Procedures to Analyze and Design Decentralized, Distributed Sensor Networks for Change Detection

by

Matthew Quarisa

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements for
the degree of Master of Applied Science

Queen's University
Kingston, Ontario, Canada
September 2019

Abstract

The emergence of the Internet of Things (IoT) has reinvigorated interest into the distributed quickest change detection problem within wireless sensor networks. Although a number of system designs addressing this problem exist in literature, the vast scale of IoT has highlighted limitations of existing methods when adapted to large networks. The purpose of this thesis is therefore to investigate and propose both system designs and design methodologies that perform well in large networks.

Numerical analysis techniques are developed that allow for accurate threshold design of the Sequential Probability Ratio Test (SPRT) and Cumulative-Sum (CUSUM) procedures with respect to desired error metrics. Tests designed using these procedures are compared with those using Wald's approximation to highlight the impact of sequential test overshoot on test design. These techniques are shown to also provide insight into the operation of these procedures over time.

Leveraging these techniques, two system designs are proposed to solve the distributed quickest change detection problem. With large sensor networks in mind, multiple simultaneous transmissions within the network are permitted and analyzed uniquely by a fusion center (FC), and the problem of limited bandwidth is considered.

These designs use the CUSUM procedure at local sensors to quantize local sensor observations into binary summary reports that are transmitted to the FC, indicating

the outcome of each local sensor's CUSUM in each time slot. Probability mass functions describing the probability of the fusion center receiving one or more reports from local sensors in each time slot are computed using the developed CUSUM analysis techniques. Accurate characterization of the local sensor reporting process allows the fusion center to implement procedures based on Bayesian and minimax formulations to the quickest change detection problem.

It is shown that using the minimax-based system design may perform well when a number of assumptions are satisfied. This design is capable of scaling to large networks and a methodology by which global and local thresholds may be chosen to meet a desired false alarm rate constraint is proposed. It is also shown that the performance of this design can be numerically computed for different choices of system design variables.

Acknowledgments

First off, I would like to thank my supervisor, Dr. Steven Blostein, for his support and guidance over the course of my MASc program. His willingness to make himself available for discussion at all times throughout this entire program fostered a supportive, collaborative environment that motivated me and helped me thrive. I have learned a lot from our time working together and I would not have been able to to complete this thesis without him.

I would like to thank and express my appreciation to all of my friends and fellow graduate students. I would especially like to thank some of my lab mates: Jordan Clarke, Saeed Rezazadeh, Phillip Oni, and Ali Hedayati. Our discussions, coffee and lunch breaks, stories, and walks around campus will be greatly missed.

I would also like to thank my entire family. A special thank you to my parents, Frank and Carol Ann, and my brother, Richard, for your support during this step of my life. Finally, I would like to thank my girlfriend, Justyna Cox. Your thoughtfulness, support and constant encouragement made this thesis possible.

Contents

Abstract **i**

Acknowledgments **iii**

Contents **iv**

List of Tables **vi**

List of Figures **vii**

Acronyms **1**

Chapter 1: Introduction **2**
1.1 Motivation 2
1.2 Existing Literature 5
1.3 Thesis Objectives 8
1.4 Contributions 9
1.5 Organization of Thesis 10

Chapter 2: Background **12**
2.1 Introduction 12
2.2 Sequential Hypothesis Testing 12
2.3 Sequential Change Detection 18
2.3.1 Minimax Formulation 18
2.3.2 Bayesian Formulation 21
2.4 Distributed Detection 23
2.5 Detection with Energy Constraints 25

Chapter 3: Numerical Computation Methods of Exact SPRT Performance **27**
3.1 Introduction 27
3.2 Algorithm 28

3.3 Application to SPRT 31
3.3.1 Gaussian Shift in Mean 32
3.3.2 Gaussian Shift in Variance 34
3.3.3 Bernoulli Change in Parameter 39
3.3.4 Change in Discrete Probability Mass Function Test 41
3.4 Application to CUSUM 43
3.5 Applications to Distributed Quickest Change Detection 51
3.5.1 Centralized, Distributed Change Detection 51
3.5.2 Decentralized, Distributed Change Detection 54
3.6 Conclusion 59

Chapter 4: A Bayesian Approach to Quickest Change Detection in Large Sensor Networks under Bandwidth Constraints 61
4.1 Introduction 61
4.2 System Model 63
4.3 Local Sensing Algorithm 65
4.4 Fusion Center Algorithm 66
4.5 Results 73
4.6 Limitations 77
4.7 Conclusion 80

Chapter 5: A Minimax Approach to Quickest Change Detection in Large Sensor Networks under Bandwidth Constraints 81
5.1 Introduction 81
5.2 System Model 83
5.3 Local Sensing Algorithm 85
5.4 Fusion Center Algorithm 86
5.5 System Design Procedure 90
5.6 Results 92
5.7 Conclusion 98

Chapter 6: Conclusions and Future Work 100
6.1 Conclusions 100
6.2 Future Work 103

Bibliography 105

List of Tables

4.1 Values of R_{FA}^{FC} and E_{DD}^{FC} computed for a $N(0,1)$ vs $N(1,1)$ test using fixed FC threshold π^*, M, h_l and L. The change parameter, ρ, is varied to show the difference in false alarm rate and expected detection delay performance caused by different values of ρ. 76

List of Figures

2.1 Detectors implementing the SPRT for a N(0,1) vs N(1,1) test. The false alarm and expected detection delay versus error probability, where $\alpha = \beta$, predicted by Wald for this test is compared to Monte Carlo simulation. 17

3.1 A comparison between numerical and Monte Carlo for a N(0,1) vs N(1,1) SPRT. Desired missed detection probability, β, is set equal to false alarm probability, α. 34

3.2 A comparison between the exact and approximate first order Chi-squared and Gamma distributions with shape parameter $y = \frac{1}{2}$. The approximate functions are defined at $x = 0$. 38

3.3 A comparison between numerical and Monte Carlo results for a N(0,1) vs N(0,3) Gaussian SPRT. Missed detection probability, β, is set equal to false alarm probability, α. 39

3.4 A comparison between numerical and Monte Carlo results for a p=.04 vs p=.15 Bernoulli SPRT. Missed detection probability, β, is set equal to false alarm probability, α. 41

3.5 A comparison between numerical and Monte Carlo results for the change in pmf problem. Missed detection probability, β, is set equal to false alarm probability, α. 42

3.6 Probability of accepting H_1 in a $N(0,1)$ vs $N(1,1)$ test with H_1 true and $h = \ln(25)$. 10^6 Monte Carlo trials are distributed over 38 delay stages to compute the probabilities. 45

3.7 Probability of accepting H_1 in a $N(0,1)$ vs $N(1,1)$ test with H_0 true and $h = \ln(25)$. 10^6 Monte Carlo trials are used over 55 delay stages in simulating these probabilities. 46

3.8 A comparison of ARL computed using the proposed method, Monte Carlo, and from (2.31) and (2.32) for a shift in mean N(0,1) vs N(1,1) CUSUM test. 49

3.9 A comparison of ARL between the proposed method, Monte Carlo and from (2.31) and (2.32) for a shift in variance N(0,1) vs N(0,3) CUSUM test. 49

3.10 A comparison of ARL between the proposed method, Monte Carlo, and from (2.31) and (2.32) for a Bernoulli p_1=0.005 vs p_0=0.2 CUSUM test. 50

3.11 A comparison of ARL between the proposed method, Monte Carlo, and from (2.31) and (2.32) for a discrete pmf where $p_{4|H_1}(x) = 0.587\delta(x) + 0.325\delta(x-1) + 0.078\delta(x-2) + 0.011\delta(x-3)$ and $p_{4|H_0}(x) = 0.977\delta(x) + 0.023\delta(x-1) + 2.215\text{x}10^{-4}\delta(x-2) + 1.210\text{x}10^{-6}\delta(x-3)$. 50

3.12 A comparison of ARL to detection between the proposed method and Monte Carlo for a centralized CUSUM with $M = 2$ under varying detection thresholds. 52

3.13 A comparison of ARL to false alarm between the proposed method and Monte Carlo for a centralized CUSUM with $M = 3$ under varying detection thresholds. 53

3.14 Trade-off curve for expected detection delay versus false alarm rate in a N(0,1) vs N(0.5,1) test. 54

3.15 A comparison between numerically computed and Monte Carlo generated $P(r_{k+1} = 1|H_{1,k+1})$, $P(r_{k+1} = 2|H_{1,k+1})$, $P(r_{k+1} \geq 3|H_{1,k+1})$ and $P(r_{k+1} = 1|H_{0,k+1})$ for a $N(0, 1)$ vs $N(0.5, 1)$ test with $h_l = \ln(15)$, $M = 13$, and $L = 3$. 59

