	**11.1**	**11.5**	11.5	10.9	**10.6**
		12.0	*10.9*	*11.1*	*13.7*	*13.6*	*11.9*	*11.1*	*11.5*	*11.9*	*11.3*	*10.8*	*10.7*
	2.5	7.8	6.9	**6.4**	**6.6**	9.2	7.9	6.9	**6.4**	**7.2**	7.2	6.8	**6.4**
		7.5	*6.7*	*6.4*	*6.9*	*8.6*	*7.5*	*6.8*	*6.5*	*7.3*	*7.0*	*6.6*	*6.4*

TABLE 5 Steady-state EQLs of SR and CUSUM charts for Poi-INARCH(1) processes with $(\beta_0, \alpha_0) = (3.5, 0.3)$. If $\text{EQL}^{\text{SR}} \leq \text{EQL}^{\text{CUSUM}}$ (italic values), EQL^{SR} in bold

(β_1, α_1)	$\delta_1 \cdot (\beta_0, \alpha_0)$				$(\delta_1 \cdot \beta_0, \alpha_0)$				$(\beta_0, \delta_1 \cdot \alpha_0)$			
δ_1	**1.1**	**1.25**	**1.5**	**2**	**1.1**	**1.25**	**1.5**	**2**	**1.1**	**1.25**	**1.5**	**2**
(β, α) h	**313**	**250.5**	**175.5**	**95.5**	**333**	**292**	**238**	**166.5**	**343.5**	**310.5**	**261.5**	**185.5**
ARL_0	366.3	366.7	366.7	369.9	367.5	367.9	367.6	367.4	367.8	367.4	367.0	367.5
	366.2	*366.9*	*366.9*	*369.4*	*367.7*	*367.5*	*367.6*	*367.4*	*367.7*	*367.4*	*367.5*	*367.7*
$\delta \cdot (\beta_0, \alpha_0)$	**28.9**	**28.0**	**30.2**	**37.0**	32.2	29.6	**28.9**	**31.4**	**25.8**	**27.6**	**27.8**	**29.7**
	28.9	*28.9*	*31.5*	*38.9*	*31.5*	*29.4*	*29.7*	*32.9*	*28.6*	*28.3*	*28.8*	*31.0*
$(\delta \cdot \beta_0, \alpha_0)$	**44.0**	**43.9**	**49.7**	**65.4**	47.8	**44.7**	**45.1**	**52.0**	**39.4**	**42.9**	**44.0**	**48.9**
	44.6	*45.7*	*52.8*	*68.9*	*47.2*	*45.1*	*46.9*	*55.5*	*45.1*	*44.9*	*46.3*	*51.8*
$(\beta_0, \delta \cdot \alpha_0)$	**74.2**	**76.4**	**83.9**	**107.3**	**79.0**	**78.2**	**79.6**	**88.6**	**62.0**	**69.9**	**73.1**	**80.2**
	76.8	*79.0*	*88.2*	*114.1*	*81.3*	*79.8*	*82.6*	*93.8*	*73.8*	*74.6*	*76.8*	*84.2*

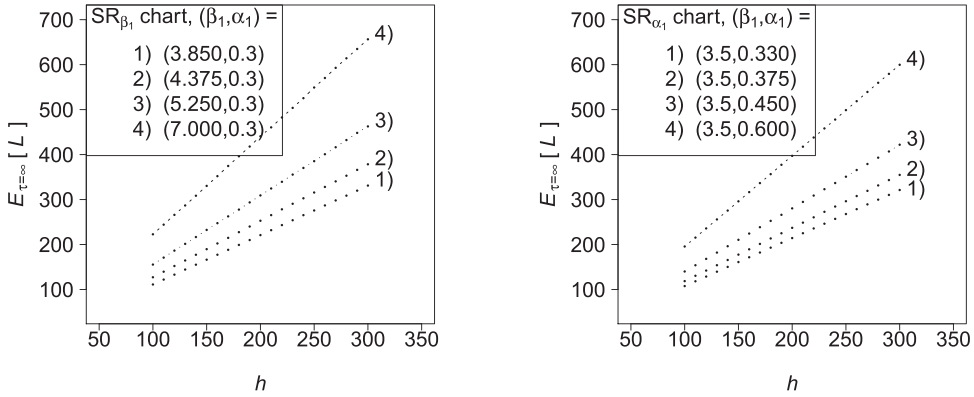

FIGURE 1 ARL to false alarm of Poi-INARCH(1) SR charts against increasing control limit h

found the SR procedure to be optimal with respect to a related steady-state concept, where the chart is repeatedly applied until a true alarm is raised. The measure used by Pollak and Tartakovsky[35] and the conditional steady-state ARL are very similar, so the outcome in the gray-shaded table cells seems plausible again. Anyway, the results in Tables 4, 5, and A.2 show quite clearly that the SR procedure beats the CUSUM method in a comparison of their steady-state performances for Poi-INARCH(1) processes.

All of the SR charts investigated in this study exhibit a linear relationship between the ARL to false alarm and the control limit such that $E_{\tau=\infty}[L] \approx c \cdot h$ for sufficiently large h (see Section 2), where the constant c differs for each chart. This is illustrated for a few of the above SR charts in Figure 1. In practice, the linearity is a great feature, because it facilitates the chart's adjustment considerably as the following section demonstrates.

5 | APPLICATION TO REAL DATA EXAMPLES

The following two data examples address different issues of practical application concerning the SR chart. By means of the first one, the process monitoring is demonstrated, pointing out the practical advantage of the SR chart. The second data example shall be used to illustrate the effects of estimation uncertainty that can occur in applications.

