Economic Design of CUSUM Control Charts

by

Thomas Croskery

April, 2014

Director of Thesis: Dr. Christopher Carolan, Ph.D.

Major Department: Mathematics

In statistical process control, control charts are one tool for monitoring the control status of a process. One such type of chart is the cumulative sum (CUSUM) chart which has advantages over other styles of control chart. A study of the economic design of CUSUM control charts is undertaken via a comparative study of long-run hourly cost (LRHC) and a computational search algorithm is used to minimize LRHC for a CUSUM chart using nine parameters confined to their respective feasible parameter spaces as defined by the chart designer. Savings over similarly designed two stage Xbar charts are discovered and presented.

Economic Design of CUSUM Control Charts

A Thesis

Presented to the Faculty of the Department of Mathematics

East Carolina University

In Partial Fulfillment of the Requirements for the Degree

Master of Arts in Mathematics

By

Thomas Croskery

April, 2014

Economic Design of CUSUM Control Charts

by

Thomas Croskery

APPROVED BY:

DIRECTOR OF
DISSERTATION/THESIS: ____
Chris Carolan, Ph.D.

COMMITTEE MEMBER: ____
Ali Al-sharadqah, Ph.D.

COMMITTEE MEMBER: ____
John Kros, Ph.D.

COMMITTEE MEMBER: ____
Chris Jantzen, Ph.D

CHAIR OF THE DEPARTMENT
OF MATHEMATICS: ____
Johannes Hattingh, Ph.D.

DEAN OF THE
GRADUATE SCHOOL: ____
Paul J. Gemperline, PhD

ACKNOWLEDGMENTS

I would like to thank my advisor for giving me the tools to succeed. I would also like to thank my wife for putting up with me.

TABLE OF CONTENTS

LIST OF TABLES v

LIST OF FIGURES vi

CHAPTER 1: STATISTICAL PROCESS CONTROL AND CONTROL CHARTS 1

Overview of Statistical Process Control 1

Control Charts 1

CUSUM Control Charts 2

Refinements to the CUSUM chart 4

Variable Sample Size and Interval Control Charts 6

Extension of VSSI to CUSUM charts 8

CHAPTER 2: ECONOMIC DESIGN AND LONG RUN HOURLY COST 9

Economic Design of Control Charts 9

Discussion of LRHC Components 10

CHAPTER 3: MARKOV CHAINS AND USEFUL PROPERTIES 12

Markov Chains and the Transition Probability Matrix 12

Transient and Recurrent States 13

Expected visits to Transient States 13

A Numerical Example 14

Applications of Markov Chains to CUSUM Control Charts 16

CHAPTER4: A MARKOV MODEL OF CUSUM CONTROL CHART ECONOMIC PERFOMRANCE 17

Model Assumptions and Justification 17

The CUSUM Chart and Associated Design Parameters 18

VSSI Using Maps to break the ‘Curse of Dimensionality’ 19

Modeling the CUSUM as a Markov Chain 21

Formulas and Derivations of Nine Transition Probability Cases 24

LHRC Calculations Revisited 29

Economic Design by Algorithmic Programming Using our Markov Model 31

An Example of our CUSUM Chart in Operation 32

CHAPTER 5: COMPARATIVE STUDY AND RESULTS 36

Preliminary Investigation 36

Comparison Scenarios 37

Summary of Results 40

REFERENCES 41

APPENDIX A: TRANSITION PROBABILITY DERIVATIONS 43

APPENDIX B: R/S-PLUS CODE FOR LRHC SEARCH 54

LIST OF TABLES

TABLE 1: SCENARIOS.. 38

TABLE 2: ECONOMICALLY DESIGNED CHART PARAMETERS............................. 39

LIST OF FIGURES

FIGURE 1: CUSUM CHART EXAMPLE.. 4

FIGURE 2: CUSUM CHART WITH V-MASK.. 5

FIGURE 3: VARIABLE SAMPLE SIZE MAP.. 20

FIGURE 4: VARIABLE SAMPLING INTERVAL MAP.. 21

FIGURE 5: CUSUM CHART AT TIME 0... 32

FIGURE 7: CUSUM CHART AT TIME 2H0.. 33

FIGURE 8: CUSUM CHART AT TIME 2H0.. 34

FIGURE 9: CUSUM CHART AT 2H0+H0.75.. 35

CHAPTER 1: STATISTICAL PROCESS CONTROL AND CONTROL CHARTS

Overview of Statistical Process Control

In a wide variety of applications, it is desirable that some particular process results in reliably similar products. These products will have one or more variable which must remain within acceptable ranges or tolerances in order to be considered viable. Processes that result in these reliable similar products are said to be “in-control.”

Processes do not remain in-control indefinitely; at some time after the process begins, something will affect a change in the process which results in unacceptable products. The agent of this change is referred to as an “assignable cause.” Once an assignable cause occurs, the process status becomes “out-of-control.”

Out-of-control processes are undesirable. They result in the generation of waste/scrap, they take away from the production of viable products, and they require costly fixes to process equipment. It is therefore desirable to implement some program of monitoring to ascertain the control status (in-control or out-of-control) of a process. This program usually relies on assumed or known statistical properties (distribution, parameters) for the variables to be controlled and assignable causes which may occur coupled with a sampling regimen. The overall program is referred to as statistical process control (SPC).

Control Charts

One particular tool for the implementation of SPC is the control chart. A control chart is a graphic representation of sequential sample statistics coupled with a rule or rules to indicate whether the sample statistics are likely coming from an in-control or out-of-control process. A process engineer will plot sequential sample statistics on the graph according to pre-determined chart parameters. After each statistic is plotted, it is checked against the rule or rules of the chart. If chart indicates the process is in-control, the process is allowed to continue; if the chart indicates the process is out-of-control, a signal is generated and the process is stopped while a search for an assignable cause is carried out.

Control charts were originally developed by Walter Shewart during the first half of the 20th century. These so-called Shewart Charts, also known commonly as Xbar charts, plot the sample mean of fixed size samples at regular intervals; e.g. the sample mean of a sample of size 10 every hour. The chart has upper and lower control-limits drawn at thresholds which are deemed significantly unlikely (at some prescribed level of significance) for a process which is in-control; e.g., 3 standard deviations away from the process mean in either direction. A signal is generated when the sample mean falls either above the upper control limit or below the lower control limit.[1]

Since the sample statistic is a measure of a random variable, there is always a chance that the plotted statistic will fall outside of the control limits when the underlying process is actually in-control. When this occurs, i.e., a signal is generated when no assignable cause exists, it is said that a false signal has occurred. False signals are undesirable as they stop an in-control process, wasting time and money while a search for a non-existent assignable cause is carried out.

CUSUM Control Charts

As process engineers required more sensitive analysis of the process control status, ever more sophisticated rules and charts were developed. One type of control chart, developed by Page during the mid 20th, century was called the **cu**mulative **sum**

[1] Nelson, Lloyd S. *Control Charts*. Wiley, 2005

(CUSUM) chart. This chart plots sequential sums of the sample statistic and compares the result to some threshold of allowable change in the overall trend of the sum. Because the chart uses a sequential sum, information from previous samples is combined with information from current samples, giving a so-called "head-start" on the detection of assignable causes.

The original chart proposed by Page plots a sum of sequential sample scores, s_i, at regular intervals. The sample scores are chosen so that the expected value of $s_i < 0$ for an in-control process, and the expected value of $s_i > 0$ for an out-of-control process. When the distance between the most recent plotted statistic and the minimum plotted statistic is greater than some control-limit, a signal is generated. This is summarized by the following[2]:

$$S_n = \sum_{i=1}^{n} s_i \quad (1)$$

The plotted statistic is (1), and a signal is generated if $S_n - \min\{S_i\}_{1\le i\le n} \ge h$

An example chart follows.

[2] Page, E. S. "Continuous Inspection Schemes." *Biometrika* 41.1/2 (1954): 100-15.

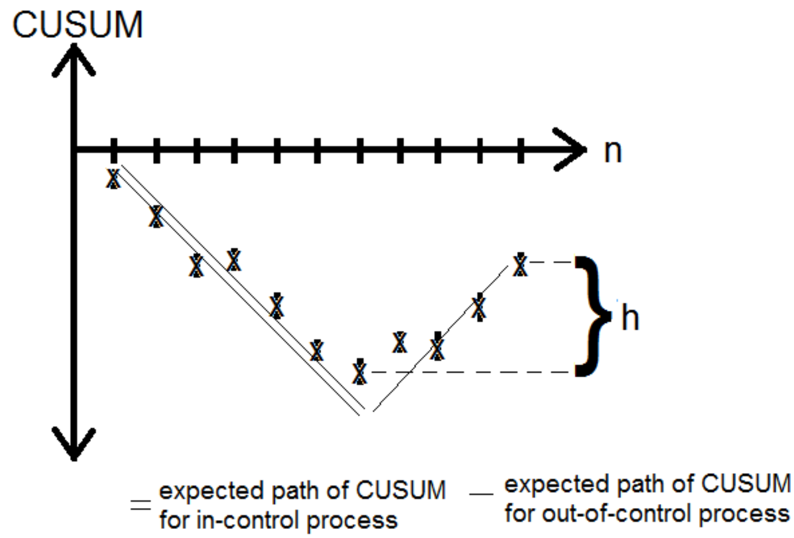

Fig. 1 When the process is in-control, the plotted sums will tend to move along the path given by (=); when the process is out-of-control the plotted sums will tend to move along the path given by (-).[3]

From the above chart it appears that an assignable cause likely occurred between the seventh and eighth sample, and a signal was generated by plotting the eleventh sample statistic.

Refinements to the CUSUM chart

One refinement to the CUSUM chart is proposed by Barnard; by subtracting the expected score, μ_c, from each sample score, c_i, the expected value of the sample statistic becomes zero. Hence the CUSUM, C_n, will tend to remain around zero. The plotted statistic is then given as the following:

(2) $$C_n = \sum_{i=1}^{n}(c_i - \mu_c)$$

[3] Page, E. S. "Continuous Inspection Schemes." *Biometrika* 41.1/2 (1954): 100-15.

A "V" shaped cut-out of half angle α is superimposed a distance d from the most recently plotted statistic. If any portion of the segments joining the plotted statistics fall outside the arms of the so-called V-mask or their extensions, either above the upper arm or below the lower arm, a signal is generated.[4] The figure below illustrates a typical CUSUM chart with such V-masking.

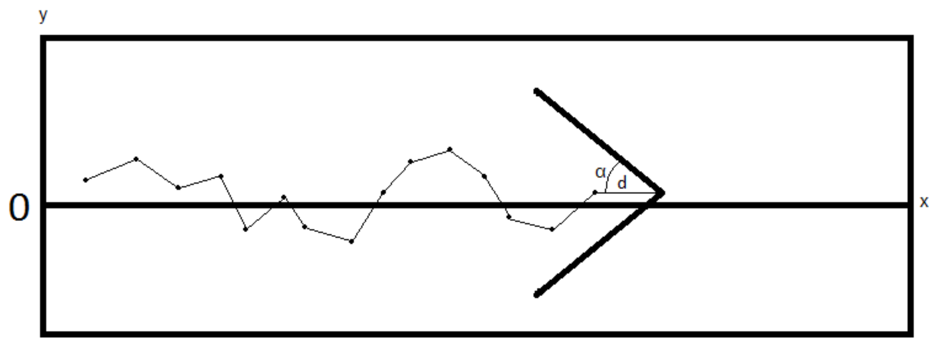

Fig. 2 The plotted statistic is centered around 0 and a superimposed V-mask is added to determine the control status of the process.[5]

A recently proposed refinement to the CUSUM control chart is given by Wu and Yang represented by (3). The first plotted statistic is 0. Subsequent sample scores are standardized by subtracting their expected value and dividing the result by the standard deviation; this standardized result is known as the z-score. This z-score is added to the previous plotted statistic and a reference value, k, is subtracted. The plotted statistic is the maximum between this number and 0.[6]

(3) $$A_n = \begin{cases} 0 \text{ for } n = 0 \\ \max\{0, (A_{n-1} + |z_n| - k)\} \text{ for } n \geq 1 \end{cases}$$

[4] Barnard, G. A. "Control Charts and Stochastic Processes." *Journal of the Royal Statistical Society. Series B (Methodological)* 21.2 (1959): 239-71.
[5] Ibid
[6] Wu, Zhang, et al. "A CUSUM Chart using Absolute Sample Values to Monitor Process Mean and Variance".IEEE , 2009. 414-418.