4.1 Performance of the proposed distributed system observing a $N(0, 1)$ vs $N(0.5, 1)$ test with $h_l = \ln(10)$, $\rho = 0.001$, $L = 3$ and 10^5 Monte Carlo trials. The number of systems in the system, M, is varied between $M = 3$ to $M = 15$. 74

4.2 Performance of the proposed distributed system observing a $N(0, 1)$ vs $N(1, 1)$ test with $h_l = \ln(10)$, $\rho = 0.001$, $M = 11$ and 10^5 Monte Carlo trials. The bandwidth constraint, L, is varied from $L = 1$ to $L = 3$. . 75

4.3 Fusion Center false alarm rate as a function of π^* for fixed local sensor threshold h_l. 78

5.1 Numerically computed pdfs given H_1 and H_0 characterizing the arrival of reports for a $N(0, 1)$ vs $N(1, 1)$ test with $h_l = \ln(25)$ and $M = 5$. A bandwidth constraint of $L = 5$ and $L = 1$ is used. 88

5.2 Expected detection delay and false alarm rate as a function of FC threshold h_f for a $N(0, 1)$ vs $N(0.5, 1)$ test with $M = 15$, $h_l = \ln(125)$, $L = 1$. 95

5.3 Expected detection delay and false alarm rate as a function of FC threshold h_f for a $N(0,1)$ vs $N(0.5,1)$ with $M = 7$, $h_l = \ln(60)$, $L = 3$. 95

5.4 Expected detection delay and false alarm rate as a function of FC threshold h_f for a $N(0,1)$ vs $N(0.5,1)$ with $M = 15$, $h_l = \ln(60)$, $L = 3$. 96

5.5 Performance of the decentralized CUSUM system observing a $N(0,1)$ vs $N(0.5,1)$ test with $h_l = \ln(15)$ and $M = 13$ under varying bandwidth constraint L. 97

5.6 Performance of the decentralized CUSUM system observing a $N(0,1)$ vs $N(0.5,1)$ test with $h_l = \ln(15)$ with varying number of sensors, M 98

Acronyms

ARL Average Run Length

CUSUM Cumulative-Sum

FA False Alarm

FC Fusion Center

IID Independent and Identically Distributed

IoT Internet of Things

PMF Probability Mass Function

PDF Probability Density Function

SPRT Sequential Probability Ratio Test

Chapter 1

Introduction

1.1 Motivation

Continuous advancement in sensor technology, especially in recent years, has resulted in an abundance of small, cheap, low power sensors on the market. Improvements in these areas has made it feasible to deploy distributed decision making systems in a wide range of industries [28]. These kind of deployments allow for decisions to be made on current events observed by the system to be based on more, and often increasingly diverse, information. This can directly result in increased efficiency, robustness and accuracy in many of these systems [20]. The Internet of Things (IoT), for example, is one area that seeks to establish itself in everyday life by taking advantage of these market trends.

The emergence of IoT has, and will continue to disrupt how businesses, governments and consumers interact with the world. To put this in perspective, it is estimated that by 2021 almost $5 trillion will have been spent on IoT in industries such as manufacturing, healthcare, security and water management [29] [30]. As the prevalence of IoT rises and smart objects within our daily routine continue to appear,

there will be new potential in taking advantage of the vast amounts of data that can be collected about our surrounding environments to make informed preventive or reactionary decisions in response to current events.

In order to realize the potential brought about by distributed decision making systems, renewed focus has been put on the field of sequential detection. The area of sequential detection generally seeks to make statistic inference about an underlying process through measurements taken sequentially in time [16]. At each time slot, a decision maker either stops and declares a decision on the process observed, or continues to observe measurements. Two main types of sequential detection problems are hypothesis testing and change detection.

Sequential hypothesis testing problems seek to distinguish the correct hypotheses from a set of observed measurements. For example, in the area of quality control, we may want to determine whether a batch of items meets a minimum level of quality guaranteed by the manufacturer. To do this, a characteristic of the item is tested and measured. We can then design a statistical test that uses these measurements to determine whether the batch of items meet or fail the guaranteed minimum level of quality. In these types of problems, there is an acceptable probability of misclassification specified by the test designer. The more strict the test is for classifying observations correctly, the longer it will take for the test to reach a decision.

Sequential change detection problems seek to identify a change between events from a set of measurements. In these problems, measurements are initially generated according to a specified probability distribution, corresponding to a specific event. At an unknown time, a change in the events occurs causing the statistical profile of the measurements to change. This change results in measurements being generated

from a different specified probability distribution. Similar to hypothesis testing, tests designed to determine when this change in events occur also have some acceptable error rate that is inversely proportional to how many observations, on average, these tests take before reaching a decision.

Within the context of distributed decision making, both hypothesis testing and change detection problems have been generalized for distributed sensor networks. In such problems, there are a number of devices that observe events simultaneously. Each device may be able to make a final decision itself, or it may forward results from locally run statistical tests based only on its own observations to a global decision maker, called a Fusion Center (FC), to make a final decision.

A specific use-case of sequential detection procedures within distributed, decentralized systems is in monitoring and data collection systems for critical infrastructure such as power grids, oil and gas networks, and roadways [16]. In these applications, a large geographic area may need to be observed, thereby requiring data from a large number of sensors at different locations. It is often not possible to have each sensor forward its observations to the FC to make a decision. This could be due to bandwidth restrictions, cost, or reliability, that limits the flow of information and requires aspects of the decision making process to be decentralized [37].

There is also significant potential for the use of these procedures within IoT in a healthcare setting [1] [19]. Distributed sensing systems can facilitate remote health monitoring and allow for better management of fitness programs, chronic diseases and elderly care. Various medical and non-medical devices, such as sensors installed within the home, disease management, and imaging equipment are being proposed to be used as smart devices to provide clinicians with valuable information that assist

in making insightful decisions about a patient's care. The integration of IoT within healthcare is expected to reduce medical costs, increase quality of life and enrich the user's experience [14]. In order to fully realize this potential, sequential detection procedures will be needed to process and make inferences that relate this data to physical events.

As evident from the use-cases discussed above, future IoT systems will involve the use of many sensors. Current approaches to sequential detection in distributed systems do not scale well to systems with many sensors because they neglect overshoot and design complexity in these tests. Motivated by this, extensions to existing distributed sequential detection procedures are needed to overcome these limitations.

1.2 Existing Literature

Historically, research into the distributed detection problem has often treated local sensors as extensions of the FC. In such systems, detection procedures are implemented globally. Local sensors immediately forward observations they receive to the FC for processing. Early examples of these systems applied to the distributed hypothesis testing problems can be seen in [37] and [38]. Similar formulations were later extended to the distributed quickest change detection problem, such as in [8] and [40].

Motivated by hardware advances in recent years that have highlighted the potential in applying these processing procedures to large distributed systems, investigation into decentralized versions of these system designs has become popular. The design choice of shifting processing to local sensors is, in part, due to constraints on the bandwidth needed for centralized solutions to operate. Wireless spectrum is currently at a premium, and therefore solutions to these problems that are efficient in their use of

spectrum are valued.

A decentralized solution to the distributed quickest change detection problem was considered in [26]. The CUSUM procedure is used at local sensors to notify the FC when a change is thought to have occurred. In the formulation considered, each local sensor may witness the change in distribution at different times. A one-shot scheme is used at the FC. This means that the FC concludes that a change has occurred after the first local sensor decision is made. This kind of decision making is also known as minimal combining. It is shown that this scheme performs well as the mean time between false alarms tends to infinity, but its performance leaves room for improvement at more modest false alarm rates. In these scenarios, it may be beneficial to provide the FC with feedback as to the state of events at local sensors instead of completely removing all processing from the FC. This work also investigates expressions and relations that describe potential choices of detection thresholds at local sensors, however explicit procedures that one can follow to design sensors that satisfy maximum system error rates are not considered. Such procedures would be useful in practice for system designers.

The distributed quickest change detection problem is also considered in [2]. Here, the change in distribution is assumed to be witnessed across all local sensors simultaneously. Two system designs that utilize the CUSUM procedure at local sensors are analyzed. In the first, each local sensor may communicate with the FC only once. When a local sensor test terminates, it is not reinitialized. The FC concludes a change has occurred when m out of a total of K local sensors have claimed a change has occurred. The second design considered has the FC make a decision when m out of K sensors agree simultaneously that a change has occurred. This happens when

m local sensor CUSUMs are operating above their detection threshold. Asymptotic analysis are performed on each design showing under which circumstances and at which values of m these designs may perform well. A limitation of this work is that given a set of sensor thresholds, a method of computing the associated FC false alarm rate is not provided. This makes determination of an appropriate set of thresholds to meet a desired false alarm rate difficult in practice.

In [45], the distributed hypothesis testing problem is investigated. In the system considered, local sensors process observations using local log likelihood ratio statistics, similar to the SPRT, with a set of predefined constants as thresholds. A single bit is transmitted to the FC when the local statistic crosses above or below the set of thresholds.The FC also implements a SPRT on the asynchronous incoming bits. This work considers the potential of using additional bits to encode the amount of sequential test overshoot that occurs at local sensors and shows how performance improves as the number of bits increases. Offline simulation is discussed as a potential method of choosing suitable thresholds for sensors when local sensors and the FC implement SPRT-like procedures.