5.1 | Claim counts of injured workers

The 96 observations of the Freeland[20] data set describe the monthly claim counts of workers in the heavy manufacturing industry who collected short-term wage loss benefits (STWLB) from January 1987 to December 1994 due to a burn-related injury; further details on this times series are in Freeland.[20] Figure 2 shows the plotted data. Its empirical mean and

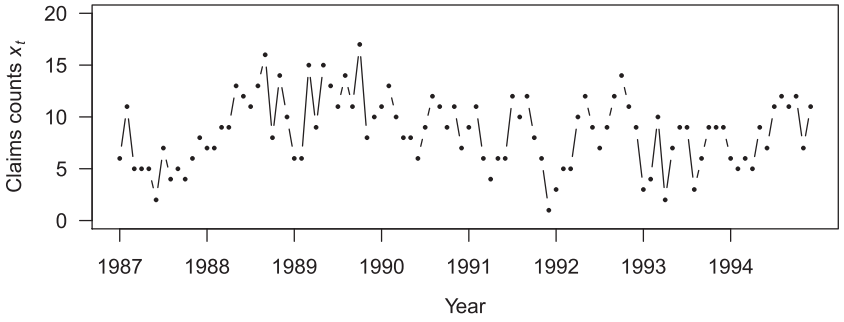

FIGURE 2 Time series of monthly claim counts of disabled workers from Freeland[20]

variance are 8.60 and 11.36, revealing a clear degree of overdispersion in the true marginal distribution. Weiß[26] identified a Poi-INARCH(1) process to have an adequate goodness of fit, even a better one than the original count data model used by Freeland.[20] The two model parameters β and α can be estimated by a direct numerical maximization of the conditional log-likelihood function $\ell(\beta, \alpha) := \ln(L(\beta, \alpha))$, with $L(\beta, \alpha)$ given by (9). Starting values required for the maximization can be obtained through formula (7) and the method of moments, which is however not recommended for estimating β and α in general.[16] The ML estimates found by Weiß[26] are $\hat{\beta} \approx 4.38$ and $\hat{\alpha} \approx 0.49$. With regard to the model definition in Section 3.1, an α-value of 0.49 means that nearly 50% of the disabled workers collecting STWLB are expected to remain in this state in the next month, while $\beta = 4.38$ implies that there are 4.38 new claims on average every month. According to (7), the model mean and the variance are given by 8.61 and 11.34, which is pretty close to their empirical counterparts.

Once the in-control model has been determined, we can start designing an SR chart by employing the estimated parameter values as β_0 and α_0 in formula (10). Next, a certain postchange scenario for the control chart must be considered in order to assign specific values to β_1 and α_1. Since we are dealing with claimants disabled due to burn injuries, it appears reasonable to assume that a practitioner is interested in detecting an increase in the claims counts, whether because of health or economic interests. Hence we want the control chart to trigger an alarm as soon as $\mu > \mu_0 = 8.61$. Let us further assume that the increase can be expected rather due to a jump in the number of newly incoming claims than due to a larger number of continuing claims (e. g., think of new machines in the heavy manufacturing industry, which cause more injuries in the beginning but not necessarily to more serious ones). For instance, choose $\beta_1 = 1.5\beta_0 = 6.57$ and $\alpha_1 = \alpha_0 = 0.49$, which translates into the same relative change of both the model mean and the variance, that is, $\mu_1 = 1.5 \times 8.61 = 12.915$ and $\sigma_1^2 = 1.5 \times 11.34 = 17.01$, see formula (7). This implies that the user wants the chart to trigger an alarm not until the mean number of claims has significantly increased. In a final step of designing the chart, an appropriate control limit h must be found. As in Section 4, we adjust h in terms of the ARL to false alarm, $E_{\tau=\infty}[L]$. At this point, one benefits from the linearity of the threshold in $E_{\tau=\infty}[L]$, that is, the fact that $E_{\tau=\infty}[L] = c \cdot h$ asymptotically holds even for Poi-INARCH(1) SR charts (see the discussion in Section 4 regarding Figure 1). We only need to calculate the constant c (via simulations), after which we can easily switch to a desired rate of false alarms without further computational effort. Since $E_{\tau=\infty}[L] > h$ (see Section 2), take $h = 370$ as the initial choice, for instance, guaranteeing an ARL to false alarm greater than 370 (a popular choice in SPC). A simulation with 10^6 replications yields $E_{\tau=\infty}[L] \approx 547.6$ so that $c = E_{\tau=\infty}[L]/h \approx 1.48$. If 547.6 now appears to be too large for practice, one can adjust the control limit to obtain the favored value. So, for an ARL to false alarm of approximately 370, use $h = 370/1.48 = 250$ instead; and, indeed, a simulation with $h = 250$ leads to $E_{\tau=\infty}[L] \approx 370.3$.

After the design parameters have been set based on the in-control data from Freeland,[20] we can now apply the control chart to a simulated path for illustrative purposes. The path is initialized by the last observation of the Freeland[20] data, that is, $x_0 = 11$. The first 40 observations, $x_1, \ldots, x_{40}$, then stem from the above in-control model, while the following 60 observations, $x_{41}, \ldots, x_{100}$, are generated with shifted parameters. As it is more realistic, let us consider out-of-control parameters for the underlying Poi-INARCH(1) process that deviate from the chart parameters β_1 and α_1: for example, $(\beta, \alpha) = (5, 0.55)$. Starting the surveillance at $t = 1$, the conditional mean delay of detection is $E_{\tau=41}[L - 40 \mid L \geq 41] = 28.6$ in this case. The considered sample path and the corresponding SR chart are displayed in Figure 3. Here, the chart first triggers an alarm at time $t = 95$, so with a delay of 55 observations after the change in the model parameters occurred. We can see, however, that the change has an immediate impact on the run of the chart and nearly causes an alarm at $t = 66$ already (which would have meant a delay of 26, i. e., closer to its mean). In practice, one is always confronted with the trade-off between a large ARL to false alarm on the one hand and a high sensitivity toward changes on the other hand. In the example of Figure 3, a practitioner may consider to lower the threshold in future in order to detect already situations like in $t = 66$, however at the cost of a smaller ARL to false alarm. Again due to the outlined linearity, one can conclude that $c \times r_{66} \approx 1.48 \times 211.48 \approx 313.0$ is about the largest possible ARL to false alarm that would still lead to an

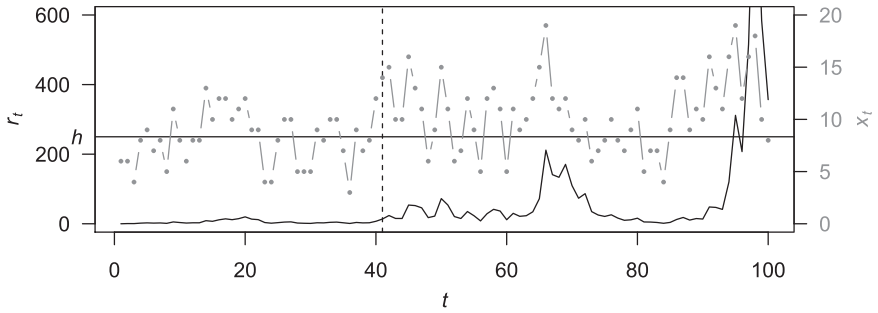

FIGURE 3 SR chart (black) for Poi-INARCH(1) sample path (gray). Change point at $\tau = 41$, control limit $h = 250$ is exceeded at $t = 95$