In-control process will result in a plotted statistic, A_n , so that $A_n \geq 0$. Shifts in the controlled variable resulting from a process running out-of-control will tend to increase the plotted statistic until it crosses some the control-limit, l. When $A_n \geq l$, a signal is generated.

Variable Sample Size and Interval Control Charts

For all of the schemes presented previously there has been either an implicit or explicit assumption that sample sizes and the intervals at which the samples are taken are both constants. Charts which follow these assumptions are referred to as fixed sample size and sampling interval (FSSI) charts. The appeal of these charts is their simplicity.

Other schemes exist, however, which allow for the sample size, sampling interval, or both to vary based on the most recent data from our chart. These are referred to alternatively as variable sample size (VSS), variable sample interval (VSI), or variable sample size and sample interval (VSSI). These charts allow the process engineer to take larger samples, more frequent samples, or both, when recent data indicates that the process may be out-of-control, even if a signal has not been generated. The advantages of each strategy are as follows.

By taking larger sample sizes, the process engineer increases the certainty that the plotted statistic is close to the true underlying parameter for the variable in question. The effect has two-fold benefits: it decreases the chance of a (false) signal if the process is actually in-control, and increases the chance of a (true) signal if the process is actually out-of-control.

Alternatively, by sampling more frequently, the amount of time which elapses between a process running out-of-control and a signal being generated is decreased. This decreases the time spent out-of-control. Less time spent out-of-control means less scrap/waste is generated and less time is wasted producing unacceptable products.

An example of a simple VSSI chart would be a two-stage Xbar chart with control limits, warning limits, sample size $\in \{n_1, n_2\}$ and sampling interval $h \in \{h_1, h_2\}$. If the plotted statistic (sample mean) falls within the warning limits, the sampling size and interval take on values n_1, h_1 respectively. If the plotted statistic falls outside the warning limits but within the control limits, then the sample size and interval take on values n_2, h_2 respectively.

Carolan et al. offer tweak on such a chart by proposing a continuously variable sampling interval. A linear map from a maximum sampling interval to a minimum sampling interval is created which depends upon the relative extremity of the previous sample statistic to the extremity of the control limit. Hence sampling interval is a strictly decreasing function of extremity of most recent sample as summarized below:

(4) $$H_k = \max\left\{t, T * \frac{\Phi(z_c) - \Phi(|z_{k-1}|)}{\Phi(z_c) - 0.5}\right\}$$

Here H_k represents the kth sampling interval, t represents the minimum allowable sampling interval, T represents the maximum allowable sampling interval, z_c represents the standardized score of the upper control-limit, z_{k-1}represents the standardized score of the $(k-1)th$, i.e. previous, sample statistic, and Φ is the standard normal distribution's cumulative density function (CDF). By combining this continuously variable sampling interval with a two-stage sample size Xbar chart, Carolan et al. report economic savings over other similar VSSI Xbar charts.[7]

Extension of VSSI to CUSUM charts

This work extends the advances of Carolan et al. to CUSUM control charts. Under a control chart scheme similar to that mentioned in Wu and Yang, maps are created from the extremity of the CUSUM relative to the control limit (hereafter referred to as the "alarm boundary") to both sampling size and sampling interval. A shape parameter for each map is introduced which influences the rate at which the control chart moves from its maximum sampling interval to minimum sampling interval or from its minimum sampling size to its maximum sampling size.

Using the notion of long-run hourly cost (LRHC) discussed in the next chapter as a measure of control chart economic performance, competing charts are compared and chart parameters are optimized. A comparative study of such "economically designed" CUSUM charts to similarly designed Xbar charts under Carolen et al.'s scheme is undertaken. Economic savings are discovered and reported.

[7] Carolan, CA, J.F. Kros, and S.E. Said. "Economic Design of Xbar Control Charts with Continuously Variable Sampling Intervals." *QUALITY AND RELIABILITY ENGINEERING INTERNATIONAL* 26.3 (2010): 235-45.

CHAPTER 2: ECONOMIC DESIGN AND LONG RUN HOURLY COST (LRHC)

Economic Design of Control Charts

Economic design of control charts refers to the directed selection of chart parameters with the goal of optimizing some economic measure of performance for the associated chart. The economic measure of chart performance is up to the "designer," and is based on whatever economic quality is desired. These may be minimum expected sampling cost, minimum expected false signal costs, etc. The economic measure used herein for economic design will be an all-encompassing metric of expected process monitoring cost over time referred to as long-run hourly cost (LRHC).

LRHC is an account of the expected total cost of running a control chart from the time the process begins in-control until a true signal is generated, an assignable cause is located and repaired, and the chart begun again divided by the expected total time for the same. The time frame between the fixing of assignable causes is referred to as a cycle of the chart. Let TC be the random total cost of completing a cycle of some chart and TT be the random total time for the same, finally let $E(*)$ be the expected value. LRHC is then given as the following:

(5) $$LRHC = \frac{E(TC)}{E(TT)}$$

As various parameters of our CUSUM chart result in different associated LRHCs, we compare LRHCs of competing control charts under a set of assumed constraints and the chart with the minimum LRHC is preferred. This is what we refer to as the economic design of the CUSUM chart. A computer search algorithm for the R/S-Plus statistical environment, detailed in Appendix B, is used to carry out the design process via Markov techniques described in the next chapter.

Discussion of LRHC Components

The expected total cost, $E(TC)$, will depend upon the following cost components:

- Sampling Costs
- Out-of-control Costs
- False-Signal Search Costs
- Assignable Cause Repair Costs

Let c_1 represent the cost associated with sampling one unit, c_2 represent the hourly cost of an out-of-control process, c_3 represent the cost of searching for a false signal, and c_4 represent the hourly cost of repairing an assignable cause. If $E(N)$ represents the expected total number of samples over one cycle of the chart, $E(TOOC)$ represents the expected time out-of-control over one cycle, $E(F)$ represents the expected number of false signals, t_1 represents the expected time to determine a signal is false, and t_2 represents the expected time for repairing a true signal, then expected total cost per cycle is the following:

(6)
$$E(TC) = c_1 E(N) + c_2 E(TOOC) + c_3 E(F) + c_4$$

Similarly if $E(PT)$ is the expected time the process is in operation over one cycle, noting that the process stops whenever a true or false signal is generated while a search for an assignable cause is carried out, then the expected total time for one cycle is the following:

(7) $$E(TT) = E(PT) + t_1 E(F) + t_2$$

Combining these results, we see that LRHC becomes:

(8) $$LRHC = \frac{c_1 E(N) + c_2 E(TOOC) + c_3 E(F) t_1 + c_4}{E(PT) + t_1 E(F) + t_2}$$

We assume $\{c_1, c_2, c_3, c_4, t_1, t_2\}$ to be known for some processes. We discuss the calculation of $E(N)$, $E(TOOC)$, $E(F)$, and $E(PT)$ in Chapter 4; these calculations require the use of Markov techniques discussed in Chapter 3 and are controlled by the selection of chart parameters.

CHAPTER 3: MARKOV CHAINS AND USEFUL PROPERTIES

Markov Chains and the Transition Probability Matrix

Consider a system which randomly transitions between states on a defined state space. Such a system will be considered a Markov chain if the distribution of the next state in the sequence depends only on the current state, and is independent of the sequence of previous states. More formally, let $X_1, X_2, X_{3,}$ … be a sequence of random variables. This sequence is a Markov chain if

$$\Pr(X_{n+1} = x | X_n = x_n, X_{n-1} = x_{n-1}, \dots, X_2 = x_2, X_1 = x_1) = \Pr(X_{n+1} = x | X_n = x_n)$$

The values which each random variable may take come from some set S which represents the state space of the Markov chain. For our purposes we consider only finite state spaces.

If our state space is constituted by m possible states, $S = \{s_1, s_2, \dots s_{m-1}, s_m\}$, and p_{ij} represents the probability that our Markov chain currently in state s_i will next be in state s_j, we can organize an mxm matrix, $\boldsymbol{P}$, such that $\boldsymbol{P}_{ij} = p_{ij}$. Since these entries are probabilities, $0 \leq p_{ij} \leq 1$ for all $i, j \in \{1,2, \dots m\}$. This matrix is referred to as the Transition Probability matrix for our associated Markov chain because its entries are the probabilities of undergoing a transition from one state to another. Since each row represents all possible transitions from some state i, the sum of any row of our matrix $\boldsymbol{P}$ will be 1.

(9) $$\sum_{j=1}^{m} \boldsymbol{P}_{ij} = 1$$

Our matrix will have the following form:

$$
P = \begin{array}{c|cccccc}
state & 1 & 2 & \cdots & & \cdots & \\
\hline
1 & p_{11} & p_{12} & \cdots & & \cdots & \\
2 & p_{21} & p_{22} & \cdots & & \cdots & \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
i & p_{i1} & p_{i2} & \cdots & & \cdots & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
m & p_{m1} & p_{m2} & \cdots & & \cdots &
\end{array}
$$

The matrix $\boldsymbol{P}$ forms the basis of all Markov analysis and as such is the backbone of our model here.

Transient and Recurrent States

Within a Markov chain, we can classify two useful types of states: those to which we will eventually return once we leave, and those for which there is a chance we may not return once we leave. These types of states are called recurrent and transient respectively. Formally, if f_i represents the probability that, starting in state i, our Markov chain will ever re-enter state i, then i is a recurrent state if $f_i = 1$ and transient if $f_i < 1$.

Recurrent states are just that: states which re-occur over and over again. Transient states on the other hand are states which occur only a finite number of times over the horizon of the Markov chain. Eventually, our Markov chain no longer transitions back to any of the transient states. It is then of interest to calculate the expected number of times a transient state will be visited before our chain is “absorbed” into some recurrent state or states.

Expected visits to Transient States

To calculate the expected number of visits to transient states over the horizon of our Markov chain, we perform some basic matrix calculations. First we define a matrix

$\boldsymbol{P_T}$ which is the matrix of transition probabilities between transient states, formed by removing any recurrent states from our matrix $\boldsymbol{P}$. Let $T = \{s_i\}$ such that s_i is transient.

Next, we define the matrix $\boldsymbol{S}$ to be the matrix of expected visits to each state, i.e. $\boldsymbol{S}_{ij} = s_{ij}$ represents the total number of times a Markov chain currently in i will visit state j. If we condition on the first transition from i to some state k, then the expected number of visits to state j is

(10) $$s_{ij} = \delta_{ij} + \sum_{k \in T} P_{ik} s_{kj} \text{ where } \delta_{ij} = \begin{cases} 1 \; if \; i = j \\ 0 \; if \; i \neq j \end{cases}$$

This translates into the following matrix equation:

(11) $$\boldsymbol{S} = \boldsymbol{I} + \boldsymbol{P_T S}$$

Solving for $\boldsymbol{S}$ yields:

(12) $$\boldsymbol{S} = (\boldsymbol{I} - \boldsymbol{P_T})^{-1}$$

We now have the matrix $\boldsymbol{S}$ whose elements s_{ij} comprise the expected number of visits to a transient state j given that we start in state i. For example, the sum of the first row of this matrix would be the total number of visits to transient states over the horizon of a Markov chain given that our chain starts in state s_1.

A Numerical Example

As an example, consider a Markov chain which undergoes transitions between four states. Let the following transition probability matrix represent the Markov chain:

$$P = \begin{array}{c|cccc} state & 1 & 2 & 3 & 4 \\ \hline 1 & 1/4 & 1/4 & 3/8 & 1/4 \\ 2 & 1/2 & 0 & 1/2 & 0 \\ 3 & 2/5 & 2/5 & 1/10 & 1/10 \\ 4 & 0 & 0 & 0 & 1 \end{array}$$

Here, state 4 is what is referred to as an absorbing state; once our process enters state four, it remains there. Thus state four is a recurrent state. The other states are all transient states. To calculate the number of visits to the transient states, we need to eliminate the recurrent states from our transition probability matrix, resulting in the modified matrix of transient state probabilities:

$$P_T = \begin{array}{c|ccc} state & 1 & 2 & 3 \\ \hline 1 & 1/4 & 1/4 & 3/8 \\ 2 & 1/2 & 0 & 1/2 \\ 3 & 2/5 & 2/5 & 1/10 \end{array}$$

We now calculate the matrix of visits to transient states, ***S***, by subtracting this matrix from the identity matrix and inverting the result:

$$\boldsymbol{S} = \begin{bmatrix} 1 - \frac{1}{4} & -\frac{1}{4} & -\frac{3}{8} \\ -\frac{1}{2} & 1 - 0 & -\frac{1}{2} \\ -\frac{2}{5} & -\frac{2}{5} & 1 - \frac{1}{10} \end{bmatrix}^{-1} = \begin{bmatrix} 8 & \frac{66}{13} & \frac{80}{13} \\ 8 & \frac{84}{13} & \frac{90}{13} \\ 8 & \frac{76}{13} & \frac{100}{13} \end{bmatrix}$$

Reading across the top row has the interpretation that if our process starts in state one, we would expect to be in state one 8 times, state two 66/13 times, and state three 80/13 times, before being absorbed into state four.