In each of these works, there are concerns in the ability of these systems to scale to large networks. One reason for this is that, with the exception of [45], the occurrence of sequential test overshoot at local sensors is largely ignored. In networks with many local sensors, the accumulated impact of overshoot can be significant. Some system designs that account for local sensor overshoot are proposed in [47] and [35], although these works analyze the distributed hypothesis testing problem. In [47], overshoot is encoded into a time delay between the decision time of a local sensor and its report transmission time. This time delay is subsequently decoded by the fusion

center to determine the amount of overshoot. Despite accounting for overshoot with this scheme, it adds unnecessary delay on system performance. In [35], quantization strategies used by local sensors to summarize their observations are examined. Such strategies may have potential for capturing overshoot in summary messages, but complicated quantization schemes incur more overhead in local sensor messages. Given that applications of these systems may involve low power sensors, such techniques may be difficult to implement in certain circumstances. It is also unclear what kind of quantization scheme should be used for the large sensor network situation under consideration.

Another limitation of existing literature is that design procedures that can be followed in practice to determine suitable sensor thresholds are not provided. Given a maximum error constraint, it is desirable to be able to specify thresholds that can design a system to meet this constraint within an acceptable tolerance. Over and under designing sensors is especially troubling in large networks, where the cumulative impact of this on performance may be significant.

1.3 Thesis Objectives

This thesis focuses on the quickest change detection problem within distributed, decentralized systems. Implementing sequential change detection procedures in such systems involve the design of statistical tests at local sensors, which directly observe measurements, and of the FC, charged with making a final decision. The objective of our research is to provide a methodology by which we can design local sensors and the FC and be capable of analyzing the performance of these systems. In doing this, we investigate system designs based on Bayesian and minimax problem formulations,

and discuss limitations of using the former in the design of large sensor networks.

1.4 Contributions

The main contributions of this thesis are as follows.

1. Numerical techniques are proposed to analyze the performance of Sequential Probability Ratio Test (SPRT) and Cumulative-Sum, or CUSUM, procedures. These techniques account for overshoot that occurs in these tests. The proposed techniques are presented for Gaussian shift in mean and variance, Bernoulli, and arbitrary discretely distributed measurements. The improved accuracy of using these techniques over using Wald's approximations are shown. We extend these numerical procedures to distributed systems to first analyze a centralized CUSUM, and then to analyze repeating local sensor CUSUMs used in decentralized systems that re-initialize on local test termination.

2. We propose two new distributed, decentralized systems based on minimax and Bayesian quickest change detection frameworks and consider the restriction of limited bandwidth at the FC. We apply our analysis of CUSUM to both designs to describe the arrival and quantity of binary reports from local sensors at the FC in terms of the probability of receiving $1, 2, ..., M$ reports in each time slot, where M is the number of sensors in the system.

3. We identify issues restricting the design of distributed, decentralized systems under the Bayesian problem formulation and motivate other avenues of solving the quickest change detection in practice.

4. We propose a design methodology for a distributed, decentralized system based

on the minimax quickest change detection formulation. We show how local sensor and FC thresholds can be chosen to meet a desired false alarm rate constraint and provide a method of analyzing the performance of these system designs.

1.5 Organization of Thesis

The organization of this thesis is as follows.

Chapter 2 outlines important historical work and results regarding hypothesis testing and change detection problems. Different formulations of these problems are reviewed. The SPRT is described in detail as a way to address the hypothesis testing problem. Limitations of this procedure are identified. An extension of the SPRT to address the change detection problem, called the CUSUM procedure, is presented and characterized through expressions describing its performance. An optimal stopping approach to solving the change detection problem is also introduced. Finally, some examples of recent work applying these procedures to distributed systems with and without limits on communication are provided.

In Chapter 3, numerical procedures for analyzing the performance of the SPRT and CUSUM procedures are presented. The accuracy of these calculations are shown with respect to Monte Carlo simulations and the improvement over using conventional approaches to their design are highlighted. The proposed procedures are applied to Gaussian shift in mean, Gaussian shift in variance, Bernoulli, and arbitrary discrete probability mass function (pmf) distributed observations. Examples on how these procedures can be applied to the distributed quickest change detection problem are discussed.

In Chapter 4, a distributed, decentralized system design is proposed based on a Bayesian formulation to the quickest change detection problem. This design uses the procedures developed in Chapter 3 to analyze local sensor performance. The FC updates its global statistic based on this analysis and the number of reports it receives from local sensors in each time slot. Limitations in using this type of problem formulation to address the quickest change detection problem from an implementation standpoint are discussed.

Motivated by the limitations of the Bayesian formulated design, in Chapter 5, a system design is proposed based on a minimax approach to the quickest change detection problem. This system involves using the CUSUM procedure at both local sensors and at the FC. The procedures of Chapter 3 are used to analyze the reporting probabilities of local sensors and allow us to propose a design methodology by which we can design local sensor and FC thresholds in this system according to a desired false alarm rate. Given these thresholds, we can then analyze the performance of a system design in terms of the expected detection delay for a guaranteed FC false alarm rate.

In Chapter 6, conclusions of the thesis are provided. Summaries and key results of the previous chapters are given. Avenues of future work are identified.

Chapter 2

Background

2.1 Introduction

In this section, sequential hypothesis testing and sequential change detection will be discussed. Historical and recent work will be identified to provide insight into this research area. The Sequential Probability Ratio Test (SPRT) will be introduced as a method of solving the hypothesis testing problem. Similarly, Bayesian and minimax formulations of the sequential change detection problem will be discussed, with solutions provided by the Shiryaev procedure and Page's cumulative-sum (CUSUM) test, respectively. Limitations of the above methods will also be identified.

2.2 Sequential Hypothesis Testing

Consider a sequence of Independent and Identically Distributed (IID) random variables X_1, X_2..., which are distributed according to pdf f_0 or f_1. This can be denoted

as

$$H_0 : \quad X_1, X_2, ..., \sim f_0 \tag{2.1}$$

$$H_1 : \quad X_1, X_2, ..., \sim f_1. \tag{2.2}$$

The sequential hypothesis testing problem involves designing a test that observes values of $X_1, X_2, ...$, to determine the true hypothesis. The performance of such tests are described by error probabilities and expected detection delays. The two types of error probabilities are false alarm and missed detection, denoted by α and β, respectively. These error probabilities are defined by

$$\alpha = P(\text{choose } H_1 | H_0 \text{ is true}) \tag{2.3}$$

$$\beta = P(\text{choose } H_0 | H_1 \text{ is true}). \tag{2.4}$$

When designing sequential tests, it is desirable to meet or exceed the error probabilities specified by the designer, and to have the least expected detection delay for either hypothesis being true. Wald proposed the SPRT as an optimal sequential test for this problem [42]. This test tracks the likelihood ratio of subsequent observations using the test statistic g_k, where

$$g_k = \sum_{i=1}^{k} \ln(\frac{f_1(X_i)}{f_0(X_i)}). \tag{2.5}$$

This test statistic is compared with two precomputed thresholds a and b after each observation. The design of suitable test thresholds are dictated by the designer's choice of α and β. Wald also proposed in [42] that a suitable range of threshold values

for such tests can be found using the inequalities

$$B \geq \frac{\beta}{1-\alpha} \tag{2.6}$$

$$A \leq \frac{1-\beta}{\alpha}. \tag{2.7}$$

To determine values of A and B in practice, Wald showed that a sufficient method to compute these thresholds such that they produce a test that always meets the designer's specified error probabilities is to consider the inequalities in (2.6) and (2.7) as approximate equalities,

$$B \simeq \frac{1-\beta}{\alpha} \tag{2.8}$$

$$A \simeq \frac{\beta}{1-\alpha}. \tag{2.9}$$

This approximation is known as Wald's approximation. Define the logarithm of these thresholds in the SPRT as

$$b = \ln(B) \tag{2.10}$$

$$a = \ln(A). \tag{2.11}$$

The SPRT detector structure is then

$$\phi^S(g_k) = \begin{cases} 1 & \text{if } g_k \geq b \\ 0 & \text{if } g_k \leq a \\ k \leftarrow k+1 & \text{else.} \end{cases} \tag{2.12}$$

where ϕ^S is the decision rule of the SPRT and g_k is given by Equation (2.5). A value

of $\phi^S(g_k) = 1$ indicates that the SPRT has chosen H_1, and conversely, $\phi^S(g_k) = 0$ indicates the SPRT has chosen H_0.