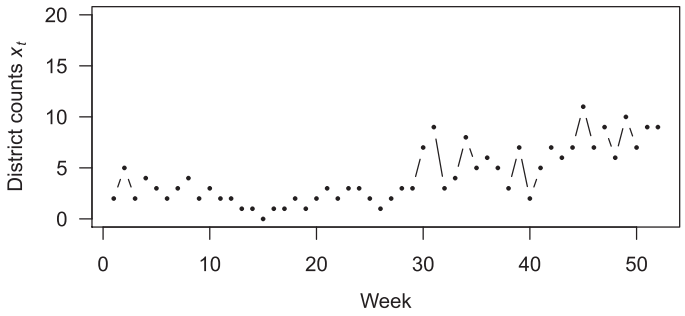

FIGURE 4 Time series of weekly counts of German districts with new cases of hantavirus infections in 2011[21]

alarm at $t = 66$. As noted by Kenett and Pollak,[6] this intuitive interpretability of the value of the SR statistic multiplied by the constant c is unique to the SR procedure and certainly an advantage over other control schemes (such as CUSUM).

5.2 | District counts with hantavirus infections

Let us now consider the possible effects estimated design parameters can have on the adjustment of the SR chart. For this purpose, we conduct a brief bootstrap study on the basis of the Bin-INARCH(1) model Weiß and Pollett[21] fitted to a count time series of German districts with newly reported hantavirus infections (see Figure 4). The data include 52 observations and have a natural upper bound of $n = 38$ (the total number of districts in Germany), justifying the use of a bounded-counts model like the Bin-INARCH(1). The empirical mean and variance are 4.17 and 7.79, respectively. Together with the empirical autocorrelation function $\tilde{\rho}(k) = 0.63^k$, the moment estimates of b and a can be calculated using (11). These estimates, in turn, serve as starting values for the ML estimation (using (12)), which finally yields $\tilde{b} \approx 0.75$ and $\tilde{a} \approx 0.03$. For a parametric bootstrap approach, $\tilde{b}$ and $\tilde{a}$ are considered to be the true model parameters, which are used to generate 500 Bin-INARCH(1) time series of length 52, just like the size of the original data set. For each bootstrap replicate, moment and ML estimations are carried out so that 500 estimates of b and a are obtained. These are the in-control parameters of the Bin-INARCH(1) SR charts b_0 and a_0, see formula (13); their distribution (without outliers) is shown in the left boxplots of the first two diagrams of Figure 5.

To keep the setup simple, the design parameters β_1 and α_1 are uniformly derived from the in-control parameters by multiplying all b_0 with 1.2 and all a_0 with 1. The corresponding control limits shall be chosen so that each chart has a planned ARL to false alarm of about 370. To determine the limits as quickly as possible, one can make use of the asymptotic linearity of the SR chart between the ARL to false alarm and the control limit, that is, $\lim_{h\to\infty} E_{\tau=\infty}[L] = c \cdot h$ (see Section 2). Taking $h = 370$ as the initial control limit for all charts, we can compute the constant c for each Bin-INARCH(1) SR chart by simulating the corresponding ARL to false alarm and solving the above equation for c. Subsequently, the control limits inducing a planned ARL to false alarm of 370 are given by calculating $370/c$. The distribution of the obtained control limits can be seen in Figure 5, too. For the reference SR chart with the true model parameters, the resulting control limit is $h \approx 346.58$. Considering the variation among the bootstrapped control limits (Figure 5), one might think that a range of about 320 to 360 should not have such a great impact on the ARL to false alarm. But since the fitted rather than the true model parameters are likewise part of the chart design, the true ARL to false alarm can considerably deviate from the planned one, as the left boxplot of the last diagram in Figure 5 illustrates. Presumably, the situation would improve,

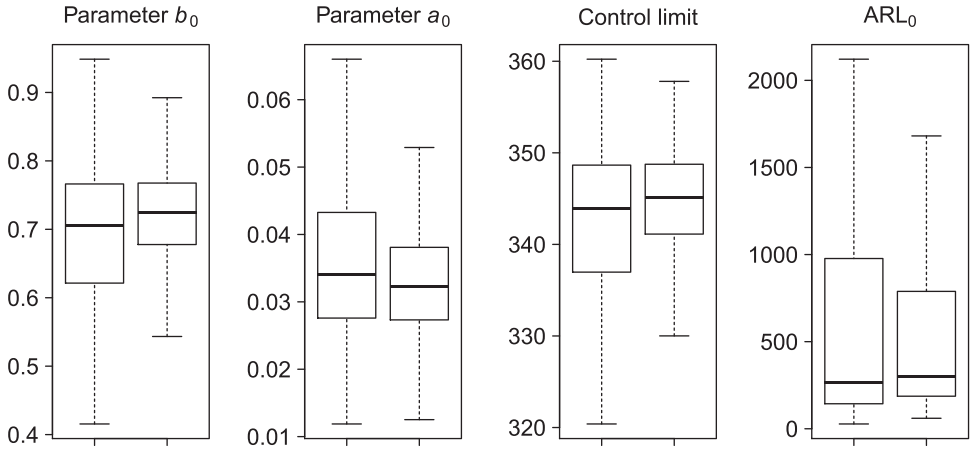

FIGURE 5 Boxplots of design parameters and true ARL to false alarm for bootstrap length 52 (left) and 104 (right)

if the lengths of the available time series were longer. Assume, for instance, an underlying observation period of 2 years for the infection counts. In that case, a corresponding parametric bootstrap experiment shows that the dispersions of the above parameters and, thus, of the ARL to false alarm visibly decrease, as can be seen from the right boxplots of each diagram in Figure 5. Hence, when adjusting the Bin-INARCH(1) SR chart, a practitioner should try to use a larger in-control sample size in order to reduce the estimation uncertainty concerning the intended ARL to false alarm as much as possible. Of course, this phenomenon is not limited to the SR chart but is well-known for a wide class of control charts in the SPC literature, see, for example, Jensen et al.[27] However, detailed recommendations regarding the required sample size for count time series with different dependence levels and structures are yet missing and should therefore be further investigated in future research.