Applications of Markov Chains to CUSUM Control Charts

The control chart we examine is a CUSUM chart, and as such the distribution of the next CUSUM statistic depends only upon the current statistic. Hence we can model our control chart as a Markov chain. As every cycle of our CUSUM chart ends with a true signal, the state which represents a true signal is a recurrent state; once we have received a true signal, we never return to any of the previous states during the same cycle. Hence all the other stages of the CUSUM chart are represented as transient states.

Thus, the calculation of the expected visits to transient states plays a critical role in our LRHC calculations. Note that this calculation tells us how many false signals occur, and also plays a critical role, along with our sampling intervals and sample sizes, in calculating the expected time out-of-control, expected number of samples, and expected time the process is operating. Further details on these calculations are located in Chapter 4 and Appendix A.

CHAPTER 4: A MARKOV MODEL OF CUSUM CONTROL CHART ECONOMIC PERFORMANCE

Model Assumptions and Justification

In order to model our CUSUM chart using the Markov techniques discussed in the previous chapter, we make three assumptions about our process:

A1. When our process is in-control, our samples come from a $N(\mu, \sigma)$ distribution where μ, σ are known

A2. When our process is out-of-control, our samples come from a $N(\mu + \delta\sigma, \sigma)$ distribution where μ, σ, δ are known

A3. The amount of time, T, until our process shifts out-of-control is a random variable with $Exp(\text{rate} = \lambda)$ distribution where λ is known

A1 and A2 are assumptions which allow us to take advantage of the CDF of the standard normal distribution. A3 is assumed because of the memoryless property of the exponential distribution. The memoryless property of the exponential distribution allows us to say that the chance our process will shift out-of-control over the next time interval given that it has yet to shift out-of-control is the same as the probability that it would shift out-of-control over that interval if the process had just started. This assumption allows us calculate the probability of a control status change independently of the elapsed time.

Finally, even though our CUSUM statistics will come from continuous distributions, in order to take advantage of the Markov techniques outlined in Chapter 3 we must model the CUSUM as proceeding in discrete steps. The following section outlines how this is done using the chart parameters under the control of the designer.

The CUSUM Chart and Associated Design Parameters

Our CUSUM chart will make use of nine design parameters which are under the control of the designer:

b	The alarm boundary
Δs	The step size by which our CUSUM statistic may increase or decrease. This value must divide b to an integer.
a	The reference value to be subtracted from our CUSUM statistic
n_{min}	The minimum sample size
n_{max}	The maximum sample size
α_1	The shape parameter which determines the rate at which we move from our minimum sample size to our maximum sample size
h_{min}	The minimum sampling interval
h_{max}	The maximum sampling interval
α_2	The shape parameter which determines the rate at which we move from our maximum sampling interval to our minimum sampling interval

Our CUSUM statistic will be the following:

(13) $$C_k = \begin{cases} 0 \text{ for } k = 0 \\ \max\left\{0, \left(\Delta s \left\lfloor \frac{C_{k-1} + |z_k| - a}{\Delta s} \right\rfloor \right)\right\} \text{ for } k \geq 1 \end{cases}$$

Here z_k is the standardized sample mean of the k^{th} sample and $\lfloor * \rfloor$ is the floor function. In this way, our CUSUM will proceed in discrete steps of size Δs, always rounded down to the nearest integer multiple of Δs. A signal will be generated if $C_k \geq b$.

VSSI Using Maps to break the 'Curse of Dimensionality'

As we have seen in previous work, allowing the sample size and interval to vary as the plotted statistic becomes more extreme is economically advantageous. Note that by using the parameters outlined above, our CUSUM breaks up into $m = \frac{b}{\Delta s} + 1$ discrete levels. If we attempted let each of these levels have its own, independently controllable, associated sample size and sampling interval, we would quickly run into issues of computational complexity.

For instance consider a chart with alarm boundary $b = 3$ and step size $\Delta s = 0.01$. We then have 301 individual states, each with their own associated, independently controllable sample and sampling interval, for a total of 602 parameters which must be optimized just for those two facets (sample size and interval) alone. This so called 'curse of dimensionality' quickly causes our design algorithm to become computationally complex, begging for a simplification.

As a solution to the 'curse of dimensionality' we propose two maps from the extremity of the most recent statistic: one to a sample size and another to a sampling interval. As an extension, we allow non-linear maps by the addition of two rate parameters,α_1, α_2, one for each map. Consider the following possible mappings.

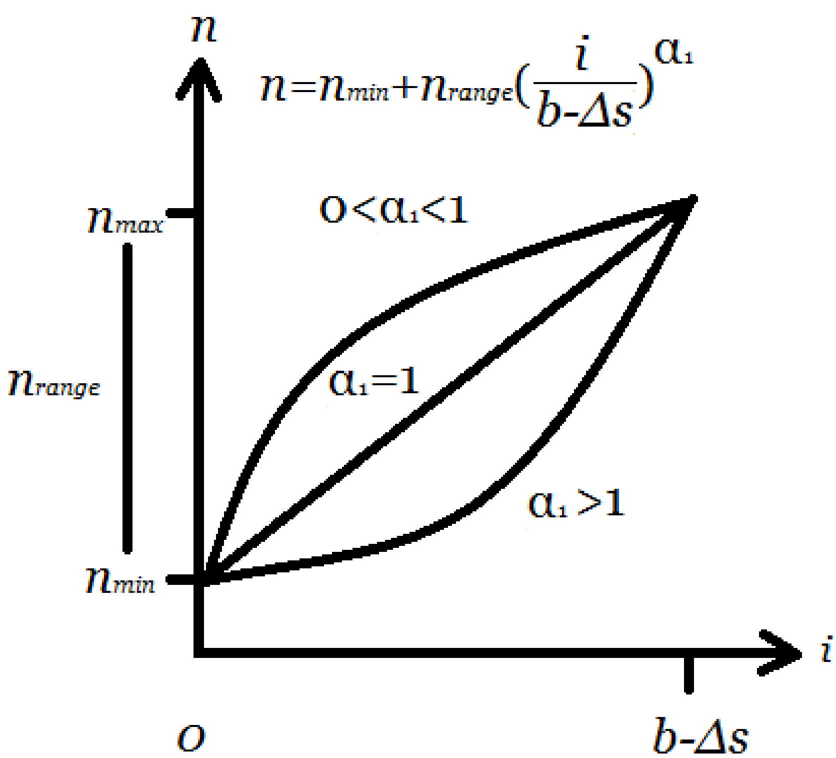

Fig. 3 By altering n_{max}, n_{min} and α_1 we can generate an infinite number of mappings from our CUSUM sum level i to a sample size.

While the above figure illustrates a continuous map for sample size, we of course must introduce integer rounding as sample sizes may only be integers. However, by controlling only three parameters, we are able to create in infinite number of possible maps which satisfy the need for greater sample sizes as we approach the alarm boundary. Similarly consider the following possible mappings for sampling interval.

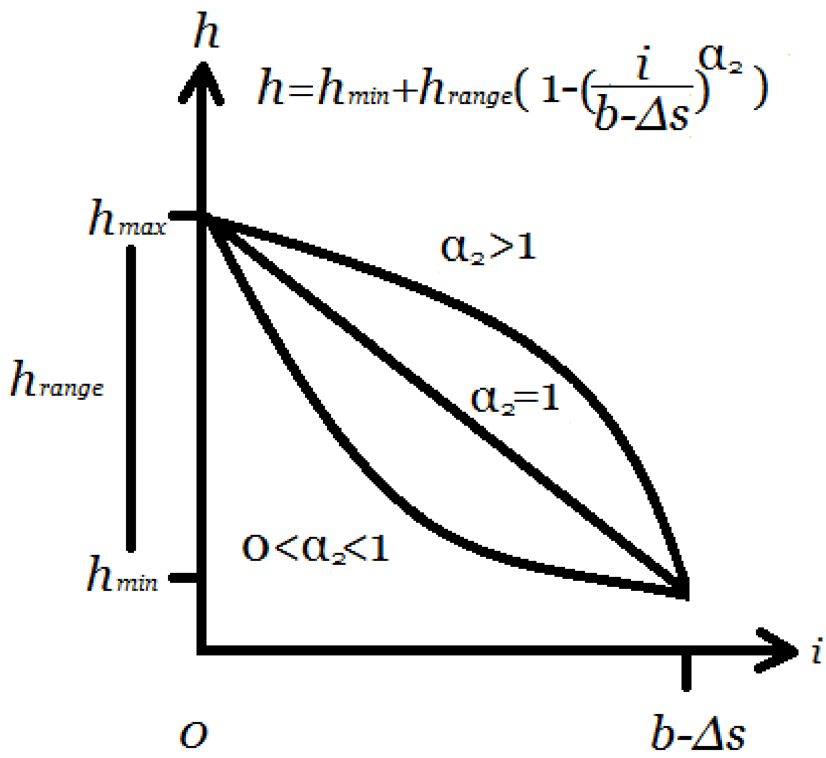

Fig. 4 By altering h_{max}, h_{min} and α_{21} we can generate an infinite number of mappings from our CUSUM sum level, i, to a sampling interval.

Again using only three parameters we are able to create an infinite number of possible mappings for sampling intervals which satisfy the need for shorter sampling intervals with more extreme plotted statistics. Here, time is continuous, as is our mapping; however in practice this continuity will be limited by the precision desired and available to the process engineer.

Modeling the CUSUM as a Markov Chain

Note that by using the parameters outlined above our CUSUM breaks up into $m = \frac{b}{\Delta s} + 1$ discrete levels. Consider also that we have two control states, either in-control or out of-control. Thus we will model our CUSUM control chart as a two dimensional Markov chain with $2m$ discrete states, (i, j), where $i \in \{0, \Delta s, 2\Delta s, \ldots, b -$

$\Delta s, b\}$ represents the sum-level and $j \in \{I, N\}$ represents the control status with I indicating "in-control" N indicating "not in-control/out-of-control." Also note that the states (b, I) and (b, N) correspond to "False Signal" and "True Signal," respectively.

We represent out transition probability matrix,$\boldsymbol{P}$, generically as follows:

state	(0,I)	(Δs,I)	(2Δs,I)	…	(b-Δs,I)	FALΔsE	(0,N)	(Δs,N)	(2Δs,N)	…	(b-Δs,N)	TRUE
(0,I)	$p_{(0,I),(0,I)}$	$p_{(0,I)(\Delta s,I)}$	$p_{(0,I),(2\Delta s,I)}$	…	$p_{(0,I),(b-\Delta s,I)}$	$p_{(0,I),(b,I)}$	$p_{(0,I),(0,N)}$	$p_{(0,I)(\Delta s,N)}$	$p_{(0,I),(2\Delta s,N)}$	…	$p_{(0,I),(b-\Delta s,N)}$	$p_{(0,I),(b,N)}$
(Δs,I)	$p_{(\Delta s,I),(0,I)}$	$p_{(\Delta s,I)(\Delta s,I)}$	$p_{(\Delta s,I),(2\Delta s,I)}$	…	$p_{(\Delta s,I),(b-\Delta s,I)}$	$p_{(\Delta s,I),(b,I)}$	$p_{(\Delta s,I),(0,N)}$	$p_{(\Delta s,I)(\Delta s,N)}$	$p_{(\Delta s,I),(2\Delta s,N)}$	…	$p_{(\Delta s,I),(b-\Delta s,N)}$	$p_{(\Delta s,I),(b,N)}$
(2Δs,I)	$p_{(2\Delta s,I),(0,I)}$	$p_{(0,I)(\Delta s,I)}$	$p_{(\Delta s,I),(2\Delta s,I)}$	…	$p_{(\Delta s,I),(b-\Delta s,I)}$	$p_{(\Delta s,I),(b,I)}$	$p_{(2\Delta s,I),(0,N)}$	$p_{(0,I)(\Delta s,N)}$	$p_{(\Delta s,I),(2\Delta s,N)}$	…	$p_{(\Delta s,I),(b-\Delta s,N)}$	$p_{(\Delta s,I),(b,N)}$
…	…	…	…	…	…	…	…	…	…	…	…	…
(b-Δs,I)	$p_{(b-\Delta s,I),(0,I)}$	$p_{(b-\Delta s,I)(\Delta s,I)}$	$p_{(b-\Delta s,I),(2\Delta s,I)}$	…	$p_{(b-\Delta s,I),(b-\Delta s,I)}$	$p_{(b-\Delta s,I),(b,I)}$	$p_{(b-\Delta s,I),(0,N)}$	$p_{(b-\Delta s,I)(\Delta s,N)}$	$p_{(b-\Delta s,I),(2\Delta s,N)}$	…	$p_{(b-\Delta s,I),(b-\Delta s,N)}$	$p_{(b-\Delta s,I),(b,N)}$
FALΔsE	1	0	0	…	0	0	0	0	0	…	0	0
(0,N)	0	0	0	…	0	0	$p_{(0,N),(0,N)}$	$p_{(0,N)(\Delta s,N)}$	$p_{(0,N),(2\Delta s,N)}$	…	$p_{(0,N),(b-\Delta s,N)}$	$p_{(0,N),(b,N)}$
(Δs,N)	0	0	0	…	0	0	$p_{(\Delta s,N),(0,N)}$	$p_{(\Delta s,N)(\Delta s,N)}$	$p_{(\Delta s,N),(2\Delta s,N)}$	…	$p_{(\Delta s,N),(b-\Delta s,N)}$	$p_{(\Delta s,N),(b,N)}$
(2Δs,N)	0	0	0	…	0	0	$p_{(2\Delta s,N),(0,N)}$	$p_{(0,N)(\Delta s,N)}$	$p_{(\Delta s,N),(2\Delta s,N)}$	…	$p_{(\Delta s,N),(b-\Delta s,N)}$	$p_{(\Delta s,N),(b,N)}$
…	…	…	…	…	…	…	…	…	…	…	…	…
(b-Δs,N)	0	0	0	…	0	0	$p_{(b-\Delta s,N),(0,N)}$	$p_{(b-\Delta s,N)(\Delta s,N)}$	$p_{(b-\Delta s,N),(2\Delta s,N)}$	…	$p_{(b-\Delta s,N),(b-\Delta s,N)}$	$p_{(b-\Delta s,N),(b,N)}$
TRUE	0	0	0	…	0	0	0	0	0	…	0	1

The transition probability matrix above can be thought of as having 4 distinct quadrants: one in which the transitions are between in-control states, one with transitions from in-control states to out-of-control states, one with transitions between out-of-control states, and one with transitions from out-of-control states to in-control states. However once an assignable cause occurs, our process does not randomly return to in-control; thus all of the transition probabilities in this later quadrant are 0.

Also, once we have reached a true signal, we never re-enter any other state within the same cycle, effectively absorbing our Markov chain in the state “True Signal.” Conversely after a false signal we will automatically restart our sum-level at zero and our process is still in-control. Hence we have two transitions for which the probability is guaranteed to be 1: from (b, N) to (b, N) and from (b, I) to $(0, I)$.

This leaves three types of control transitions as noted above, each with three types of sum-level transitions: to a sum-level of zero, strictly between zero and b, and b or greater, for a total of nine cases which must be considered in order to complete our transition probability matrix. We outline the general formulas for each of these nine cases in the following section.

Formulas and Derivations of Nine Transition Probability Cases

Brief derivations of the formulas for the nine remaining cases of transition probabilities follow. Full derivations can be found in Appendix A.

Case 1: $(i, I) \rightarrow (j, I);\ \ j = 0$

Description: From any in-control sum-level to sum-level zero, remaining in-control.

Formula: $[2(\Phi(a - i + \Delta s)^{+} - 0.5)](e^{-h_i \lambda})$

Here Φ is the standard normal CDF, $(a - i + \Delta s)^+ = \max\{0, a - i + \Delta s\}$, and h_i is the sampling interval associated with the i^{th} sum-level.

Case 1 is calculated as the probability that we remain in-control over the sampling interval h_i multiplied by the probability that the next sum lands in the target area. Our target area is anything less than Δs, since we will round it down to 0 and anything less than 0 will round that up to zero. The chance we remain in-control is the chance that our time to failure occurs after sampling interval. Hence, we need to calculate $\Pr(i + |z| - a < \Delta s) * \Pr(T > h_i)$, given $z \sim N(0,1)$ and $T \sim Exp(rate = \lambda)$. This is $\Pr(T > h_i) * \Pr(|z| < (a - i + \Delta s)+)$ which is $\left(e^{-h_i \lambda}\right) * 2[\Pr(z < (a - i + \Delta s)+) - 0.5]$.

Case 2: $(i, I) \to (j, I);\ 0 < j < b$

Description: In-control state to non-zero in-control state

Formula: $(e^{-h_i \lambda})[2\left(\Phi((q + \Delta s)^+) - \Phi(q^+)\right)]$

Here $q = j + a - i$ and the other notes above still apply.

For Case 2, our target is not as wide. We calculate instead the chance of landing between our target sum-level and the next highest sum-level; since we are staying within control, we again calculate the chance that our failure occurs after our sampling interval. This is represented by $\Pr(k > h_i) * \Pr(j < i + |z| - a < j + \Delta s)$. Rearranging yields $\left(e^{-h_i \lambda}\right) * \Pr\left((j - i + a)^+ < |z| < \left((j - i + a) + \Delta s\right)^+\right)$. This result can be simplified by letting $q = j + a - i$; $\left(e^{-h_i \lambda}\right) * 2[\Pr(q^+ < z < (q + \Delta s)^+)]$. Again, using Φ to represent the standard normal CDF yields the above formula for case 2.

Case 3: $(i, I) \to (j, I); j = b$ a.k.a. "FALSE"

Description: In-control state false signal (cross alarm boundary in-control)

Formula: $\left[2\left(1 - \Phi(b + a - i)\right)\right]\left(e^{-h_i\lambda}\right)$

Here b is the alarm boundary and the other notes above still apply.

Calculating Case 3 requires our sum-level to reach above the boundary while our control status remains in-control. Here we calculate the $\Pr(i + |z| - a > b) * \Pr(k > h_i)$. Rearranging yields $\left(e^{-h_i\lambda}\right)\Pr(|z| > b - i + a)$. Using the complement rule, doubling to account for the absolute value, and using Φ to represent the standard normal CDF gives us the above formula for case 3. Here we do not bring the right hand term up to zero because b is necessarily greater than i and a is positive so $b - i + a$ must be positive.

Case 4: $(i, I) \to (j, N);\ \ j = 0$

Description: In-control state to state zero out-of-control

Formula: $\left[\Phi\left((a - i + \Delta s)^+ - \sqrt{n_i}\delta\right) - \Phi\left(-(a - i + \Delta s)^+ - \sqrt{n_i}\delta\right)\right](1 - e^{-h_i\lambda})$

Here n_i is the sample size associated with sum-level *i*, δ is the standardized mean shift, and all previous notes still apply.

Case 4 represents a shift out-of-control during the sampling interval. This shift will be reflected in the distribution of our standardized sample statistic, i.e., $z \sim N(\sqrt{n_i}\delta, 1)$ instead of $z \sim N(0,1)$. To correct for this, we subtract $\sqrt{n_i}\delta$ off of every standardized statistic. However this means we are no longer comparing points that are symmetric about the mean, and must compute each portion separately; before we could compute once and double the result. The other change is represented by the probability that our exponential time to failure will be less than the sampling interval. Combining these concepts allows us derive the above formula for case 4.

Case 5: $(i, I) \rightarrow (j, N); 0 < \; j < b$

Description: In-control state to non-zero out-of-control state

Formula: $[\left(\Phi\left((q+\Delta s)^{+}-\sqrt{n_i}\delta\right)+\Phi\left(-(q^{+})-\sqrt{n_i}\delta\right)\right)-\left(\Phi\left(q^{+}-\sqrt{n_i}\delta\right)+\Phi\left(-(q+s)^{+}-\sqrt{n_i}\delta\right)\right)](1-e^{-h_i\lambda})$

As in Case 2, we must include the calculation of our mean shift and the probability that our process shifts out-of-control before we begin sampling again. The complication introduced by the shift is compounded by the introduction of a non-zero value for j; simplifying by letting $q = j - i + a$ cleans up the formula a bit yielding the above for case 5.

Case 6: $(i, I) \rightarrow (j, N);\; j = b$ a.k.a. "TRUE"

Description: In-control state to true signal (cross alarm boundary and go out-of-control)

Formula: $\left[\left(1-\Phi\left(b+a-i-\sqrt{n_i}\delta\right)\right)+\left(\Phi\left(-b-a+i-\sqrt{n_i}\delta\right)\right)\right](1-e^{-h_i\lambda})$

Case 6 requires that our plotted statistic fall above the boundary and that the process goes out-of-control. Again recall we must add an amount of $\sqrt{n_i}\delta$ to our standardized score. This ends up subtracting from both sides; our probabilities will be two sided and not symmetric so we must calculate each individually. Finally we multiply by the probability that our process shift occurs before the next sample is taken. Letting Φ represent the standard normal CDF generates the above formula for case 6.

Case 7: $(i,N) \to (j,N);\;\; j = 0$

Description: out-of-control state to state zero out-of-control

Formula: $\left[\Phi\left((a-i+s)^{+}-\sqrt{n_i}\delta\right)-\Phi\left(-(a-i+s)^{+}-\sqrt{n_i}\delta\right)\right]$

Case 8: $(i,N) \to (j,N); 0 < \; j < b$

Description: out-of-control state to non-zero out-of-control state

Formula: $\left(\Phi\left((q+s)^{+}-\sqrt{n_i}\delta\right)+\Phi\left(-(q^{+})-\sqrt{n_i}\delta\right)\right)-\left(\Phi\left(q^{+}-\sqrt{n_i}\delta\right)+\Phi\left(-(q+s)^{+}-\sqrt{n_i}\delta\right)\right)$

Case 9: $(i,N) \to (j,N); j = b$ a.k.a. "TRUE"

Description: out-of-control state to true signal (cross alarm boundary)

Formula: $\left[\left(1-\Phi\left(b+a-i-\sqrt{n_i}\delta\right)\right)+\left(\Phi\left(-b-a+i-\sqrt{n_i}\delta\right)\right)\right]$

Cases 7-9 mimic the changes between states of cases 4-6, however we are remaining out-of-control; once we are out-of-control, the chance that our next sample is out-of-control is 100%, so we do not need to multiply by any factor relating to the control status of the process.

LHRC Calculations Revisited

Since we have our Markov model which can generate the transition probability matrix for any set of the nine given parameters, we can now more explicitly calculate the expected values of $E(N)$, $E(PT)$, $E(F)$, and $E(TOOC)$ as mentioned in Chapter 2 using the useful Markov chain properties discussed in Chapter 3.

Recall that our Markov chain is made up of $2m$ discrete states where $m = \frac{b}{\Delta s} + 1$. Only the final state, 'True Signal,' represented by our two dimensional index (b, N) is recurrent, the remaining states are transient. Removing this state results in our $\boldsymbol{P_T}$ matrix which will be a $(2m-1)\text{x}(2m-1)$ matrix of transient states. We then calculate our matrix of visits to transient states, $\boldsymbol{S}$, as outlined in Chapter 3. Since our process begins at 0 and in-control, $(0, I)$, the first row of this $\boldsymbol{S}$ matrix gives the expected number of visits to every state for a cycle of the chart.

To calculate the $E(F)$ it is enough to check the entry in the $\boldsymbol{S}$ matrix column corresponding to state (b, I), 'False Signal,' which is $\boldsymbol{S}_{1,m}$. This entry is precisely the expected number of visits to the state 'False Signal' for a cycle of the chart.

To calculate $E(PT)$, the expected time the process is in operation, we calculate the sum of the product of the number of visits to the states in which our process is operating with the corresponding sampling interval. The states for which the process is operating correspond to states (i, j) so that $i \in \{0, \Delta s, 2\Delta s, \dots, b - \Delta s\}$ and $j \in \{I, N\}$. Let $\tau_{(i,j)}$ represent the number of visits to the state (i, j). We have

(14) $$E(PT) = \sum_{j \in \{I,N\}} \sum_{i=0}^{b-\Delta s} \tau_{(i,j)} h_i$$

Notice we exclude states where the sum-level (first dimension index, i) is b; these states correspond to signals, which stops our process. The number of visits to these states is in the first row of the $\boldsymbol{S}$ matrix, corresponding to entries $\boldsymbol{S}_{1,1}$ through $\boldsymbol{S}_{1,m-1}$, and $\boldsymbol{S}_{1,m+1}$ through $\boldsymbol{S}_{1,2m-1}$.