We define T as the number of observations needed for the SPRT to terminate, or the stopping time of the SPRT. We can also express this more formally as $T(\phi^S)$, the stopping time associated with decision rule ϕ^S. It was shown by Wald in [42] that the SPRT will always terminate after a finite number of observations,

$$\mathrm{P}(T < \infty) = 1. \tag{2.13}$$

Using the above expressions, Wald proposed approximate expressions in [42] for the stopping time of these tests under H_1 and H_0

$$E_1[T] \simeq \mu_1^{-1} \frac{aA(B-1) + bB(1-A)}{B-A} \tag{2.14}$$

$$E_0[T] \simeq \mu_0^{-1} \frac{a(B-1) + b(1-A)}{B-A}, \tag{2.15}$$

where

$$\mu_j = E_j[\ln(\frac{f_1(X_j)}{f_0(X_j)})] \text{ for } j = 0, 1. \tag{2.16}$$

The optimality of the SPRT was established through the Wald-Wolfowitz Theorem [43]. Denote any other sequential decision rule as ϕ, such that

$$\alpha(\phi) \leq \alpha(\phi^S) \tag{2.17}$$

$$\beta(\phi) \leq \beta(\phi^S), \tag{2.18}$$

where α and β are the error probabilities associated with decision rules ϕ^S and ϕ, as defined in Equations (2.3) and (2.4). Let $E_j[T(\phi^S)]$ denote the expected stopping

time under H_j using ϕ^S, and $E_j[T(\phi)]$ the expected stopping time under H_j using ϕ. The Wald-Wolfowitz theorem states that, for IID observations,

$$E_j[T(\phi)] \geq E_j[T(\phi^S)] \text{ for } j = 0, 1. \tag{2.19}$$

In other words, for a given false alarm and missed detection probability, the SPRT requires on average fewer observations than any other sequential decision procedure [43].

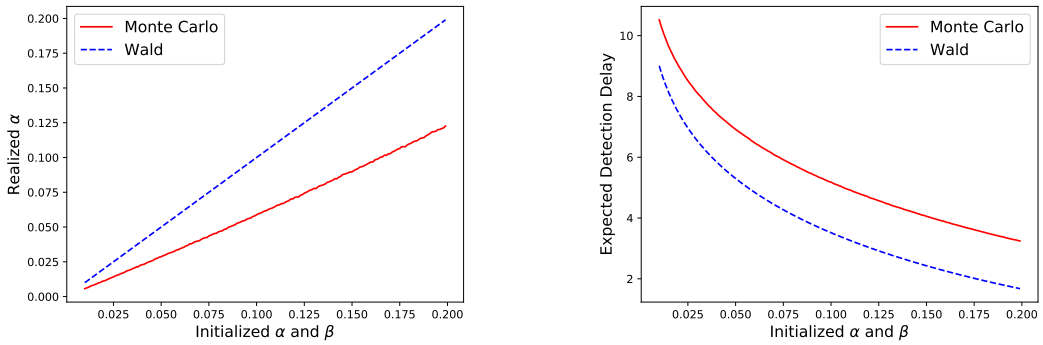
An important point to note about the SPRT is that, in general, the boundaries computed, a and b, will not provide with equality the desired error probability constraints. They will produce a test that provides a more conservative error probability. By extension of the Wald-Wolfowitz theorem, this means that the expected delay of such tests will increase.

These expressions ignore excess over the boundary, or overshoot. Overshoot occurs because at test termination, the test statistic, g_k, may not exactly be equal to a or b. The distribution of this overshoot, $g_n - a$ or $g_n - b$, determines the accuracy of Wald's approximations. If excess over the boundary is neglected, then (2.8) and (2.9) can be considered to design tests that meet error constraints with equality [48]. Equations (2.14) and (2.15) will also then compute the exact expected delay of the SPRT under H_1 and H_0.

The impact of overshoot on the design of SPRTs is illustrated by example in Figures 2.1(a) and 2.1(b). We denote the Gaussian pdf with mean μ and variance σ^2 as

$$N(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}. \tag{2.20}$$

Figure 2.1(a) shows the actual false alarm of $N(0,1)$ vs $N(1,1)$ SPRT with thresholds a and b designed using (2.10) and (2.11), respectively, found via Monte Carlo simulation. The α and β values used in these expressions were varied between 0.01 and 0.19. It is shown that the test designed by these expressions provides a lower false alarm than desired, resulting in a longer expected detection delay as displayed in Figure 2.1(b).



(a) False alarm

(b) Expected detection delay

Figure 2.1: Detectors implementing the SPRT for a N(0,1) vs N(1,1) test. The false alarm and expected detection delay versus error probability, where $\alpha = \beta$, predicted by Wald for this test is compared to Monte Carlo simulation.

As error probabilities decrease, the boundaries a and b get more negative and positive, respectively. As these boundaries tend towards negative and positive infinity, the relative amount of excess versus value of the boundary becomes smaller. In other words, these inequalities can be regarded as near equalities for very small error probabilities [21].

2.3 Sequential Change Detection

Consider now an IID sequence of observations corresponding to H_0. At some point in time, the true hypothesis may change to H_1. This can be denoted by

$$H_0: \quad X_1, X_2, ..., X_{\Gamma-1} \quad \sim f_0 \tag{2.21}$$

$$H_1: \quad X_\Gamma, X_{\Gamma+1}, ..., X_{\Gamma+k} \quad \sim f_1, \tag{2.22}$$

where Γ represents the unknown change time between distributions.

The goal of change detection tests is to detect this change of hypothesis as quickly as possible according to some acceptable error constraint. Bayesian and minimax formulations of this problem are described next.

2.3.1 Minimax Formulation

The minimax formulation assumes no known change point probability distribution for Γ. Lorden proposed one version of this type of formulation [22]. He assumed that the change time is determistic but unknown. Performance indices that characterize solutions to this formulation are the worst-case expected detection delay and the average run length (ARL) to false alarm.

The worst-case expected detection delay is defined as the expected delay for a CUSUM to choose H_1 under the worst-case initial condition, where the CUSUM test statistic is equal to zero when the change in hypothesis from H_0 to H_1 occurs. The average run length to false alarm is given by

$$\text{ARL to FA} = E_0[T], \tag{2.23}$$

where T is the stopping time of a CUSUM test. This can be thought of as the expected time until a false alarm given that no change ever occurs, and thus H_0 is always true. We therefore seek solutions to the change detection problem that

$$\begin{aligned} &\min\ E_1[T] \\ &\text{subject to ARL to FA} \geq \zeta. \end{aligned} \tag{2.24}$$

Lorden showed that Page's CUSUM test was an asymptotically optimal test in minimizing $E_1[T]$ as $\zeta \to \infty$. Moustakides extended this result in [24] by showing that the CUSUM test was optimal in the case of minimizing the worst case expected delay for $1 < \zeta < \infty$ among all stopping rules that satisfy $E_0[T] = \zeta$.

The CUSUM test was originally proposed in [27], with decision rule defined by

$$\phi^C(S_k) = \begin{cases} 1 & \text{if } S_k \geq h \\ k \leftarrow k+1 & \text{else,} \end{cases} \tag{2.25}$$

where S_k is the CUSUM test statistic given by

$$S_k = \max_{1 \leq j \leq k} \sum_{i=j}^{k} \ln\left(\frac{f_1(X_i)}{f_0(X_i)}\right) \tag{2.26}$$

and h is the detection threshold. A value of $\phi^C(S_k) = 1$ indicates that the CUSUM test has terminated and concluded H_1 is true at time k. This means that a change has occurred. If S_k has not surpassed h, H_0 is considered true and another observation is taken. Page showed that CUSUM can be expressed as a repeated sequence of SPRTs with boundaries 0 and h, and an initial statistic S_0 of zero. Each time the SPRT exceeds the lower boundary, it is reset to zero. This can be seen by rewriting (2.26)

as

$$S_k = \max(S_{k-1} + L(X_k), 0) \text{ for } k \geq 1 \tag{2.27}$$

$$S_0 = 0, \tag{2.28}$$

where

$$S_{k-1} = \sum_{j=1}^{k-1} L(X_j), \tag{2.29}$$

and

$$L(X_j) = \ln(\frac{f_1(X_j)}{f_0(X_j)}). \tag{2.30}$$

This test is continuously run until the detection threshold is reached. Algorithm 1 summarizes the CUSUM procedure.

Algorithm 1 CUSUM test

```
k = 0
S_0 = 0
    loop:
        S_{k+1}=S_k + log(f_1(X_{k+1}) / f_0(X_{k+1}))
        if S_{k+1} ≥ h:
            break
        if S_{k+1} < 0:
            S_{k+1} = 0
        k = k + 1
    end loop
```

Wald's approximations can also be used to obtain expressions that relate the ARL to false alarm and expected detection delay with the CUSUM detection threshold, h.