6 | CONCLUSION AND OUTLOOK

Using the Poi-INARCH(1) process as the representative case for modeling count time series, the above performance analysis revealed the SR procedure to be as good as its more popular competitor, the CUSUM method. Which of the two specifically performs better depends on the considered run length concept: The CUSUM method appears to be the better choice under zero-state measures, whereas the SR procedure has a superior steady-state performance. This is consistent with related propositions and investigations for serial independent observations in the literature (e. g., Moustakides et al.[17]). If, however, the true postchange parameters can be expected to deviate downward from the actual design parameters of the chart, the SR chart can be recommended for monitoring Poi-INARCH(1) processes independently from the run length concept. The key advantage of SR is the linearity of the control limit in the ARL to false alarm. It enables a fast and simple adjustment of the chart (see the application example in Section 5.1), whereas the CUSUM design relies on a computationally more demanding bisection method. To sum up, at an overall performance that is at least as good as the one of the CUSUM method, the SR chart benefits from its practical applicability and, thus, is the more appropriate choice for monitoring Markovian count processes such as the Poi-INARCH(1) process.

Like other monitoring schemes, the SR chart can be applied to a wide range of processes. In the context of count time series, the above evaluation study can easily be adopted to other established models, such as the INAR(1) or the binomial AR(1) process of McKenzie,[9] by inserting the transition probabilities of the respective Markov processes in (4) and (5). As part of future research, it would be interesting to ascertain whether the advantages of SR even apply for more complex models. Take, for instance, hidden-Markov models (HMMs), which have gained more and more attention in recent years. Ottenstreuer et al.[36] found that CUSUM charts based on the HMM's likelihood function show considerable deficiencies in detecting changes within a Poisson HMM's structure. Finding out whether the SR method is a more useful alternative (also in terms of the computing time) is an appealing task for future research. Further research opportunities are the investigation of SR for categorical processes, or, in view of the above bootstrap study, an extended comparative analysis of the effects of parameter estimation with other control charts. To conclude, there are several research opportunities regarding the SR chart, indicating that the method is still a niche tool in the SPC community. In this respect, the article

tried to show that the SR procedure is a serious alternative to common control charts like CUSUM or Shewhart and deserves more attention than it receives so far.

ACKNOWLEDGMENTS

The author is indebted to Professor Christian Weiß for suggesting a study about the Shiryaev–Roberts Procedure for count time series as a research project, and for many helpful comments in this regard. This work also benefited from the valuable discussions with Professor Sven Knoth, for which the author is very grateful.

ORCID

Sebastian Ottenstreuer https://orcid.org/0000-0001-5988-2173

REFERENCES

1. Page E. Continuous inspection schemes. *Biometrika*. 1954;41(1):100–115.
2. Roberts SW. Control charts tests based on geometric moving averages. *Technometrics*. 1959;1(3):234–250.
3. Shiryaev AN. The problem of the most rapid detection of a disturbance in a stationary process. *Soviet Math Dokl*. 1961;2:795–799.
4. Roberts SW. A comparison of some control chart procedures. *Technometrics*. 1966;8(3):411–430.
5. Pollak M. The Shiryaev-Roberts changepoint detection procedure in retrospect—Theory and practice. In: *Proceedings of the 2nd International Workshop on Sequential Methodology*, Troyes, France: University of Technology of Troyes; 2009:15–17 June.
6. Kenett RS, Pollak M. Data-analytic aspects of the Shiryayev Roberts control chart: surveillance of a non-homogeneous Poisson process. *J Appl Stat*. 1996;23(1):125–137.
7. Alwan LC. Effects of autocorrelation on control chart performance. *Commun Stat – Theory Methods*. 1992;21(4):1025–1049.
8. Harris TJ, Ross WH. Statistical process control procedures for correlated observations. *Can J Chem Eng*. 1991;69(1):48–57.
9. McKenzie E. Some simple models for discrete variate time series. *Water Resour Bull*. 1985;21(4):645–650.
10. Weiß CH. SPC methods for time-dependent processes of counts—a literature review. *Cogent Math*. 2015;2(1):111–116.
11. Rydberg TH, Shephard N. BIN models for trade-by-trade data. Modelling the number of trades in a fixed interval of time. In: *Econometric Society World Congress 2000*, Contributed Papers No 0740, Econometric Society.
12. Heinen A. Modelling time series count data: an autoregressive conditional Poisson model. CORE Discussion Paper 2003/62, Belgium: University of Louvain.
13. Ferland R, Latour A, Oraichi D. Integer-valued GARCH processes. *J Time Ser Anal*. 2006;27(6):923–942.
14. Weiß CH. *An Introduction to Discrete-Valued Time Series*. Chichester: John Wiley & Sons, Inc; 2018.
15. Weiß CH. The INARCH(1) model for overdispersed time series of counts. *Commun Stat–Simul Comput*. 2010;39(6):1269–1291.
16. Weiß CH, Testik MC. Detection of abrupt changes in count data time series: cumulative sum derivations for INARCH(1) models. *J Qual Technol*. 2012;44(3):249–264.
17. Moustakides GV, Polunchenko AS, Tartakovsky AG. Numerical comparison of CUSUM and Shiryaev-Roberts procedures for detecting changes in distributions. *Commun Stat-Theory Methods*. 2009;38(16–17):3225–3239.
18. Pollak M, Siegmund D. A diffusion process and its applications to detecting a change in the drift of Brownian motion. *Biometrika*. 1985;72(2):267–280.
19. Polunchenko AS, Raghavan V. Comparative performance analysis of the cumulative sum chart and the Shiryaev-Roberts procedure for detecting changes in autocorrelated data. *Appl Stochastic Models Bus Ind*. 2018;34(6):922–948.
20. Freeland RK. *Statistical Analysis of Discrete Time Series with Applications to the Analysis of Workers Compensation Claims Data* (PhD thesis), Canada: University of British Columbia; 1998. Retrieved from https://open.library.ubc.ca/cIRcle/collections/ubctheses/831/items/1.0088709
21. Weiß CH, Pollett PK. Binomial autoregressive processes with density-dependent thinning. *J Time Ser Anal*. 2014;35:115–132.
22. Crosier RB. A new two-sided cumulative sum quality control scheme. *Technometrics*. 1986;28(3):187–194.
23. Pollak M. Average run lengths of an optimal method of detecting a change in distribution. *Ann Stat*. 1987;15(2):749–779.
24. Yakir B. On the average run length to false alarm in surveillance problems which possess an invariance structure. *Ann Stat*. 1998;26(3):1198–1214.
25. Neumann MH. Absolute regularity and ergodicity of Poisson count processes. *Bernoulli*. 2011;17(4):1268–1284.
26. Weiß CH. Modelling time series of counts with overdispersion. *Stat Method Appl*. 2009;18(4):507–519.
27. Jensen WA, Jones-Farmer LA, Champ CW, Woodall WH. Effects of parameter estimation on control chart properties: a literature review. *J Qual Technol*. 2006;38(4):349–364.
28. Weiß CH, Pollett PK. Binomial autoregressive processes with density dependent thinning. *J Time Ser Anal*. 2014;35(2):115–132.
29. Ahmad S, Riaz M, Abbasi SA, Lin Z. On monitoring process variability under double sampling scheme. *Int J Prod Econ*. 2013;142(2):388–400.
30. Ahmad S, Riaz M, Hussain S, Abbasi SA. On auxiliary information-based control charts for autocorrelated processes with application in manufacturing industry. *Int J Adv Manuf Technol*. 2019;100:1965–1980.
31. Ahmad S, Riaz M, Abbasi SA, Lin Z. On median control charting under double sampling scheme. *Eur J Ind Eng*. 2014;8(4):478–512.
32. Wu Z, Jiao JX, Yang M, Liu Y, Wang ZJ. An enhanced adaptive CUSUM control chart. *IIE Trans*. 2009;41(7):642–653.
33. Moustakides GV. Optimal stopping times for detecting changes in distributions. *Ann Stat*. 1986;14(4):1379–1387.