Similarly we calculate the total number of samples, $E(N)$, using the same transient states mentioned above. This time we sum the product of those same $\tau_{(i,j)}$ with the sample size corresponding to sum-level i. We have

(15) $$E(N) = \sum_{j\in\{I,N\}} \sum_{i=0}^{b-\Delta s} \tau_{(i,j)} n_i$$

Finally, to calculate expected time out-of-control, $E(TOOC)$, we subtract the expected time until an assignable cause occurs from the expected time the process is in operation. By A3, the time until an assignable cause occurs is an exponentially distributed random variable with rate λ, hence the expected value for this random variable is $\frac{1}{\lambda}$. We have

(16) $$E(TOOC) = E(PT) - \frac{1}{\lambda} = \left(\sum_{j\in\{I,N\}} \sum_{i=0}^{b-\Delta s} \tau_{(i,j)} h_i\right) - \frac{1}{\lambda}$$

Thus LRHC (8) calculation becomes:

(8) $$LRHC = \frac{c_1 E(N) + c_2 E(TOOC) + c_3 E(F) t_1 + c_4}{E(PT) + t_1 E(F) + t_2}$$

$$(17) \qquad LRHC = \frac{c_1\left(\sum_{j\in\{I,N\}}\sum_{i=0}^{b-\Delta s}\tau_{(i,j)}n_i\right)+c_2\left(\left(\sum_{j\in\{I,N\}}\sum_{i=0}^{b-\Delta s}\tau_{(i,j)}h_i\right)-\frac{1}{\lambda}\right)+c_3\tau_{(b,I)}t_1+c_4}{\left(\sum_{j\in\{I,N\}}\sum_{i=0}^{b-\Delta s}\tau_{(i,j)}h_i\right)+t_1\tau_{(b,I)}+t_2}$$

Economic Design by Algorithmic Programming Using our Markov Model

The economic design of our CUSUM control chart is carried out by an appropriately written computer search algorithm. This search algorithm takes the feasible parameters input by the process engineer along with the known process constants, i.e. costs, false signal search and true signal repair time, and rate of assignable causes, along with an initial set of parameter values and uses the Markov model outline previously to calculate the LRHC for those best guess parameters.

Then, allowing the first parameter to vary within its parameter space and holding all others constant, the computer program checks the LRHC of all possible values for this first parameter. Whenever a lesser LRHC is encountered, the ‘best guess’ for this parameter is updated to the value which resulted in the lesser LRHC. Once all values of this first parameter have been search over, the computer program moves on to the next parameter, holding the others constant. This process is repeated until all parameters have been searched over one time.

At this point, the computer program compares the LRHC that it started with from its initial ‘best guess’ parameters and the LRHC of this first pass of updated ‘best guess’ parameters. If the LRHC’s are the same, then the algorithm has reached at least a locally optimal point. If the LRHC’s are different, the algorithm begins again with these updated ‘best guess’ parameters and searches again. This is continued until an optimal set of parameters is found. These parameters, along with the LRHC of the CUSUM control chart are reported to the process engineer, who can then set up and use the control chart to monitor the control status of the process.

An Example of our CUSUM Chart in Operation

It may be conceptually helpful to consider the following example of our CUSUM chart in operation. Consider the following CUSUM chart with alarm boundary b = 1.5, step size s = 0.25, and reference value a = 0.75. Our first plotted statistic will be 0 at time 0.

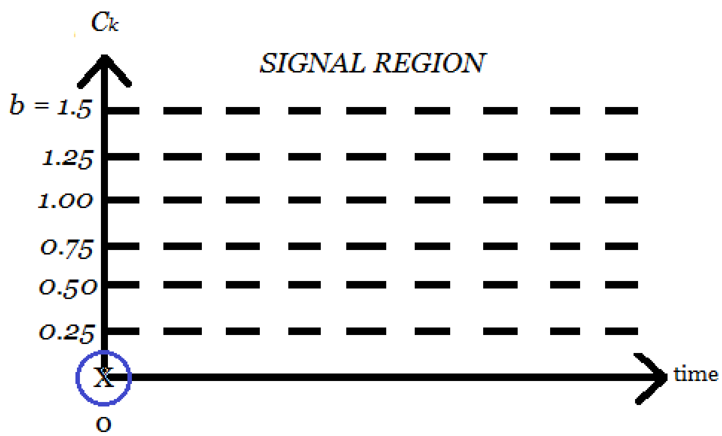

Fig. 5 At time 0, our sum begins at 0

We then wait some amount of time (the sampling interval associated with being in sum-level zero), call it h_0, and then will sample at some specified sample size (associated with being in sum-level zero) n_0. Imagine our standardized sample statistic, z_1, is -0.62. We take the floor of $\left(\frac{0+|-0.62|-0.75}{0.25}\right)$, which is -1, and multiply this by s to get -0.25. Since this is less than 0, we again plot 0.

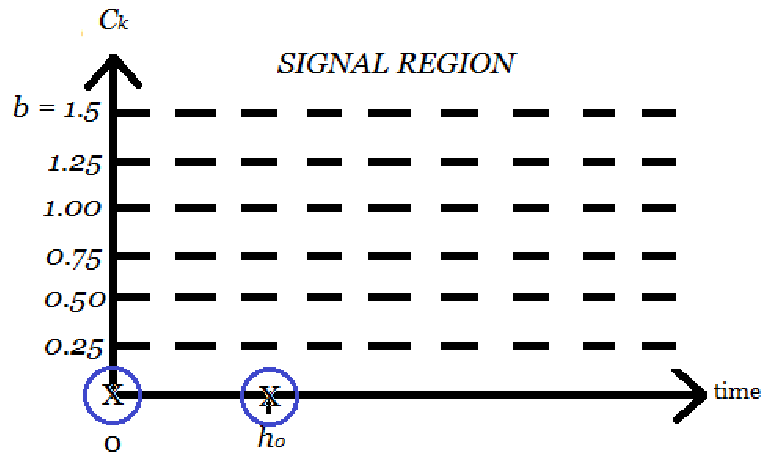

Fig. 6 After waiting some amount of time, we will plot our next CUSUM statistic; in our example, this statistic turns out to be 0 again.

Since we are still in sum-level zero, we again wait h_0 time units before taking our sampling of size n_0. Now, suppose our sample statistic, z_2, is 1.68. We repeat the process above to find that our next plotted statistic is $0.25\left\lfloor\frac{(0+|1.68|-0.75)}{0.25}\right\rfloor$ which is 0.75. We then plot the statistic 0.75 at time $2h_0$.

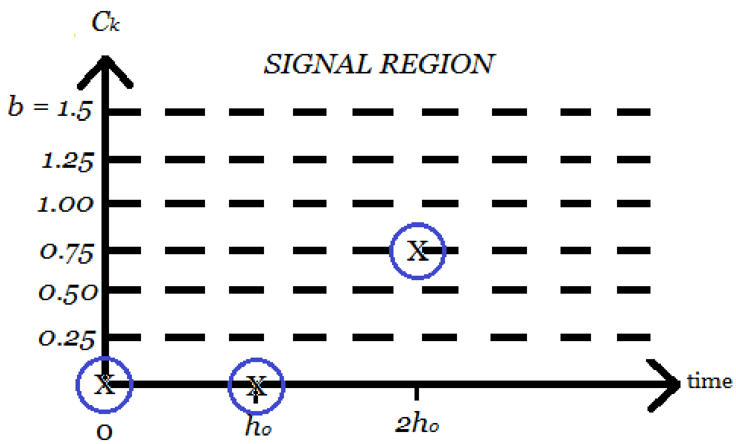

Fig. 7 After waiting the predetermined amount of time, we sample again. Our CUSUM statistic has increased to 0.75. Since we have moved closer to the alarm boundary, we may wish to take a larger sample sooner than if we had remained at zero.

Our sum-level has now increased to 0.75, which is closer to the alarm boundary; we may be concerned that an assignable cause has occurred, but we have not been given a signal to action. Instead, we now change our sampling interval to the predetermined interval associated with sum-level 0.75, call it $h_{0.75}$. After waiting this amount of time, we conduct another sample, this time of size $n_{0.75}$. Imagine now that our standardized statistic, z_3, turns out to be 0.32. Again we plot $0.25\left\lfloor\frac{(0.75+|0.32|-0.75)}{0.25}\right\rfloor$ which is 0.25. We then plot 0.25 at time $2h_0+h_{0.75}$.

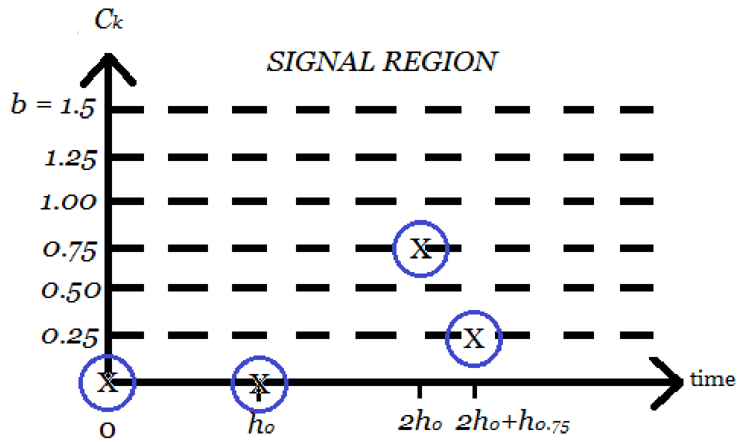

Fig. 8 Having waited the predetermined amount of time due to our last sample, we sample again and find that our CUSUM statistic has fallen back to 0.25. Perhaps the previous sample was just an anomaly.

We proceed in this fashion until our plotted statistic lands at or above the alarm boundary. At that time a signal to action is generated and we investigate for an assignable cause. If no cause is found, the signal was a false signal and the sum is started again at 0. If an assignable cause is found, it is removed and the process is restarted again on a new chart.

CHAPTER 5: COMPARATIVE STUDY AND RESULTS

Preliminary Investigation

Some initial test cases designed by the computer search algorithm using constants in the study by Carolan et al. indicate few important generalities. First, the optimal parameter for Δs is always the smallest possible value. This observation is reasonable because the smaller the value of Δs, the better the approximation of a continuous state space for our Markov Chain. The minimum value of Δs is restricted by the computer processing memory. In our investigation, the minimum feasible value of Δs is 0.005. All economically designed CUSUM charts in our investigation use this value of Δs.

A surprising result of the preliminary investigation is that for the sampling interval, given by the map $h_i = h_{min} + (h_{max} - h_{min})\left(1 - \left(\frac{i}{b-\Delta s}\right)^{\alpha_2}\right)$ the minimum feasible value of α_2 is always optimal. Notice that the smaller α_2, the closer the map becomes to a discrete step function given piecewise as:

(18) $$h_i = \begin{cases} h_{max} & \text{if } i = 0 \\ h_{min} & \text{if } i > 0 \end{cases}$$

This result is unexpected in light of the Carolan et al. study which uncovered benefits utilizing a continuous, linear map from statistic extremity to sampling interval. An explanation of this is that because of the reference value which our CUSUM subtracts off of each statistic, our optimal design will hold our plotted statistic at sum-level zero almost until an assignable cause occurs; thus, any extremity in the plotted statistic is taken as ‘warning’ similar to the 2-stage Xbar chart discussed in Chapter 1.

Because of this, the above piecewise function for sampling interval is implemented in the design algorithm and lesser LRHC costs are discovered as a result. This eliminates the need for the parameter α_2, hence it is not included in the results reported here.

Another preliminary finding is that the minimum sampling interval, h_{min}, is always chosen so for the minimum feasible value. This is expected since the sooner a sample is taken after receiving evidence the process may be out-of-control, the sooner the process engineer will likely receive a True Signal, decreasing the costly time spent out-of-control. Hence all CUSUM control charts economically designed by the search algorithm chart have $h_{min} = 0.05$ hours.

Comparison Scenarios

To gauge the impact of improvements afforded by our CUSUM chart design algorithm, we use the sixteen scenarios found in Carolan et al. for comparison. These scenarios are outlined below in Table 1.