This process is described in [32], where the following expressions are derived:

$$E_0[T] \simeq \frac{|e^h - h - 1|}{|\mu_0|} \tag{2.31}$$

$$E_1[T] \simeq \frac{|e^{-h} + h - 1|}{|\mu_1|} \tag{2.32}$$

where

$$\mu_\theta = E_\theta[\ln(\frac{f_1(X_\theta)}{f_0(X_\theta)})] \text{ for } \theta = 0, 1. \tag{2.33}$$

The approximations in these expressions have the same origin as (2.14) and (2.15).

2.3.2 Bayesian Formulation

In the Bayesian formulation of the change detection problem, the change time Γ is assumed to be random with some known distribution. Often, this distribution is considered to be geometric according to change parameter, ρ, that is assumed known and a prior distribution, π_0. This type of change distribution is popular because of the memoryless property of geometric distributions. The prior distribution, π_0, indicates the probability that the change occurs before the start of this test. This change distribution is characterized by

$$P(\Gamma = k) = \pi_0 I_{k=0} + (1 - \pi_0)\rho(1 - \rho)^{k-1} I_{k\geq 1}, \tag{2.34}$$

where I_k denotes the indicator function, defined as

$$I_k = \begin{cases} 1 & \text{if } k = n \\ 0 & \text{if } k \neq n \end{cases} \tag{2.35}$$

and

$$I_{k\geq n} = \begin{cases} 1 & \text{if } k \geq n \\ 0 & \text{if } k < n. \end{cases} \tag{2.36}$$

The goal of this formulation is to detect a change in distribution as quickly as possible subject to a prescribed false alarm probability constraint. Unlike the previously discussed minimax formulation, any solution to this problem has a false alarm probability that is less than one. The stopping time for such a test, T, can be optimized by considering the trade off between the expected detection delay and probability of false alarm by formulating an optimization problem. This trade off has been formulated by Shiryaev in [31] as

$$\inf_T [P_0(T < \Gamma) + cE_1[(T - \Gamma)^+], \tag{2.37}$$

where c denotes the relative cost of delay versus false alarm, and $(x)^+ = \max(x, 0)$.

It was shown by Shiryaev that the posterior probability of a change occurring at or before time k, π_k, can be used as a suitable statistic in these problems [31]. The posterior probability, π_k is defined as

$$\pi_k = P(\Gamma \leq k | F_k) \tag{2.38}$$

where F_k denotes the history of the prior observations up to the k^{th} time slot. In other words, the posterior probability is a function of all observations observed by the detector. Under the geometric change distribution of (2.34), the sequence of

posteriors, π_k can be computed recursively by

$$\pi_k = \frac{(\pi_{k-1} + (1-\pi_{k-1})\rho) f_1(X_k)}{(\pi_{k-1} + (1-\pi_{k-1})\rho) f_1(X_k) + (1 - (\pi_{k-1} + (1-\pi_{k-1})\rho)) f_0(X_k)}. \tag{2.39}$$

It was shown by Shiryaev in [31] that for an appropriately chosen threshold π^*, the following decision rule is optimum in solving the above optimization problem:

$$\phi^B(\pi_k) = \begin{cases} 1 & \text{if } \pi_k \geq \pi^* \\ k \leftarrow k+1 & \text{else.} \end{cases} \tag{2.40}$$

A value of $\phi^B(\pi_k) = 1$ indicates a change has occurred at time k, otherwise another observation is taken.

2.4 Distributed Detection

More recently, sequential detection procedures have been applied to distributed problems. In such cases, there may be a number of detectors that each observe a different sequence of observations. These sensors can either forward observations to a global decision maker, called a fusion center (FC), or transmit summary reports of the observations. The FC is often in charge of making a final decision on the test, based on the information that is sent.

A number of existing works have explored distributed hypothesis testing and change detection. In [13], a decentralized system is proposed for the distributed hypothesis testing problem. The SPRT is used at both local sensors and the FC. The choice of using the SPRT globally is motivated by the fact that local detector decisions, under certain conditions, can be viewed as an IID input process at the FC.

Approximate expressions for the error probabilities and expected detection delays at the fusion center are also provided, with conditions under which these approximations may become exact. A comparison of the proposed system to a distributed system employing a centralized SPRT is provided.

A similar system is considered in [46], but proposes a level triggered sampling mechanism to sample the log likelihood ratio function at local sensors. Such a procedure is similar to the SPRT. Thresholds are set a fixed distance from the starting point of the test. When the accumulated log likelihood ratio passes either threshold, a single bit indicating which threshold was passed is transmitted to the FC. The local test is then restarted, with two new thresholds initiated the same fixed distance from the current starting point. Fusion rules are derived for channels between local sensors and the FC under different noise models. The asymptotic performance in terms of detection delay of this system over different channels is found.

A Bayesian framework for decentralized hypothesis testing is proposed in [41]. Local sensors send summary messages of their observations to the FC, which performs a sequential test globally to determine the true hypothesis. The design goal of this system is to minimize the total expected cost resulting from the sequential procedure over all admissible decision policies at the FC, and over all possible choices of local decision functions. This problem is shown to be tractable for the case where the system has full feedback and local memory restricted to past decisions, and a solution is obtained via a unique fixed point solution to a Bellman equation, which is referred to in [41] as dynamic programming. Finite horizon and infinite horizon versions of this problem are considered.

A distributed change detection problem is considered in [26]. The change in

distribution can occur at different times for each of the N sensors in the system. A one-shot CUSUM scheme is proposed, where local sensors implement a CUSUM procedure and only communicate once with the FC to signal a detection. A minimal strategy is used at the FC, meaning that at the time of the first detection signaled by a local sensor, an alarm is triggered. It is demonstrated that as the mean time between false alarms tends to infinity, there is no loss of information at the FC by employing this minimal one shot strategy compared to a centralized procedure, with respect to Lorden's criterion.

A distributed system where local sensors quantize their observations to a finite alphabet and transmit them to the FC is considered in [25]. A CUSUM strategy is applied at the FC on the received observations. The performance of the global CUSUM test is related to the local sensor quantizers. A means of obtaining combined detection and quantization strategies is proposed that utilize Lorden's criterion.

In [40], a Bayesian formulation of the distributed change detection problem is proposed, where a priori knowledge of the change distribution is available. An optimal solution that utilizes dynamic programming is provided and evaluated numerically for different observation distributions.

2.5 Detection with Energy Constraints

A number of approaches consider the problem of detection in energy constrained environments. In [11] and [12], minimax and Bayesian formulations of the change detection problem with energy constraints on a single sensor are considered. The sensor's energy is used by taking an observation and is replenished according to a random process. Such a formulation could be useful in modelling wireless sensor

networks powered by renewable energy. The authors propose a power allocation scheme and detection strategy for such a problem.

A constraint on the expected number of observations that can be observed is considered in [4] and [3]. The proposed approaches are shown to be asymptotically optimal as the false alarm rate and false alarm probability, respectively, approach zero. [5] builds off these works by considering this problem in a distributed system. In addition to a cost of obtaining observations, a cost of communication between local sensors and the FC is also considered. Two algorithms are proposed and a performance analysis is provided.

In [44], the distributed change detection problem within a Bayesian framework is considered. Constraints on both the probability of false alarm and on the number of communications between local sensors and the FC are proposed in this problem. Solutions for systems with a constraint on the expected number of communications and on the absolute number of communications are proposed. The CUSUM procedure is used at local sensors. When the CUSUM procedure terminates at the local sensor, it sends a binary report to the FC indicating it has reached a decision, before reinitializing. The arrival of such reports at the FC is assumed to be a Poisson process. This allows the FC to update its posterior statistic indicating its confidence in a change having occurred in the system. The FC is in charge of making a final decision on whether a change has occurred.

Chapter 3

Numerical Computation Methods of Exact SPRT Performance

3.1 Introduction

In this chapter, an algorithm is introduced that allows for overshoot to be taken into account in the design of SPRT thresholds, allowing for more accurate tests with respect to design parameters. This algorithm is based on generalizing recursive computations of SPRT performance that were developed in [7] for Gaussian shift in mean problems. Here, we extend this procedure to Gaussian shift in variance, Bernoulli, and discrete probability mass function (pmf) type problems, and propose its application to CUSUM tests. Results obtained from numerical simulations of the proposed procedure are compared to those obtained from Monte Carlo trials and Wald's approximation. Applications of the developed procedure to the distributed quickest change detection problem are discussed.

3.2 Algorithm

Consider the SPRT introduced in Chapter 2. After each observation, the log likelihood ratio is computed and added to the cumulative statistic g_k. Each log likelihood ratio observation is a random variable, and as such, has its own probability density function (pdf) that is dictated by the pdfs of H_0 and H_1. This is denoted in the equations below as $f(w)$. We note that the pdf of a sum of independent random variables is their convolution.

Without loss of generality, from here on we assume H_1 is true. If H_0 is true, the derivation is similar. Consider an SPRT with lower and upper log thresholds a and b, respectively. The pdf of the SPRT statistic in stage k given that it reaches stage k, $f_k(x)$, can be computed using the fact that the pdf of each log likelihood ratio observation is IID.