34. Lorden G. Procedures for reacting to a change in distribution. *Ann Math Stat*. 1971;42(6):1897–1908.
35. Pollak M, Tartakovsky AG. Optimality properties of the Shiryaev-Roberts procedure. *Statistica Sinica*. 2009;19(4):1729–1739.
36. Ottenstreuer S, Weiß CH, Knoth S. Control charts for monitoring a Poisson hidden Markov process. *Qual Reliab Eng Int*. 2021;37(2):1–18.
37. Brook D, Evans DA. An approach to the probability distribution of CUSUM run length. *Biometrika*. 1972;59(3):539–549.

AUTHOR BIOGRAPHY

Sebastian Ottenstreuer is a research assistant in the Department of Mathematics and Statistics at the Helmut Schmidt University in Hamburg, Germany. He received his BSc (2013) and MSc (2016) degrees in economathematics from the University of Würzburg, Germany.

How to cite this article: Ottenstreuer S. The Shiryaev–Roberts control chart for Markovian count time series. *Qual Reliab Eng Int*. 2022;38:1207–1225. https://doi.org/10.1002/qre.2945

APPENDIX A: FURTHER RESULTS

TABLE A.1 Zero-state MDRLs of SR and CUSUM charts for Poi-INARCH(1) processes with $(\beta_0, \alpha_0) = (3.5, 0.3)$. If $\text{MDRL}^{\text{SR}} \leq \text{MDRL}^{\text{CUSUM}}$ (italic values), MDRL^{SR} in bold. Table cells in gray where $(\beta, \alpha) = (\beta_1, \alpha_1)$

(β, α)	(β_1, α_1)	$\delta_1 \cdot (\beta_0, \alpha_0)$				$(\delta_1 \cdot \beta_0, \alpha_0)$				$(\beta_0, \delta_1 \cdot \alpha_0)$			
	δ_1	1.1	1.25	1.5	2	1.1	1.25	1.5	2	1.1	1.25	1.5	2
	δ h	**313**	**250.5**	**175.5**	**95.5**	**333**	**292**	**238**	**166.5**	**343.5**	**310.5**	**261.5**	**185.5**
	ARL_0	366.3	366.7	366.7	369.9	367.5	367.9	367.6	367.4	367.8	367.4	367.0	367.5
		366.2	*366.9*	*366.9*	*369.4*	*367.7*	*367.5*	*367.6*	*367.4*	*367.7*	*367.4*	*367.5*	*367.7*
$\delta \cdot (\beta_0, \alpha_0)$	1.025	162	**156**	**169**	**190**	175	**153**	**159**	**174**	223	172	159	**167**
		150	*159*	*174*	*193*	*151*	*153*	*163*	*179*	*153*	*152*	*157*	*170*
	1.05	111	**101**	**116**	**142**	127	101	**104**	**121**	173	122	106	**113**
		96	*104*	*122*	*147*	*99*	*98*	*109*	*128*	*101*	*98*	*103*	*117*
	1.1	68	54	**60**	**83**	83	57	**53**	**64**	120	76	59	**59**
		52	*53*	*64*	*88*	*56*	*51*	*55*	*69*	*57*	*52*	*52*	*61*
	1.25	32	20	**18**	**23**	43	25	19	**18**	60	34	24	19
		20	*17*	*18*	*25*	*24*	*19*	*17*	*19*	*22*	*19*	*17*	*17*
	1.5	17	10	8	**7**	26	14	9	**7**	30	16	11	8
		10	*8*	*7*	*7*	*13*	*10*	*8*	*7*	*10*	*8*	*7*	*7*
	2	9	5	4	**3**	15	8	5	4	12	7	5	4
		5	*4*	*3*	*3*	*7*	*5*	*4*	*3*	*4*	*4*	*4*	*3*
	2.5	6	4	**3**	**2**	11	6	4	**3**	7	5	4	3
		4	*3*	*3*	*2*	*5*	*4*	*3*	*3*	*3*	*3*	*3*	*2*
$(\delta \cdot \beta_0, \alpha_0)$	1.025	188	**185**	**197**	**213**	199	**181**	**188**	**200**	247	199	**188**	**197**
		178	*189*	*202*	*213*	*178*	*181*	*191*	*204*	*183*	*182*	*188*	*199*
	1.05	140	**136**	**154**	**177**	153	**131**	**139**	**159**	204	153	141	**153**
		128	*141*	*160*	*179*	*127*	*131*	*145*	*164*	*134*	*133*	*140*	*156*
	1.1	92	**82**	**97**	**125**	106	80	**84**	**103**	153	104	88	**96**
		76	*84*	*104*	*128*	*78*	*77*	*88*	*109*	*83*	*80*	*84*	*100*
	1.25	47	33	**34**	**51**	59	37	**31**	**36**	88	53	38	**35**
		32	*30*	*37*	*54*	*35*	*30*	*31*	*40*	*36*	*33*	*31*	*35*