Constants

Scenario	c_1	c_2	c_3	c_4	t_1	t_2	$(1/\lambda)$	δ
1	$ 2.00	$ 500.00	$1,500.00	$1,000.00	2	1	100	0.5
2	$ 5.00	$ 500.00	$1,500.00	$1,000.00	2	1	100	0.5
3	$ 2.00	$ 1,500.00	$1,500.00	$1,000.00	2	1	100	0.5
4	$ 5.00	$ 1,500.00	$1,500.00	$1,000.00	2	1	100	0.5
5	$ 2.00	$ 500.00	$3,000.00	$1,000.00	5	1	100	0.5
6	$ 5.00	$ 500.00	$3,000.00	$1,000.00	5	1	100	0.5
7	$ 2.00	$ 1,500.00	$3,000.00	$1,000.00	5	1	100	0.5
8	$ 5.00	$ 1,500.00	$3,000.00	$1,000.00	5	1	100	0.5
9	$ 2.00	$ 500.00	$1,500.00	$1,000.00	2	1	100	1
10	$ 5.00	$ 500.00	$1,500.00	$1,000.00	2	1	100	1
11	$ 2.00	$ 1,500.00	$1,500.00	$1,000.00	2	1	100	1
12	$ 5.00	$ 1,500.00	$1,500.00	$1,000.00	2	1	100	1
13	$ 2.00	$ 500.00	$3,000.00	$1,000.00	5	1	100	1
14	$ 5.00	$ 500.00	$3,000.00	$1,000.00	5	1	100	1
15	$ 2.00	$ 1,500.00	$3,000.00	$1,000.00	5	1	100	1
16	$ 5.00	$ 1,500.00	$3,000.00	$1,000.00	5	1	100	1

Table 1. 16 comparative scenarios

Key:

c_1	Cost per sample
c_2	Hourly out-of-control cost
c_3	Hourly false signal search costs
c_4	Hourly assignable cause repair costs
t_1	Mean time searching for a false signal
t_2	Mean time repairing a true signal
$(1/\lambda)$	Expected hours until assignable cause occurs
δ	Size process mean shift due to assignable cause

By loading these above scenarios into the computer search algorithm, the following parameters in Table 2 below are found to be optimal for a CUSUM chart as described in Chapter 4.

	Optimized Parameters						Results		
	b	a	h_{max}	n_{min}	n_{max}	α_1	LRHC CUSUSM	LRHC Xbar	Savings
Scenario 1	2.97	1.38	3.65	19	27	0.37	$ 39.95	$ 40.28	0.83%
2	2.40	1.35	6.00	19	23	0.29	$ 56.94	$ 57.13	0.33%
3	3.08	1.37	2.10	19	34	0.71	$ 62.43	$ 62.91	0.77%
4	2.52	1.36	3.35	19	26	0.49	$ 91.83	$ 92.19	0.39%
5	3.64	1.41	3.75	20	31	0.45	$ 40.45	$ 40.58	0.32%
6	3.13	1.39	6.15	20	26	0.26	$ 58.06	$ 57.89	-0.29%
7	3.90	1.39	2.15	20	43	0.86	$ 63.19	$ 63.22	0.05%
8	3.29	1.39	3.45	20	30	0.52	$ 93.33	$ 93.19	-0.15%
9	3.40	1.45	2.10	6	11	0.93	$ 25.27	$ 25.41	0.57%
10	2.95	1.44	3.35	6	9	0.97	$ 33.94	$ 34.28	1.00%
11	3.60	1.43	1.05	5	16	0.99	$ 36.95	$ 37.50	1.48%
12	3.17	1.42	1.65	5	11	0.85	$ 51.79	$ 52.71	1.74%
13	4.31	1.43	1.85	5	14	0.86	$ 25.36	$ 25.76	1.53%
14	3.93	1.41	2.95	5	10	0.73	$ 34.12	$ 34.55	1.24%
15	4.18	1.45	1.05	5	19	0.95	$ 37.27	$ 37.69	1.11%
16	3.88	1.43	1.65	5	13	0.85	$ 52.32	$ 53.11	1.48%

*$\Delta s = 0.005$ for all scenarios; $h_{min} = 0.05$ for all scenarios

Table 2. Results for economically designed CUSUM control chart vs. similar Xbar chart

Key:

b	The standardized score of the alarm boundary
a	The reference value which will be subtracted from our CUSUM statistic
h_{max}	The maximum sampling interval
n_{min}	The minimum sample size
n_{max}	The maximum sample size
α_1	The shape parameter which determines the rate at which we move from our minimum sample size to our maximum sample size

Summary of Results

Our economically designed CUSUM control charts perform better, having lower LRHCs than similarly designed Xbar charts with continuously variable sampling intervals, under all scenarios except scenarios 6 and 8. These two scenarios both have the higher cost per sample ($5).

Savings for the CUSUM chart range from -0.29% to 1.74% with an average savings of 0.78%. Optimal alarm boundaries range from 2.40 to 4.31 standard deviations with an average of 3.4 standard deviations. Reference values were fairly close together, ranging from 1.35 to 1.45 standard deviations with an average of 1.41 standard deviations. Maximum sampling intervals range from 1.05 to 6.15 hours with an average of 2.89 hours. Minimum sample sizes range from 5 to 20 with an average of 12.4. Maximum sample sizes range from 9 to 43 with an average of 21.4. The shape parameter, α_1, for the map from minimum sample size to maximum sample size ranges from 0.26 to 0.99 with an average of 0.69.

REFERNCES

[1] Barnard, G. A. "Control Charts and Stochastic Processes." *Journal of the Royal Statistical Society. Series B (Methodological)* 21.2 (1959): 239-71. Print.

[2] Carolan, Christopher A., J.F. Kros, and S.E. Said. "Economic Design of Xbar Control Charts with Continuously Variable Sampling Intervals." *QUALITY AND RELIABILITY ENGINEERING INTERNATIONAL* 26.3 (2010): 235-45. Print.

[3] Celano, G. et al. “The Economic Performance of CUSUM t Control Charts for Monitoring Short Production Runs.” *Quality Technology and Quantitative Management* 9.4 (2012): 339-354. Print.

[4] Helo, Petri T., and Jianxin Roger Jiao. "Optimization Design of a CUSUM Control Chart Based on Taguchi's Loss Function." *The international journal of advanced manufacturing technology* 35.11/12 (2008): 1234-43. Print.

[5] Jeh-Nan Pan, and Su-Tsu Chen. "The Economic Design of CUSUM Chart for Monitoring Environmental Performance." *Asia Pacific Management Review* 10.2 (2005): 155-161. Print.

[6] Nelson, Lloyd S. *Control Charts*. Wiley, 2005. Print.

[7] Ou Yanjing, et al. “An Adaptive CUSUM Chart with Single Sample Size for Monitoring Process Mean and Variance.” *Quality and Reliability Engineering International* 29.7 (2012): 1027-1039. Print.

[8] Page, E. S. "Continuous Inspection Schemes." *Biometrika* 41.1/2 (1954): 100-15. Print.

[9] Reynolds, Marion R., Raid W. Amin, and Jesse C. Arnold. "CUSUM Charts with Variable Sampling Intervals." *Technometrics* 32.4 (1990): 371-84. Print.

[10] Shu, LJ, and W. Jiang. "A Markov Chain Model for the Adaptive CUSUM Control Chart." *JOURNAL OF QUALITY TECHNOLOGY* 38.2 (2006): 135-47. Print.

[11] Taylor, Howard M. "The Economic Design of Cumulative Sum Control Charts." *Technometrics* 10.3 (1968): 479-88. Print.

[12] Wu, Zhang, et al. "A CUSUM Chart using Absolute Sample Values to Monitor Process Mean and Variance".IEEE , 2009. 414-418. Print.

[13] Wu, Zhang, et al. "Optimization Designs and Performance Comparison of Two CUSUM Schemes for Monitoring Process Shifts in Mean and Variance." *European journal of operational research* 205.1 (2010): 136-50. Print.

APPENDIX A: DERIVATION OF TRANSITION PROBABILITY FORMULAS

1. Case: $(i, I) \rightarrow (j, I),\ j = 0$
Formula: $[2(\Phi((a-i)+\Delta s)^{+} - 0.5)](e^{-h_i \lambda})$
Derivation-*Given:*
$i \in \{0, \Delta s, 2\Delta s, \dots, b - \Delta s\}$
$j = 0$
$i - a < \Delta s$
$z = \dfrac{\bar{x} - \mu}{\sigma/\sqrt{n_i}} \sim N(0,1)$
$h_i \in \left[h_{min}, h_{min} + h_{range}\ \right] s.t.\, h_i = h_{min} + h_{range}\left(1 - \dfrac{i}{b - \Delta s}\right)^{\alpha_h}\ where\ \alpha_h \in (0,1)$
$h_{min}, h_{range} \in (0, \infty)$
$Let\ q = j - i + a$
$P_{(i,I)(0,I)} = \Pr(0 \leq i + |z| - a < 0 + \Delta s) \cap \Pr\left(h_i < T \middle| T \sim Exp(\lambda)\right)$
$^{*}= \Pr(0 \leq |z| < (a - i + \Delta s)^{+})\left(e^{-(h_i \lambda)}\right)$
$= \Pr(0 < |z| < (a - i + \Delta s)^{+})\left(e^{-(h_i \lambda)}\right)$
$= [\Pr(0 < z < (a - i + \Delta s)^{+}) + \Pr(-(a - i + \Delta s)^{+} < z < 0)]\left(e^{-(h_i \lambda)}\right)$
$= 2\left(\Phi(a - i + \Delta s)^{+} - \Phi(0)\right)\left(e^{-(h_i \lambda)}\right)$
$= 2(\Phi(a - i + \Delta s)^{+} - 0.5)\left(e^{-(h_i \lambda)}\right)$
$^{*}(a - i + \Delta s > 0\ since\ i - a < \Delta s \rightarrow 0 < a - i + \Delta s)$

2. Case: $(i,I) \rightarrow (j,I),\ j > 0$
Formula: $[2(\Phi((q+\Delta s)^{+}) - \Phi(q^{+}))](e^{-h_i\lambda})$ where $q = j - i + a$
Derivation- *Given:*
$i \in \{0, \Delta s, 2\Delta s, \dots, b - \Delta s\}$
$j \in \{\Delta s, 2\Delta s, \dots, b - \Delta s\}$
$i - a < j + \Delta s$
$z = \dfrac{\bar{x} - \mu}{\sigma/\sqrt{n_i}} \sim N(0,1)$
$h_i \in [h_{min}, h_{min} + h_{range}]\ s.t.\ h_i = h_{min} + h_{range}\left(1 - \dfrac{i}{b - \Delta s}\right)^{\alpha_h}\ where\ \alpha_h \in (0,1)$
$h_{min}, h_{range} \in (0, \infty)$
$Let\ q = j - i + a$
$P_{(i,I)(j,I)} = \Pr(j \leq i + |z| - a < j + \Delta s) \cap \Pr(h_i < T | T \sim Exp(\lambda))$
$= \Pr(q \leq |z| < (q + \Delta s)^{+})\ (e^{-(h_i\lambda)})$
$= \Pr(q^{+} < |z| < (q + \Delta s)^{+})\ (e^{-(h_i\lambda)})$
$= [\Pr(-(q + \Delta s)^{+} < z < -(q^{+})) + \Pr(q^{+} < z < (q + \Delta s)^{+})](e^{-(h_i\lambda)})$
*$= 2(\Phi(q + \Delta s)^{+} - \Phi(q^{+}))(e^{-(h_i\lambda)})$
*$((q + \Delta s) > 0\ since\ i - a < j + \Delta s \rightarrow 0 < j - i + a + \Delta s, hence\ 0 < (q + \Delta s))$

3. Case: $(i, I) \rightarrow FALSE$
Formula: $\left[2\left(1-\Phi(b+a-i)\right)\right]\left(e^{-h_i\lambda}\right)$ where b is the alarm boundary
Derivation- *Given:*
$i \in \{0, \Delta s, 2\Delta s, \dots, b-\Delta s\}$
$z = \dfrac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(0,1)$
$b+a-i > 0$
$h_i \in \left[h_{min}, h_{min}+h_{range}\ \right]\ s.t.\ h_i = h_{min}+h_{range}\left(1-\dfrac{i}{b-\Delta s}\right)^{\alpha_h}\ where\ \alpha_h \in (0,1)$
$h_{min}, h_{range} \in (0, \infty)$
$Let\ q = j - i + a$
$P_{(i,I)(False)} = \Pr(b \le i + |z| - a) \cap \Pr\left(h_i < T \middle| T \sim Exp(\lambda)\right)$
$= \Pr(b + a - i \le |z|)\left(e^{-(h_i\lambda)}\right)$
$= [\Pr(b+a-i < z) + \Pr(z < -b-a+i)]\left(e^{-(h_i\lambda)}\right)$
$= \left[2\left(1-\Phi(b+a-i)\right)\right]\left(e^{-h_i\lambda}\right)$