Given that the test reaches stage 1 with probability one, the probability of reaching stage k can be expressed in terms of the conditional probabilities

$$d_k(H_1) = \prod_{j=2}^{k} \mathrm{P}(\text{test reaches stage } j \mid \text{ test reaches stage } j-1), \tag{3.1}$$

where $k \geq 2$. In the terms of the above equation, the probability of reaching SPRT test stage $k+1$ given that the test reaches stage k must be

$$\mathrm{P}(\text{test reaches stage } k+1 \mid \text{ test reaches stage } k) = \int_a^b f_k(x)dx, \tag{3.2}$$

where

$$f_k(x) = \frac{(f \star f_{k-1}^{trunc})(x)}{\int_{-\infty}^{\infty}(f \star f_{k-1}^{trunc})(t)dt} \tag{3.3}$$

and

$$f_{k-1}^{trunc}(x) = \begin{cases} f_{k-1}(x) & a < x < b \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

The initial term of Equation (3.3), $f_1(x)$, is the pdf of a single log likelihood ratio observation. In other words, the pdf of a SPRT statistic in stage k is the convolution of a single log likelihood ratio observation pdf, $f(w)$, and the truncated pdf of the SPRT statistic at the previous delay stage, $f_{k-1}^{trunc}(x)$, where this truncation occurs between thresholds a and b. Normalization is needed in Equation (3.3) to ensure $f_k(x)$ is a pdf because of this truncation. We can therefore obtain the probability of reaching each SPRT stage by successive convolutions using numerical integration, and integrating the result over the region between the SPRT thresholds.

The probability of choosing H_1 at stage k given that stage k is reached is given by

$$\begin{aligned} \gamma_k(H_1) &\equiv \mathrm{P}(\text{choose } H_1 \text{ at stage } k \mid \text{ test reaches stage } k) \\ &= \int_b^\infty f_k(x)dx \\ &= \int_{-\infty}^\infty f_k(x)dx - \int_{-\infty}^b f_k(x)dx \\ &= 1 - \int_{-\infty}^b f_k(x)dx \\ &= 1 - \int_{-\infty}^b \int_a^b f(x-w) f_{k-1}(w)dwdx \\ &= 1 - \int_a^b \int_{-\infty}^b f(x-w) f_{k-1}(w)dxdw \\ &= 1 - \int_a^b F(b-w) f_{k-1}(w)dw. \end{aligned} \tag{3.5}$$

Using (3.2) and (3.5), the probability of choosing H_1 at any stage is then

$$\text{P(choose } H_1) = \sum_{k=1}^{\infty} \gamma_k(H_1) d_k(H_1) \tag{3.6}$$

with corresponding missed detection probability

$$\beta = \text{1-P(choose } H_1). \tag{3.7}$$

The average test length of an SPRT given H_1, $E_1[T]$, can be calculated as

$$E_1[T] = \sum_{k=1}^{\infty} d_k(H_1). \tag{3.8}$$

When (3.6) is truncated to a finite number of stages, N, a lower bound is achieved for $P(\text{choose } H_1)$. An upper bound can also be obtained by adding the probability of reaching stage $N+1$ to this

$$\sum_{k=1}^{N} \gamma_k(H_1) d_k(H_1) < P(\text{choose } H_1) < \sum_{k=1}^{N} \gamma_k(H_1) d_k(H_1) + d_{N+1}(H_1). \tag{3.9}$$

These bounds can be used to determine at which stage it is appropriate to terminate execution of the procedure. We seek a value of N that, if we exceed it, will insignificantly affect our computed delay and error probabilities. Using (3.9), we can determine when proceeding to stage $N+1$ will have diminishing returns on the computed error probability compared to stopping at stage N. For our numerical results, we used a value of N where continuing to $N+1$ had an effect of less than three orders of magnitude relative to stage N. In other words, this is when the probability of reaching the next stage is three orders of magnitude less than the current error probability. We define this difference in magnitude between delay stages as ϵ and

show this method of termination in Algorithm 2.

3.3 Application to SPRT

The expressions in Section 3.2 are computed numerically for different observation distributions and compared to results from Monte Carlo trials. Results for tests observing Gaussian shift in mean and variance, Bernoulli and discrete pmf observation distributions are shown in the following sections. The structure of each detector is dependent on the observation distribution and are derived for each case. All Monte Carlo trials are performed using 10^5 iterations. Algorithm 2 shows pseudocode describing the implementation of the above expressions on SPRTs.

Algorithm 2 SPRT Analysis Algorithm when H_1 is true

$k = 1$
while true:
    generate $f(x)$
    if $k = 1$:
        $f_1(x) = f(x)$
        P(reach stage 1) = 1
        P(reach stage 2) = $\int_a^b f_1(x)dx$
        P(accept H_1 at stage 1) = $\int_b^\infty f_1(x)dx$
        P(accept H_1 at any stage up to 1) = P(reach stage 1)P(accept H_1 at stage 1)
        error(1)= 1 − P(accept H_1 at any stage up to 1)
        if $\frac{\text{error(1)+P(reach stage 2)}}{\text{P(reach stage 2)}} \geq \epsilon$:
            break
    if $k \geq 2$:
        f_{k-1}^{trunc}=truncate(f_{k-1},a,b)
        f_k=$(f_{k-1}^{trunc} \star f)(x)$
        P(reach stage $k+1$) = $\int_a^b f_k(x)dx$
        P(accept H_1 at stage k) = $\int_b^\infty f_k(x)dx$
        P(select H_1 at any stage) = P(reach stage k)P(accept H_1 at stage k) + P(select H_1 at any stage)
        error(k) = 1 − P(select H_1 at any stage)
        if $\frac{\text{error}(k)+\text{P(reach stage } k+1)}{\text{P(reach stage } k+1)} \geq \epsilon$:
            break
    $k = k + 1$
end while
$E_\theta[T]$=sum(P(reach stage k))

3.3.1 Gaussian Shift in Mean

For the Gaussian shift in mean test, the pdf of the log likelihood ratio of each observation will also be Gaussian. For a $N(0,1)$ vs $N(S,1)$ test, the mean and variance of this pdf and the corresponding test thresholds can be computed in terms of the log likelihood function

$$\sum_{k=1}^{n} L(X_{k|\theta}) = \begin{cases} \text{choose } H_1 & \geq b \\ \text{choose } H_0 & \leq a \end{cases} \tag{3.10}$$

where

$$\begin{aligned} L(X_{k|\theta}) &= \ln\left(\frac{\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(X_{k|\theta}-S)^2}{2}}}{\frac{1}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{X_{k|\theta}^2}{2}}}\right) && (3.11) \\ &= \ln(e^{X_{k|\theta}S - \frac{S^2}{2}}) && (3.12) \\ &= X_{k|\theta}S - \frac{S^2}{2} && (3.13) \\ &= Y_{k|\theta} \sim N(\mu_\theta^{'}, \sigma_\theta^{'2}),\ \theta\epsilon\{0,1\}. && (3.14) \end{aligned}$$

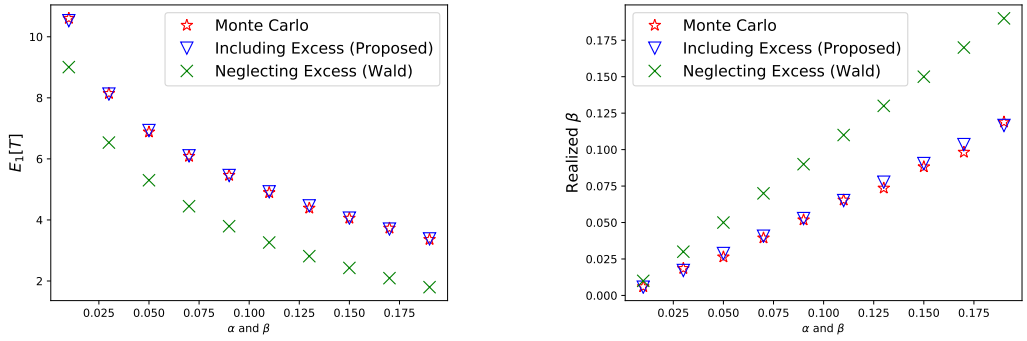
The mean and variance of the Gaussian log likelihood ratio pdf can be found by taking the expectation and variance of (3.14) with respect to the true hypothesis. This results in

$$\begin{aligned} \mu_1^{'} &= \frac{S^2}{2} && (3.15) \\ \mu_0^{'} &= \frac{-S^2}{2} && (3.16) \\ \sigma_\theta^{'2} &= S^2. && (3.17) \end{aligned}$$

The test thresholds, b and a, are found from (2.10) and (2.11), respectively.