(Continues)

TABLE A.1 (Continued)

(β, α)	(β_1, α_1)	$\delta_1 \cdot (\beta_0, \alpha_0)$				$(\delta_1 \cdot \beta_0, \alpha_0)$				$(\beta_0, \delta_1 \cdot \alpha_0)$			
	δ_1	**1.1**	**1.25**	**1.5**	**2**	**1.1**	**1.25**	**1.5**	**2**	**1.1**	**1.25**	**1.5**	**2**
	δ h	**313**	**250.5**	**175.5**	**95.5**	**333**	**292**	**238**	**166.5**	**343.5**	**310.5**	**261.5**	**185.5**
	1.5	27	17	14	**17**	37	21	15	**14**	52	29	20	15
		16	*14*	*13*	*18*	*20*	*15*	*13*	*14*	*18*	*15*	*14*	*13*
	2	16	9	7	**6**	24	13	8	**6**	28	15	10	7
		9	*7*	*6*	*6*	*11*	*9*	*7*	*6*	*9*	*7*	*6*	*6*
	2.5	12	7	5	**4**	19	10	6	5	19	10	7	5
		6	*5*	*4*	*4*	*9*	*7*	*5*	*4*	*6*	*5*	*4*	*4*
$(\beta_0, \delta \cdot \alpha_0)$	1.025	225	**215**	**217**	**229**	239	219	**216**	**221**	276	230	216	**217**
		213	*215*	*220*	*230*	*219*	*215*	*217*	*222*	*216*	*212*	*212*	*217*
	1.05	190	**179**	**186**	**204**	206	183	**182**	**190**	245	195	180	**185**
		177	*180*	*190*	*208*	*182*	*180*	*184*	*195*	*178*	*175*	*178*	*186*
	1.1	142	**130**	**139**	**164**	161	133	**133**	**144**	201	147	131	**136**
		126	*130*	*143*	*169*	*132*	*130*	*135*	*150*	*127*	*124*	*127*	*138*
	1.25	78	61	**64**	**86**	96	67	**61**	**68**	129	82	64	**62**
		60	*59*	*67*	*92*	*66*	*60*	*62*	*73*	*63*	*58*	*57*	*63*
	1.5	43	28	**26**	**34**	57	34	27	**27**	77	45	31	26
		28	*25*	*26*	*37*	*33*	*27*	*25*	*28*	*30*	*27*	*24*	*25*
	2	20	12	10	**10**	30	16	11	**9**	35	19	13	10
		12	*10*	*9*	*10*	*15*	*12*	*10*	*9*	*12*	*11*	*9*	*9*
	2.5	12	7	6	**5**	19	10	7	6	17	10	8	6
		7	*6*	*5*	*5*	*9*	*7*	*6*	*5*	*7*	*6*	*6*	*5*

TABLE A.2 Steady-state MDRLs of SR and CUSUM charts for Poi-INARCH(1) processes with $(\beta_0, \alpha_0) = (3.5, 0.3)$. If $\text{MDRL}^{\text{SR}} \leq \text{MDRL}^{\text{CUSUM}}$ (italic values), MDRL^{SR} in bold. Table cells in gray where $(\beta, \alpha) = (\beta_1, \alpha_1)$

(β, α)	(β_1, α_1)	$\delta_1 \cdot (\beta_0, \alpha_0)$				$(\delta_1 \cdot \beta_0, \alpha_0)$				$(\beta_0, \delta_1 \cdot \alpha_0)$			
	δ_1	**1.1**	**1.25**	**1.5**	**2**	**1.1**	**1.25**	**1.5**	**2**	**1.1**	**1.25**	**1.5**	**2**
	δ h	**313**	**250.5**	**175.5**	**95.5**	**333**	**292**	**238**	**166.5**	**343.5**	**310.5**	**261.5**	**185.5**
	ARL_0	366.3	366.7	366.7	369.9	367.5	367.9	367.6	367.4	367.8	367.4	367.0	367.5
		366.2	*366.9*	*366.9*	*369.4*	*367.7*	*367.5*	*367.6*	*367.4*	*367.7*	*367.4*	*367.5*	*367.7*
$\delta \cdot (\beta_0, \alpha_0)$	1.025	**117**	**142**	**164**	**188**	**109**	**130**	**149**	**171**	**86**	**116**	**136**	**159**
		133	*152*	*171*	*191*	*127*	*141*	*158*	*177*	*128*	*137*	*150*	*167*
	1.05	**75**	**90**	**112**	**141**	**73**	**82**	**96**	**118**	**61**	**76**	**86**	**106**
		83	*99*	*119*	*146*	*81*	*89*	*103*	*125*	*82*	*87*	*97*	*114*
	1.1	**42**	**45**	**57**	**82**	**43**	**43**	**47**	**62**	**37**	**43**	**45**	**54**
		44	*49*	*62*	*87*	*44*	*45*	*52*	*68*	*45*	*45*	*48*	*59*
	1.25	18	**15**	**16**	**22**	20	17	**15**	**17**	**16**	17	**16**	**16**
		17	*15*	*17*	*25*	*19*	*16*	*15*	*18*	*17*	*16*	*16*	*16*
	1.5	9	**7**	**7**	**7**	11	9	**7**	**7**	**8**	**8**	**7**	**7**
		8	*7*	*7*	*7*	*10*	*8*	*7*	*7*	*8*	*8*	*7*	*7*
	2	**5**	**4**	**4**	**3**	**6**	**5**	**4**	**4**	**4**	**4**	**4**	**4**
		5	*4*	*4*	*3*	*6*	*5*	*4*	*4*	*5*	*4*	*4*	*4*
	2.5	**4**	**3**	**3**	**3**	5	**4**	**3**	**3**	**4**	**3**	**3**	**3**
		4	*3*	*3*	*3*	*4*	*4*	*3*	*3*	*4*	*3*	*3*	*3*