4. Case: $(i, I) \rightarrow TRUE$
Formula: $\left[\left(1 - \Phi\left(b + a - i - \sqrt{n_i}\delta\right)\right) + \left(\Phi\left(-b - a + i - \sqrt{n_i}\delta\right)\right)\right]\left(1 - e^{-h_i\lambda}\right)$
Derivation- *Given:*
$i \in \{0, \Delta s, 2\Delta s, \ldots, b - \Delta s\}$
$z^* = \frac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(\sqrt{n_i}\delta, 1),\ z = \frac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(0,1)$
$b + a - i > 0$
$h_i \in \left[h_{min}, h_{min} + h_{range}\ \right] s.t. h_i = h_{min} + h_{range}\left(1 - \frac{i}{b - \Delta s}\right)^{\alpha_h}\ where\ \alpha_h \in (0,1)$
$h_{min}, h_{range} \in (0, \infty)$
$n_i \in \left[n_{min}, n_{min} + n_{range}\ \right] s.t. n_i = round\left(n_{min} + n_{range}\left(1 - \left(1 - \frac{i}{b - \Delta s}\right)^{\alpha_n}\right)\right)$
$where\ \alpha_n \in (0,1)$
$n_{min}, n_{range} \in \mathbb{Z}$
$P_{(i,I)(False)} = \Pr(b \leq i + |z^*| - a) \cap \Pr\left(h_i > T \middle| T \sim Exp(\lambda)\right)$
$= \Pr(b + a - i \leq |z^*|)\left(1 - e^{-(h_i\lambda)}\right)$
$= \left[\Pr\left(b + a - i < z + \sqrt{n_i}\delta\right) + \Pr\left(z + \sqrt{n_i}\delta < -b - a + i\right)\right]\left(1 - e^{-(h_i\lambda)}\right)$
$= \left[\Pr\left(b + a - i - \sqrt{n_i}\delta < z\right) + \Pr\left(z < -b - a + i - \sqrt{n_i}\delta\right)\right]\left(1 - e^{-(h_i\lambda)}\right)$
$= \left[\left(1 - \Phi\left(b + a - i - \sqrt{n_i}\delta\right)\right) + \Phi\left(-b - a + i - \sqrt{n_i}\delta\right)\right]\left(1 - e^{-h_i\lambda}\right)$

5. Case: $(i, I) \to (j, N),\ j = 0$

Formula: $\left[\Phi\left((a - i + \Delta s)^+ - \sqrt{n_i}\delta\right) - \Phi\left(-(a - i + \Delta s)^+ - \sqrt{n_i}\delta\right)\right](1 - e^{-h_i\lambda})$

Derivation- *Given:*

$i \in \{0, \Delta s, 2\Delta s, \dots, b - \Delta s\}$

$j = 0$

$i - a < \Delta s$

$z^* = \frac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(\sqrt{n_i}\delta, 1),\ z = \frac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(0,1)$

$h_i \in \left[h_{min}, h_{min} + h_{range}\right]\ s.t.\, h_i = h_{min} + h_{range}\left(1 - \frac{i}{b - \Delta s}\right)^{\alpha_h}\ where\ \alpha_h \in (0,1)$

$h_{min}, h_{range} \in (0, \infty)$

$n_i \in \left[n_{min}, n_{min} + n_{range}\right]\ s.t.\, n_i = round\left(n_{min} + n_{range}\left(1 - \left(1 - \frac{i}{b - \Delta s}\right)^{\alpha_n}\right)\right)$

$where\ \alpha_n \in (0,1)$

$n_{min}, n_{range} \in \mathbb{Z}$

$P_{(i,I)(j,N)} = \Pr(0 \le i + |z^*| - a < \Delta s) \cap \Pr\left(h_i > T \middle| T \sim Exp(\lambda)\right)$

*$= \Pr(0 \le |z^*| < (a - i + \Delta s)^+)\left(1 - e^{-(h_i\lambda)}\right)$

$= [\Pr(-(a - i + \Delta s)^+ < z^* < (a - i + \Delta s)^+)]\left(1 - e^{-(h_i\lambda)}\right)$

$= [\Pr\left(-(a - i + \Delta s)^+ < z + \sqrt{n_i}\delta < (a - i + \Delta s)^+\right)]\left(1 - e^{-(h_i\lambda)}\right)$

$= [\Pr\left(-(a - i + \Delta s)^+ - \sqrt{n_i}\delta < z < (a - i + \Delta s)^+ - \sqrt{n_i}\delta\right)]\left(1 - e^{-(h_i\lambda)}\right)$

$= [\Phi\left((a - i + \Delta s)^+ - \sqrt{n_i}\delta\right) - \Phi\left(-(a - i + \Delta s)^+ - \sqrt{n_i}\delta\right)]\left(1 - e^{-(h_i\lambda)}\right)$

*$(a - i + \Delta s > 0\ since\ i - a < \Delta s \to 0 < a - i + \Delta s)$

6. Case: $(i, I) \to (j, N),\ j > 0$

Formula: $\begin{pmatrix} \left(\Phi\left((q+\Delta s)^+ - \sqrt{n_i}\delta\right) + \Phi\left(-(q^+) - \sqrt{n_i}\delta\right)\right) \\ -\left(\Phi\left(q^+ - \sqrt{n_i}\delta\right) + \Phi\left(-(q+\Delta s)^+ - \sqrt{n_i}\delta\right)\right) \end{pmatrix} (1 - e^{-h_i\lambda})$

Derivation- *Given:*

$i \in \{0, \Delta s, 2\Delta s, \dots, b - \Delta s\}$
$j \in \{\Delta s, 2\Delta s, \dots, b - \Delta s\}$
$i - a < j + \Delta s$
$z^* = \frac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(\sqrt{n_i}\delta, 1),\ z = \frac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(0,1)$
$h_i \in \left[h_{min}, h_{min} + h_{range}\right]\ s.t.\ h_i = h_{min} + h_{range}\left(1 - \frac{i}{b - \Delta s}\right)^{\alpha_h}\ where\ \alpha_h \in (0,1)$
$h_{min}, h_{range} \in (0, \infty)$
$n_i \in \left[n_{min}, n_{min} + n_{range}\right]\ s.t.\ n_i = round\left(n_{min} + n_{range}\left(1 - \left(1 - \frac{i}{b - \Delta s}\right)^{\alpha_n}\right)\right)$
$where\ \alpha_n \in (0,1)$
$n_{min}, n_{range} \in \mathbb{Z}$
$Let\ q = j - i + a$
$P_{(i,I)(j,N)} = \Pr(j \le i + |z^*| - a < j + \Delta s) \cap \Pr\left(h_i > T \middle| T \sim Exp(\lambda)\right)$
$= \Pr(q^+ \le |z^*| < (q + \Delta s)^+)\left(1 - e^{-(h_i\lambda)}\right)$
$= \Pr(q^+ < |z^*| < (q + \Delta s)^+)\left(1 - e^{-(h_i\lambda)}\right)$
$= [\Pr(-(q+\Delta s)^+ < z^* < -(q^+)) + \Pr(q^+ < z^* < (q+\Delta s)^+)]\left(1 - e^{-(h_i\lambda)}\right)$
$= [\Pr\left(-(q+\Delta s)^+ < z + \sqrt{n_i}\delta < -(q^+)\right) + \Pr(q^+ < z + \sqrt{n_i}\delta < q + \Delta s)]\left(1 - e^{-(h_i\lambda)}\right)$
$= \big[\Pr\left(-(q+\Delta s)^+ - \sqrt{n_i}\delta < z < -(q^+) - \sqrt{n_i}\delta\right)$
$\quad + \Pr\left(q^+ - \sqrt{n_i}\delta < z < (q+\Delta s)^+ - \sqrt{n_i}\delta\right)\big]\left(1 - e^{-(h_i\lambda)}\right)$
*$= \big[\Phi\left((q+\Delta s)^+ - \sqrt{n_i}\delta\right) + \Phi\left(-(q^+) - \sqrt{n_i}\delta\right) - \left(\Phi\left(q^+ - \sqrt{n_i}\delta\right) + \Phi\left(-(q+\Delta s)^+ - \sqrt{n_i}\delta\right)\right)\big]\left(1 - e^{-(h_i\lambda)}\right)$
*$\left((q+\Delta s) > 0\ since\ i - a < j + \Delta s \to 0 < j - i + a + \Delta s, hence\ 0 < (q + \Delta s)\right)$

7. Case: $(i,N) \to (j,N),\ j = 0$

Formula:$\left[\Phi\left((a - i + \Delta s)^+ - \sqrt{n_i}\delta\right) - \Phi\left(-(a - i + \Delta s)^+ - \sqrt{n_i}\delta\right)\right]$

Derivation- *Given:*

$i \in \{0, \Delta s, 2\Delta s, \dots, b - \Delta s\}$

$j = 0$

$i - a < \Delta s$

$z^* = \frac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(\sqrt{n_i}\delta, 1),\ z = \frac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(0,1)$

$h_i \in \left[h_{min}, h_{min} + h_{range}\ \right]\ s.t.\ h_i = h_{min} + h_{range}\left(1 - \dfrac{i}{b - \Delta s}\right)^{\alpha_h}\ where\ \alpha_h \in (0,1)$

$h_{min}, h_{range} \in (0, \infty)$

$n_i \in \left[n_{min}, n_{min} + n_{range}\ \right]\ s.t.\ n_i = round\left(n_{min} + n_{range}\left(1 - \left(1 - \dfrac{i}{b - \Delta s}\right)^{\alpha_n}\right)\right)$

$where\ \alpha_n \in (0,1)$

$n_{min}, n_{range} \in \mathbb{Z}$

$P_{(i,N)(j,N)} = \Pr(0 \le i + |z^*| - a < \Delta s)$

$^* = \Pr(0 \le |z^*| < (a - i + \Delta s)^+)$

$= [\Pr(-(a - i + \Delta s)^+ < z^* < (a - i + \Delta s)^+)]$

$= [\Pr\left(-(a - i + \Delta s)^+ < z + \sqrt{n_i}\delta < (a - i + \Delta s)^+\right)]$

$= [\Pr\left(-(a - i + \Delta s)^+ - \sqrt{n_i}\delta < z < (a - i + \Delta s)^+ - \sqrt{n_i}\delta\right)]$

$= [\Phi\left((a - i + \Delta s)^+ - \sqrt{n_i}\delta\right) - \Phi\left(-(a - i + \Delta s)^+ - \sqrt{n_i}\delta\right)]$

$^*(a - i + \Delta s > 0\ since\ i - a < \Delta s \to 0 < a - i + \Delta s)$

8. Case: $(i, N) \to (j, N),\ j > 0$

Formula: $\left(\Phi\left((q+\Delta s)^{+} - \sqrt{n_i}\delta\right) + \Phi\left(-(q^{+}) - \sqrt{n_i}\delta\right)\right) - \left(\Phi\left(q^{+} - \sqrt{n_i}\delta\right) + \Phi\left(-(q + \Delta s)^{+} - \sqrt{n_i}\delta\right)\right)$

Derivation- *Given:*

$i \in \{0, \Delta s, 2\Delta s, \dots, b - \Delta s\}$

$j \in \{\Delta s, 2\Delta s, \dots, b - \Delta s\}$

$i - a < j + \Delta s$

$z^{*} = \frac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(\sqrt{n_i}\delta, 1),\ z = \frac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(0,1)$

$h_i \in \left[h_{min}, h_{min} + h_{range}\,\right]\ s.t.\ h_i = h_{min} + h_{range}\left(1 - \frac{i}{b - \Delta s}\right)^{\alpha_h}\ where\ \alpha_h \in (0,1)$

$h_{min}, h_{range} \in (0, \infty)$

$n_i \in \left[n_{min}, n_{min} + n_{range}\,\right]\ s.t.\ n_i = round\left(n_{min} + n_{range}\left(1 - \left(1 - \frac{i}{b - \Delta s}\right)^{\alpha_n}\right)\right)$

$where\ \alpha_n \in (0,1)$

$n_{min}, n_{range} \in \mathbb{Z}$

$P_{(i,N)(j,N)} = \Pr(j \le i + |z^{*}| - a < j + \Delta s)$

$= \Pr(q^{+} \le |z^{*}| < (q + \Delta s)^{+})$

$= \Pr(q^{+} < |z^{*}| < (q + \Delta s)^{+})$

$= [\Pr(-(q + \Delta s)^{+} < z^{*} < -(q^{+})) + \Pr(q^{+} < z^{*} < (q + \Delta s)^{+})]$