Using (3.14), Algorithm 2 is applied to numerically compute the expected detection delay and probability of missed detection of a $N(0,1)$ vs $N(1,1)$ test where the test thresholds are designed using Wald's approximation with varying desired error probabilities. Our aim is to show that, for desired error probabilities α and β, computing test thresholds using Wald's approximation over designs the SPRT with respect to desired error probabilities. This causes increased expected detection delay. Using the technique developed above, we can better estimate the SPRT error probabilities and expected detection delay. This can then be used to compute more accurate test thresholds with respect to error constraints, resulting in lower expected detection delay.

A comparison between results obtained by accounting for excess over the boundary, or sequential test overshoot, using our proposed numerical computations, Monte Carlo simulation and Wald's approximation are shown in Figures 3.1(a) and 3.1(b). Numerically computed values closely approximate those obtained via Monte Carlo, which is a significant improvement in accuracy compared to using Wald's approximation.



(a) Expected delay (b) Missed detection

Figure 3.1: A comparison between numerical and Monte Carlo for a N(0,1) vs N(1,1) SPRT. Desired missed detection probability, β, is set equal to false alarm probability, α.

3.3.2 Gaussian Shift in Variance

For the Gaussian shift in variance case, the log likelihood ratio test structure of (3.10) for a N(0,1) vs N(0,V) test is now:

$$L(X_{k|\theta}) = \ln\left(\frac{\frac{1}{\sqrt{2\pi V}}e^{-\frac{X_{k|\theta}^2}{2V}}}{\frac{1}{\sqrt{2\pi}}e^{-\frac{X_{k|\theta}^2}{2}}}\right) \tag{3.18}$$

$$= \ln\left(\frac{1}{\sqrt{V}}e^{-\frac{X_{k|\theta}^2}{2V}+\frac{X_{k|\theta}^2}{2}}\right) \tag{3.19}$$

$$= \ln\left(\frac{1}{\sqrt{V}}\right) - \frac{X_{k|\theta}^2}{2V} + \frac{X_{k|\theta}^2}{2} \tag{3.20}$$

which can be simplified and rearranged to

$$2\left(\frac{a-\ln\left(\frac{1}{\sqrt{V}}\right)}{1-\frac{1}{V}}\right) < X_{k|\theta}^2 < 2\left(\frac{b-\ln\left(\frac{1}{\sqrt{V}}\right)}{1-\frac{1}{V}}\right). \tag{3.21}$$

This results in the detector structure

$$\sum_{k=1}^{n} X_{k|\theta}^2 = \begin{cases} \text{choose hypothesis 1} & \geq b_n^{'} \\ \text{choose hypothesis 0} & \leq a_n^{'} \end{cases} \tag{3.22}$$

where

$$X_{k|\theta} \sim N(0, \sigma_\theta^2) \tag{3.23}$$

and

$$b_n^{'} = \quad 2(\frac{b - n\ln(\frac{1}{\sqrt{V}})}{1-\frac{1}{V}}) \tag{3.24}$$

$$a_n^{'} = \quad 2(\frac{a - n\ln(\frac{1}{\sqrt{V}})}{1-\frac{1}{V}}). \tag{3.25}$$

In this case, the test thresholds $a_n^{'}$ and $b_n^{'}$ increase after every observation observed. Under H_0, $X_{k|0}^2$ is first-order Chi-squared distributed, denoted as $\chi^2(1)$, where

$$\chi^2(z) = \frac{1}{2^{\frac{z}{2}}\mathcal{G}(\frac{z}{2})} x^{\frac{z}{2}-1} e^{-\frac{x}{2}}, \tag{3.26}$$

and $\mathcal{G}(m)$ is the gamma function

$$\mathcal{G}(m) = (m-1)!. \tag{3.27}$$

This follows from the fact that under H_0, $X_{1|0}, X_{2|0}, ..., X_{n|0}$ are independent standard

normal random variables. The sum of their squares

$$Q_0 = \sum_{k=1}^{n} X_{k|0}^2 \tag{3.28}$$

is distributed according to the Chi-squared distribution with n degrees of freedom, $Q_0 \sim \chi^2(n)$. Since the pdf of the cumulative SPRT statistic is truncated after each observation at boundaries a and b, it is more useful to consider each individual term $X_{k|0}^2$ as being first order Chi-squared distributed.

Similarly under H_1, $X_{k|1}^2$ is Gamma distributed, denoted by $\Gamma(\frac{1}{2}, 2V)$, where

$$\Gamma(y, z) = \frac{1}{z^y \mathcal{G}(y)} x^{y-1} e^{-\frac{x}{z}}. \tag{3.29}$$

This is because under H_1, $X_{1|1}, X_{2|1}, ..., X_{n|1}$ are independent zero mean Gaussian random variables with variance V. The sum of their squares

$$Q_1 = \sum_{k=1}^{n} X_{k|1}^2 \tag{3.30}$$

is distributed according to the Gamma distribution, $Q_1 \sim \Gamma(\frac{n}{2}, 2V)$. Since again the pdf of the cumulative SPRT statistic is truncated at boundaries a and b, it is similarly more useful to consider each individual term $X_{k|1}^2$ as being Gamma distributed with $n = 1$.

In practice, performing the numerical computations with $\chi^2(1)$ and $\Gamma(\frac{1}{2}, 2V)$ distributions is impossible because these pdfs are not bounded at $x = 0$. This problem is illustrated Figure 3.2(a). This issue was bypassed by truncating each pdf at a value close to zero, $x = t$, and normalizing the resulting pdf by the respective distribution's

cdf evaluated at this truncation value, $1 - F(t)$. For example, the value of t used in Figure 3.2(b) to show the resulting pdfs was $t = 0.01$. This provides approximations to these pdfs, denoted $\widetilde{\chi}^2(1)$ and $\widetilde{\Gamma}(\frac{1}{2}, z)$, that are given by

$$\widetilde{\chi}^2(1) = \begin{cases} \frac{\frac{1}{2^{\frac{1}{2}}\mathcal{G}(\frac{1}{2})} x^{-\frac{1}{2}} e^{-\frac{x}{2}}}{1 - F_{chi}(t;1)} & \text{if } x \geq t \\ 0 & \text{if } x < t, \end{cases} \tag{3.31}$$

where $F_{chi}(t; 1)$ is the cdf of a first order Chi-squared distribution

$$F_{chi}(t; 1) = \frac{1}{\mathcal{G}(\frac{1}{2})} \gamma(\frac{1}{2}, \frac{t}{2}), \tag{3.32}$$

and

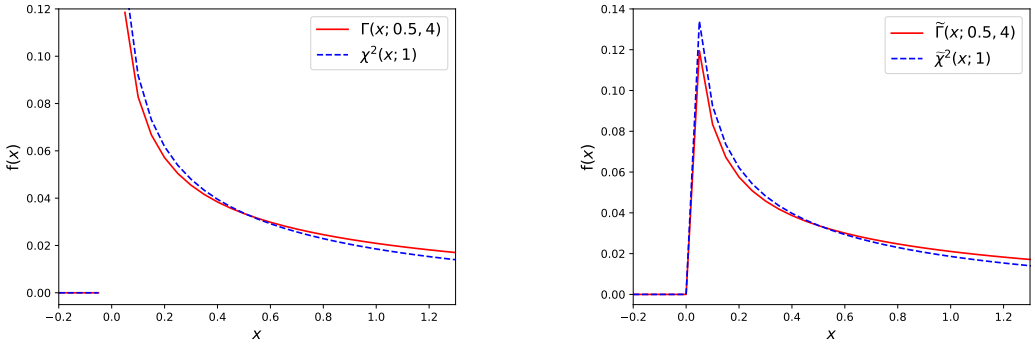
$$\widetilde{\Gamma}(\frac{1}{2}, z) = \begin{cases} \frac{\frac{1}{z^{\frac{1}{2}}\mathcal{G}(\frac{1}{2})} x^{-\frac{1}{2}} e^{-\frac{x}{z}}}{1 - F_{gamma}(t; \frac{1}{2}, z)} & \text{if } x \geq t \\ 0 & \text{if } x < t, \end{cases} \tag{3.33}$$

where $F_{gamma}(t; \frac{1}{2}, z)$ is the cdf of a Gamma distribution with shape parameter $y = \frac{1}{2}$

$$F_{gamma}(t; \frac{1}{2}, z) = \frac{1}{\mathcal{G}(\frac{1}{2})} \gamma(\frac{1}{2}, \frac{t}{z}). \tag{3.34}$$

In these expressions, γ denotes the lower incomplete gamma function

$$\gamma(s, r) = \int_0^r x^{s-1} e^{-x} dx. \tag{3.35}$$

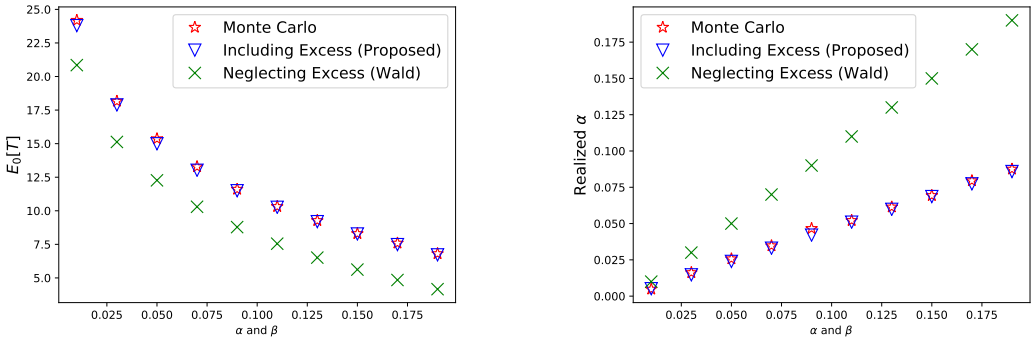


(a) Exact distributions

(b) Approximate distributions

Figure 3.2: A comparison between the exact and approximate first order Chi-squared and Gamma distributions with shape parameter $y = \frac{1}{2}$. The approximate functions are defined at $x = 0$.