(Continues)

TABLE A.2 (Continued)

(β, α)	(β_1, α_1)	$\delta_1 \cdot (\beta_0, \alpha_0)$				$(\delta_1 \cdot \beta_0, \alpha_0)$				$(\beta_0, \delta_1 \cdot \alpha_0)$			
	δ_1	**1.1**	**1.25**	**1.5**	**2**	**1.1**	**1.25**	**1.5**	**2**	**1.1**	**1.25**	**1.5**	**2**
	δ h	**313**	**250.5**	**175.5**	**95.5**	**333**	**292**	**238**	**166.5**	**343.5**	**310.5**	**261.5**	**185.5**
$(\delta \cdot \beta_0, \alpha_0)$	1.025	**139**	**171**	**192**	**211**	**127**	**156**	**177**	**197**	**99**	**139**	**164**	**189**
		160	*181*	*198*	*213*	*152*	*169*	*185*	*201*	*155*	*166*	*180*	*195*
	1.05	**99**	**123**	**149**	**175**	**92**	**109**	**130**	**155**	**76**	**101**	**119**	**144**
		113	*135*	*157*	*178*	*106*	*120*	*140*	*163*	*111*	*119*	*133*	*154*
	1.1	**60**	**71**	**93**	**124**	**58**	**63**	**76**	**100**	**51**	**63**	**70**	**89**
		65	*80*	*102*	*128*	*63*	*69*	*84*	*108*	*67*	*70*	*79*	*98*
	1.25	27	**26**	**32**	**50**	**28**	**26**	**26**	**35**	**25**	**27**	**27**	**31**
		26	*27*	*35*	*53*	*28*	*26*	*28*	*39*	*29*	*28*	*28*	*34*
	1.5	14	**12**	**12**	**16**	16	14	**12**	**12**	**13**	14	13	**12**
		13	*12*	*13*	*18*	*15*	*13*	*12*	*13*	*14*	*13*	*12*	*13*
	2	8	**6**	6	**5**	**9**	8	**6**	**5**	**7**	**7**	**6**	**6**
		7	*6*	*5*	*6*	*9*	*7*	*6*	*5*	*7*	*7*	*6*	*6*
	2.5	6	5	**4**	4	**7**	6	5	**4**	**5**	**5**	5	**4**
		5	*4*	*4*	*3*	*7*	*5*	*4*	*4*	*5*	*5*	*4*	*4*
$(\beta_0, \delta \cdot \alpha_0)$	1.025	**173**	**200**	**212**	**228**	**160**	**191**	**205**	**217**	**115**	**166**	**191**	**209**
		193	*207*	*217*	*230*	*189*	*201*	*210*	*221*	*185*	*195*	*204*	*214*
	1.05	**142**	**165**	**182**	**203**	**133**	**158**	**172**	**187**	**98**	**136**	**157**	**177**
		159	*173*	*187*	*207*	*156*	*167*	*178*	*193*	*150*	*160*	*170*	*183*
	1.1	**101**	**117**	**134**	**162**	**98**	**112**	**124**	**140**	**75**	**97**	**110**	**128**
		112	*125*	*140*	*168*	*111*	*119*	*130*	*148*	*105*	*112*	*120*	*135*
	1.25	**49**	**52**	**61**	**85**	**52**	**51**	**55**	**66**	**41**	**47**	**49**	**57**
		51	*55*	*66*	*91*	*53*	*53*	*58*	*72*	*50*	*50*	*53*	*62*
	1.5	**24**	**22**	**24**	**33**	27	**24**	**23**	**25**	**22**	**23**	**22**	**23**
		24	*23*	*25*	*37*	*26*	*24*	*23*	*28*	*24*	*23*	*22*	*24*
	2	**11**	10	**9**	**10**	13	11	10	**9**	**10**	**10**	**9**	**9**
		11	*9*	*9*	*10*	*12*	*10*	*9*	*9*	*10*	*10*	*9*	*9*
	2.5	**7**	**6**	**6**	**6**	9	**7**	**6**	**6**	**6**	**6**	**6**	**6**
		7	*6*	*6*	*6*	*8*	*7*	*6*	*6*	*7*	*6*	*6*	*6*

APPENDIX B: RUN LENGTH COMPUTATIONS

The ARLs and MDRLs in this article are not computed exactly but approximated via simulations (10^6 replications). Simulating a path of an Poi-INARCH(1) process is, however, somewhat cumbersome, since an explicit expression for the marginal distribution the process can be initialized with is not known.[15] Instead, we utilize its ergodicity, that is, the fact that

$$P(X_t = i) = \lim_{n \to \infty} P(X_t = i \mid X_{t-n} = j) \text{ for all } i, j \in \mathbb{N}_0,$$

to basically obtain a marginal distribution close to the required stationary distribution (see Weiß[15] for more information about this approach). For this purpose, a prerun of length $n = 2000$ (analogous to Weiß and Testik[16]) is generated each time before starting the actual simulation of the Poi-INARCH(1) process. So by setting $X_{-2000} := \mu_0 = \beta_0/(1 - \alpha_0)$ and gradually generating new $X_t \sim \text{Poi}(\beta_0 + \alpha_0 x_{t-1})$ for $t = -1999, \ldots, 0$, the resulting X_0 should be distributed nearly the same as under the true prechange probability measure. After having generated $X_0 = x_0$, the regular data for process monitoring, $X_1, X_2, \ldots$, can be simulated according to $X_t \sim \text{Poi}(\beta + \alpha x_{t-1})$, $t = 1, 2, \ldots$, where the model parameters depend on the specific scenario considered; for example, $(\beta, \alpha) = (\beta_0, \alpha_0)$ for in-control or $(\beta, \alpha) \neq (\beta_0, \alpha_0)$ for out-of-control zero-state considerations. The convention that $X_{-2000}, \ldots, X_0$ are based on the in-control model—regardless of the control state