$= [\Pr\left(-(q + \Delta s)^{+} < z + \sqrt{n_i}\delta < -(q^{+})\right) + \Pr(q^{+} < z + \sqrt{n_i}\delta < q + \Delta s)]$

$= \left[\Pr\left(-(q + \Delta s)^{+} - \sqrt{n_i}\delta < z < -(q^{+}) - \sqrt{n_i}\delta\right)\right.$

$\qquad\qquad \left. + \Pr\left(q^{+} - \sqrt{n_i}\delta < z < (q + \Delta s)^{+} - \sqrt{n_i}\delta\right)\right]$

*$= \left[\Phi\left((q + \Delta s)^{+} - \sqrt{n_i}\delta\right) + \Phi\left(-(q^{+}) - \sqrt{n_i}\delta\right) - \left(\Phi\left(q^{+} - \sqrt{n_i}\delta\right) + \Phi\left(-(q + \Delta s)^{+} - \sqrt{n_i}\delta\right)\right)\right]$

*$\left((q + \Delta s) > 0\ since\ i - a < j + \Delta s \to 0 < j - i + a + \Delta s, hence\ 0 < (q + \Delta s)\right)$

9. Case: $(i, N) \to TRUE$
Formula: $\left[\left(1-\Phi\left(b+a-i-\sqrt{n_i}\delta\right)\right)+\left(\Phi\left(-b-a+i-\sqrt{n_i}\delta\right)\right)\right]$
where b is the alarm boundary
Derivation- *Given:*
$i \in \{0, \Delta s, 2\Delta s, \dots, b-\Delta s\}$
$z^* = \frac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(\sqrt{n_i}\delta, 1),\ z = \frac{\bar{x}-\mu}{\sigma/\sqrt{n_i}} \sim N(0,1)$
$b+a-i>0$
$h_i \in \left[h_{min}, h_{min}+h_{range}\right]\ s.t.\ h_i = h_{min}+h_{range}\left(1-\frac{i}{b-\Delta s}\right)^{\alpha_h}\ where\ \alpha_h \in (0,1)$
$h_{min}, h_{range} \in (0, \infty)$
$n_i \in \left[n_{min}, n_{min}+n_{range}\right]\ s.t.\ n_i = round\left(n_{min}+n_{range}\left(1-\left(1-\frac{i}{b-\Delta s}\right)^{\alpha_n}\right)\right)$
$where\ \alpha_n \in (0,1)$
$n_{min}, n_{range} \in \mathbb{Z}$
$Let\ q = j-i+a$
$P_{(i,N)(TRUE)} = \Pr(b \le i+|z^*|-a)$
$= \Pr(b+a-i \le |z^*|)\left(1-e^{-(h_i\lambda)}\right)$
$= \left[\Pr\left(b+a-i < z+\sqrt{n_i}\delta\right)+\Pr\left(z+\sqrt{n_i}\delta < -b-a+i\right)\right]$
$= \left[\Pr\left(b+a-i-\sqrt{n_i}\delta < z\right)+\Pr\left(z < -b-a+i-\sqrt{n_i}\delta\right)\right]$
$= \left[\left(1-\Phi\left(b+a-i-\sqrt{n_i}\delta\right)\right)+\Phi\left(-b-a+i-\sqrt{n_i}\delta\right)\right]$

APPENDIX B: R/S-PLUS ECONOMIC DESIGN SEARCH ALGORITHM

```
Parameter.Search<-function(x,y,z){

v<-x[1,1:ncol(x)]

#store individual parameters

b<-v[1]
s<-v[2]
a<-v[3]
hmin<-v[4]
hmax<-v[5]
h.alpha<-v[6]
nmin<-v[7]
nmax<-v[8]
n.alpha<-v[9]

#store ranges of parameters

b.values<-y[1,2:(1+y[1,1])]
s.values<-y[2,2:(1+y[2,1])]
a.values<-y[3,2:(1+y[3,1])]
hmin.values<-y[4,2:(1+y[4,1])]
hmax.values<-y[5,2:(1+y[5,1])]
halpha.values<-y[6,2:(1+y[6,1])]
nmin.values<-y[7,2:(1+y[7,1])]
nmax.values<-y[8,2:(1+y[8,1])]
nalpha.values<-y[9,2:(1+y[9,1])]

#store constants

lambda<-z[1]
delta<-z[2]
false.time<-z[3]
true.time<-z[4]
sample.cost<-z[5]
hrly.ooc.cost<-z[6]
hrly.false.cost<-z[7]
hrly.true.cost<-z[8]

starting.values<-
c(b,s,a,hmin,hmax,h.alpha,nmin,nmax,n.alpha,lambda,delta,false.time,true.time,sample.
cost,hrly.ooc.cost,hrly.false.cost,hrly.true.cost)

#########-Begin Long Run Hourly Cost Function-##########
```

```
LRHC<-
function(b,s,a,hmin,hmax,h.alpha,nmin,nmax,n.alpha,lambda,delta,false.time,true.time,s
ample.cost,hrly.ooc.cost,hrly.false.cost,hrly.true.cost){

r<- round(b/s)
m <-(2*(r))+1
P <- matrix(0,m,m)
states <- seq(0,b-s, by=s)
rownames(P) <- c((0:((r)-1))*s,"False",(0:((r)-1))*s)
colnames(P) <- c((0:((r)-1))*s,"False",(0:((r)-1))*s)
hrange<-(hmax-hmin)
nrange<-(nmax-nmin)
hmin.long<-rep(hmin,(length(states)-1))
h <- c(hmax,hmin.long)
n <- nmin + nrange*(states/(b-s))^n.alpha
n <- round(n)
P[1:length(states),1] <- 2*(pnorm(a-states+s)-0.5)*exp(-
h*lambda)*as.integer(states<=a+s)
P[(r+1),1] <- 1
P[1:length(states),r+2] <- (pnorm(a-states-sqrt(n)*delta+s)-pnorm(-a+states-
sqrt(n)*delta-s))*(1-exp(-h*lambda))*as.integer(states<=a+s)
P[(r+2):m,(r+2)] <- (pnorm(a-states-sqrt(n)*delta+s)-pnorm(-a+states-sqrt(n)*delta-
s))*as.integer(states<=a+s)
P[1:length(states),r+1] <- 2*(1-pnorm(b-states+a))*exp(-h*lambda)
for(j in 2:(r))  {
q <- states[j] - states + a
q.sum.s <- pmax(0,q+s)
q <- pmax(0,q)
shift <- sqrt(n)*delta
P[1:length(states),j] <- 2*(pnorm(q.sum.s)-pnorm(q))*exp(-h*lambda)
P[1:length(states),j+r+1] <- (pnorm(q.sum.s-shift)-pnorm(q-shift)-pnorm(-q.sum.s-
shift)+pnorm(-q-shift))*(1-exp(-h*lambda))
P[(r+2):m,j+r+1] <- (pnorm(q.sum.s-shift)-pnorm(q-shift)-pnorm(-q.sum.s-shift)+pnorm(-
q-shift))
}
I<-diag(1,m)
PT<-I-P
T<-solve(PT)
#Finds "identity minus P" and inverts

transitions<-T[1,1:m]
#The first row of the above matrix is the
#number of visits to the transient states

times<-c(h,false.time,h)
```

```
#Represents the time per visit in each state
#(in-control, false, out-of-control) as given by sampling interval

samples<-c(n,0,n)
#Represents the number of samples per state
#(in-control, false, out-of-control) as given by sample size n

visit.lengths<-c(times*transitions,true.time)
#vector of time per visit x visits to each state,
#including time searching for assignable cause

samples.cost.total<-sum(samples*transitions*sample.cost)
#total cost of all samples,
#samples per transition x transitions x cost per sample

cycle.time.total<-sum(visit.lengths)
#Total cycle length
#sum of all state visit lenghts

ooc.time.total<-(sum(visit.lengths[((r)+2):m])+sum(visit.lengths[1:(r)]))-(1/lambda)
#Out-of-control time total

ooc.cost.total<-ooc.time.total*hrly.ooc.cost
#hourly out-of-control costs x time spent out-of-control

false.cost.total<-visit.lengths[(r)+1]*hrly.false.cost
#total costs of searching for phantom cause

cycle.cost.total<-
sum(samples.cost.total,ooc.cost.total,false.cost.total,hrly.true.cost*true.time)
#adds all costs for the cycle

L.R.H.C<-(cycle.cost.total/cycle.time.total)
#total cost per cycle/total hours per cycle

return(L.R.H.C)
}
############-End Long Run Hourly Cost Function-############

startingLRHC<-do.call(LRHC,as.list(starting.values))
referenceLRHC<-startingLRHC

################-Begin Search Algorithm-###################
```

```
for(i in b.values){b.optimum<-
LRHC(i,s,a,hmin,hmax,h.alpha,nmin,nmax,n.alpha,lambda,delta,false.time,true.time,sa
mple.cost,hrly.ooc.cost,hrly.false.cost,hrly.true.cost)
if(b.optimum<=referenceLRHC)
{referenceLRHC<-b.optimum
b<-i}}

for(i in s.values){s.optimum<-
LRHC(b,i,a,hmin,hmax,h.alpha,nmin,nmax,n.alpha,lambda,delta,false.time,true.time,sa
mple.cost,hrly.ooc.cost,hrly.false.cost,hrly.true.cost)
if(s.optimum<=referenceLRHC)
{referenceLRHC<-s.optimum
s<-i}}

for(i in a.values){a.optimum<-
LRHC(b,s,i,hmin,hmax,h.alpha,nmin,nmax,n.alpha,lambda,delta,false.time,true.time,sa
mple.cost,hrly.ooc.cost,hrly.false.cost,hrly.true.cost)
if(a.optimum<=referenceLRHC)
{referenceLRHC<-a.optimum
a<-i}}

for(i in hmin.values){hmin.optimum<-
LRHC(b,s,a,i,hmax,h.alpha,nmin,nmax,n.alpha,lambda,delta,false.time,true.time,sample
.cost,hrly.ooc.cost,hrly.false.cost,hrly.true.cost)
if(hmin.optimum<=referenceLRHC)
{referenceLRHC<-hmin.optimum
hmin<-i}}

for(i in hmax.values){hmax.optimum<-
LRHC(b,s,a,hmin,i,h.alpha,nmin,nmax,n.alpha,lambda,delta,false.time,true.time,sample.
cost,hrly.ooc.cost,hrly.false.cost,hrly.true.cost)
if(hmax.optimum<=referenceLRHC)
{referenceLRHC<-hmax.optimum
hmax<-i}}

for(i in halpha.values){halpha.optimum<-
LRHC(b,s,a,hmin,hmax,i,nmin,nmax,n.alpha,lambda,delta,false.time,true.time,sample.c
ost,hrly.ooc.cost,hrly.false.cost,hrly.true.cost)
if(halpha.optimum<=referenceLRHC)
{referenceLRHC<-halpha.optimum
h.alpha<-i}}

for(i in nmin.values){nmin.optimum<-
LRHC(b,s,a,hmin,hmax,h.alpha,i,nmax,n.alpha,lambda,delta,false.time,true.time,sample
.cost,hrly.ooc.cost,hrly.false.cost,hrly.true.cost)
if(nmin.optimum<=referenceLRHC)
```

```
{referenceLRHC<-nmin.optimum
nmin<-i}}

for(i in nmax.values){nmax.optimum<-
LRHC(b,s,a,hmin,hmax,h.alpha,nmin,i,n.alpha,lambda,delta,false.time,true.time,sample.
cost,hrly.ooc.cost,hrly.false.cost,hrly.true.cost)
if(nmax.optimum<=referenceLRHC)
{referenceLRHC<-nmax.optimum
nmax<-i}}

for(i in nalpha.values){nalpha.optimum<-
LRHC(b,s,a,hmin,hmax,h.alpha,nmin,nmax,i,lambda,delta,false.time,true.time,sample.c
ost,hrly.ooc.cost,hrly.false.cost,hrly.true.cost)
if(nalpha.optimum<=referenceLRHC)
{referenceLRHC<-nalpha.optimum
n.alpha<-i}}

#################-End Search Algorithm-###################

final.values<-c(b,s,a,hmin,hmax,h.alpha,nmin,nmax,n.alpha,referenceLRHC)
initial.values<-c(starting.values[1:9],startingLRHC)
results<-rbind(final.values,initial.values)
colnames(results)<-
c("b","s","a","hmin","hmax","h.alpha","nmin","nmax","n.alpha","LRHC")
rownames(results)<-c("1-Pass Optimal Parameters","Initial Parameters")

return(results)
}
```