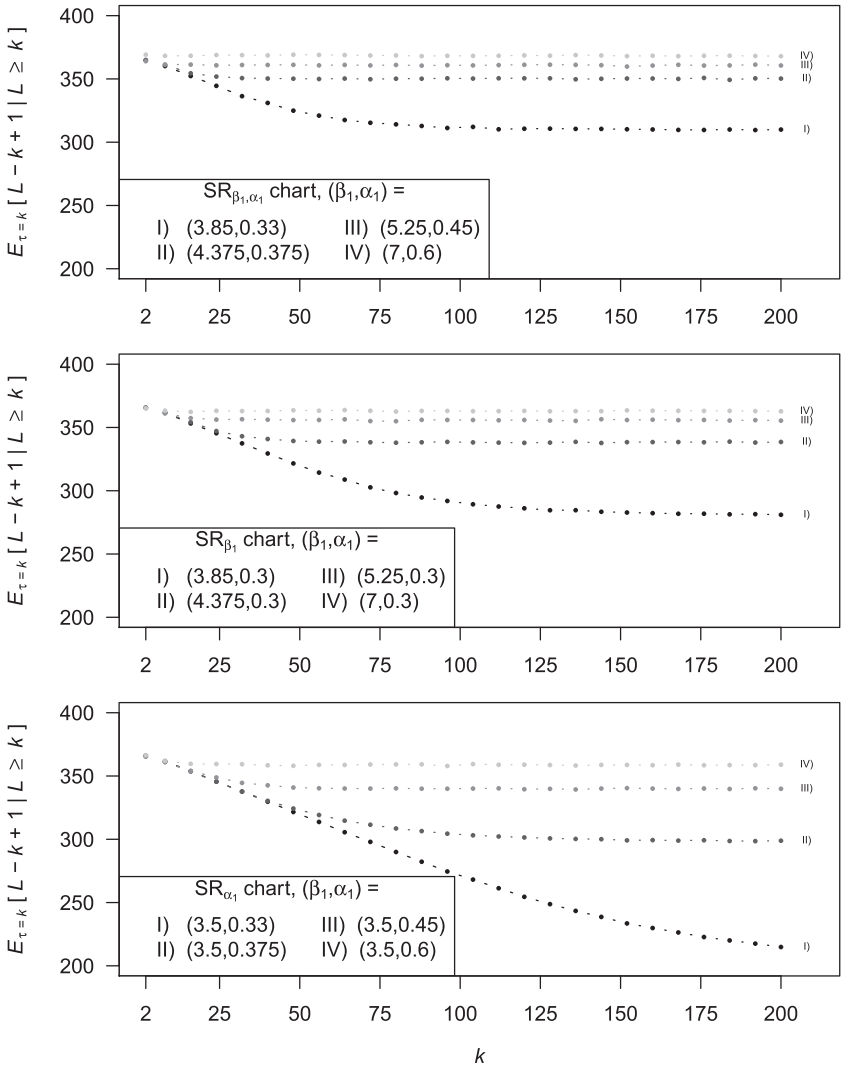

FIGURE B.1 Conditional expected delay of Poi-INARCH(1) SR charts against change point $\tau = k$ for $(\beta, \alpha) = (\beta_0, \alpha_0) = (3.5, 0.3)$. Corresponding control limits in Table 2 or 4

of the actual observations $X_1, X_2, \ldots$—is adopted from Weiß and Testik[16] in order to have comparable values in Section 4. In principle, other approaches are also conceivable, such as using postchange parameters for the prerun already (when considering out-of-control scenarios).

For simulating the zero-state ARL and MDRL of a chart, each path, $X_1, X_2, \ldots$, is generated until the corresponding statistic R_t exceeds the chart's control limit h at some time L. The empirical mean and median over all these stopping times then yield the estimated values for the true ARL and MDRL, respectively. The different (β, α) values used for simulation in this study are listed in Section 4.

Of course, a steady-state run length cannot be simulated directly. As stated in Section 4, $(L - 199 \mid L \geq 200)$ therefore serves as an estimator for $\lim_{\tau\to\infty}(L - \tau + 1 \mid L \geq \tau)$ in this research. The goodness of this approximation concerning the steady-state ARLs of the investigated SR charts can be inferred from Figure B.1, where the convergence behavior of $E_{\tau=k}[L - k + 1 \mid L \geq k]$ is shown. All curves, except one, are constant for $\tau \geq 175$ (apart from minor oscillations due to simulations), indicating that the conditional expected delay with $\tau = 200$ indeed approximates the steady-state ARL pretty well. For the chart with $(\beta_1, \alpha_1) = (3.5, 0.33)$, $\tau = 200$ is not sufficient for an approximation, as can be seen from the lower graph of Figure B.1 In this particular case, $E_{\tau=k}[L - k + 1 \mid L \geq k]$ converges rather slowly, so that only $\tau = 450$ was found to be good enough. For reasons of clarity, the continued curve is omitted here. Note that for each chart, the rate of convergence is independent of the specific control state of the underlying process. (This directly follows from the Markov chain approach of Brook and Evans[37] for calculating steady-state ARLs, which can be found, e.g., in Weiß.[14] However, the approach can be applied only approximately, due to the quasi-continuous distribution of R_t, and, hence, does not yield more accurate results than simulations.)

For every path of $(L - 199 \mid L \geq 200)$, it must be guaranteed that the first 199 realizations of $(X_t)_{\mathbb{N}}$ stem from the in-control model, that is, $(\beta, \alpha) = (\beta_0, \alpha_0)$, and that R_t does not cause a false alarm for $t = 1, \ldots, 199$. For this reason, any run not fulfilling this condition is neither taken into account nor numbered among the 10^6 replications used for one simulation. If the condition is fulfilled, the path is continued by generating $X_{200}, X_{201}, \ldots$ according to either $(\beta, \alpha) = (\beta_0, \alpha_0)$ (in-control) or $(\beta, \alpha) \neq (\beta_0, \alpha_0)$ (out-of-control). The simulation is terminated when the SR statistic exceeds the threshold the first time $L \geq 200$. The empirical mean and median over all $L - 199$ are finally utilized for estimating $E_{\tau=200}[L - 199 \mid L \geq 200]$ and $MED_{\tau=200}[L - 199 \mid L \geq 200]$, respectively.