Expected detection delay and false alarm values are numerically computed and compared to results from Monte Carlo simulation and Wald's approximation in Figures 3.3(a) and 3.3(b). These approximate pdf's were used in the Monte Carlo trials to negate error caused by using different pdfs. Similar to the shift in mean case, results from the numerical computations closely approximate those from Monte Carlo.



(a) Expected delay

(b) False alarm

Figure 3.3: A comparison between numerical and Monte Carlo results for a N(0,1) vs N(0,3) Gaussian SPRT. Missed detection probability, β, is set equal to false alarm probability, α.

3.3.3 Bernoulli Change in Parameter

For the change in parameter p_θ Bernoulli test, the detector structure given in (3.10) can be expanded as:

$$
\begin{aligned}
L(X_{k|\theta}) &= \ln\left(\frac{p_1^{X_{k|\theta}}(1-p_1)^{1-X_{k|\theta}}}{p_0^{X_{k|\theta}}(1-p_0)^{1-X_{k|\theta}}}\right) && (3.36)\\
&= \ln(p_1^{X_{k|\theta}}) + \ln((1-p_1)^{1-X_{k|\theta}}) - \ln(p_0^{X_{k|\theta}}) - \ln((1-p_0)^{1-X_{k|\theta}}) && (3.37)\\
&= X_{k|\theta}(\ln(\frac{p_1}{p_0}) - \ln(\frac{1-p_1}{1-p_0})) - \ln(\frac{1-p_0}{1-p_1}), && (3.38)
\end{aligned}
$$

which can be simplified and rearranged to

$$
\frac{b}{\ln(\frac{p_1}{p_0}) - \ln(\frac{1-p_1}{1-p_0})} + \frac{\ln(\frac{1-p_0}{1-p_1})}{\ln(\frac{p_1}{p_0}) - \ln(\frac{1-p_1}{1-p_0})} < X_{k|\theta} < \frac{b}{\ln(\frac{p_1}{p_0}) - \ln(\frac{1-p_1}{1-p_0})} + \frac{\ln(\frac{1-p_0}{1-p_1})}{\ln(\frac{p_1}{p_0}) - \ln(\frac{1-p_1}{1-p_0})}. \quad (3.39)
$$

This results in the detector structure

$$\sum_{k=1}^{n} X_{k|\theta} = \begin{cases} \text{choose } H_1 & \geq b_n^{'} \\ \text{choose } H_0 & \leq a_n^{'} \end{cases} \tag{3.40}$$

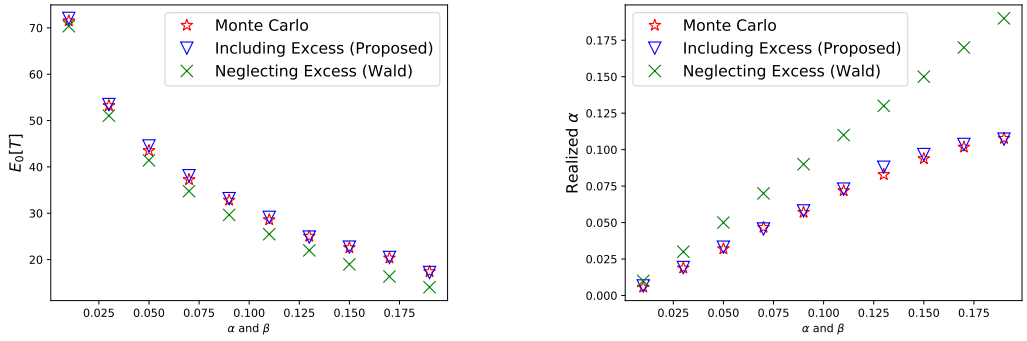
where

$$X_{k|\theta} \sim \text{Bernoulli}(p_\theta) \tag{3.41}$$

$$b_n^{'} = \frac{b}{\ln(\frac{p_1}{p_0}) - \ln(\frac{1-p_1}{1-p_0})} + n\frac{\ln(\frac{1-p_0}{1-p_1})}{\ln(\frac{p_1}{p_0}) - \ln(\frac{1-p_1}{1-p_0})} \tag{3.42}$$

$$a_n^{'} = \frac{b}{\ln(\frac{p_1}{p_0}) - \ln(\frac{1-p_1}{1-p_0})} + n\frac{\ln(\frac{1-p_0}{1-p_1})}{\ln(\frac{p_1}{p_0}) - \ln(\frac{1-p_1}{1-p_0})}. \tag{3.43}$$

Similar to the Gaussian shift in variance case, test thresholds $a_n^{'}$ and $b_n^{'}$ increase after every subsequent observation. In other words, test thresholds at stage n differ from the test thresholds at stage $n+1$. Expected detection delay and probability of false alarm are computed using the proposed numerical computations for an example with $p_0 = 0.04$ and $p_1 = 0.15$ and test thresholds computed from Wald's approximation. These results are compared in Figures 3.4(a) and 3.4(b) with those predicted from Wald and from Monte Carlo simulation. Similar to the above cases, by accounting for sequential test overshoot using our numerical computations we can closely approximate results obtained from Monte Carlo simulation.



(a) Expected delay

(b) False alarm

Figure 3.4: A comparison between numerical and Monte Carlo results for a p=.04 vs p=.15 Bernoulli SPRT. Missed detection probability, β, is set equal to false alarm probability, α.

3.3.4 Change in Discrete Probability Mass Function Test

This procedure can also be applied to a change in pmf test. This can be considered as a generalization of the change in parameter Bernoulli test discussed above, where observations can now take on a finite set of discrete values. To do this, we expand the log likelihood ratio function in the detector structure from (3.10) to

$$Z_{k|\theta} = L(X_{k|\theta}) \tag{3.44}$$

$$L(X_{k|\theta}) = \ln\left(\frac{p_{n|H_1}(X_{k|\theta})}{p_{n|H_0}(X_{k|\theta})}\right), \tag{3.45}$$

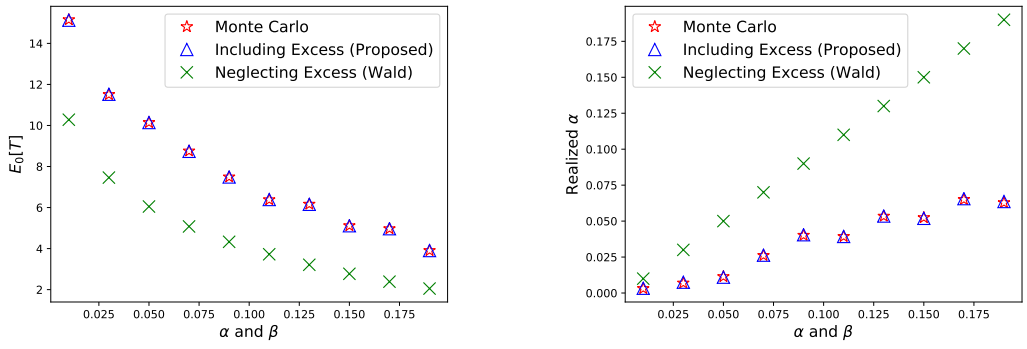
where $p_{n|H_1}(X_{k|\theta})$ and $p_{n|H_0}(X_{k|\theta})$ are discrete pmfs with n non-zero terms describing the observations under H_1 and H_0, respectively. The pmf of each log likelihood ratio

sample can then be found by

$$P_{Z_{k|\theta}}(z_i) \quad = \quad \sum_{\{j:L_k(x_j)=z_i\}} p_{n|\theta}(x_j). \tag{3.46}$$

Expected detection delay and probability of false alarm for an example set of pmfs are numerically computed and compared with those found from Monte Carlo simulation and using Wald's approximation in Figures 3.5(a) and 3.5(b). The examples pmfs considered are $p_{4|H_1}(x) = 0.587\delta(x)+0.325\delta(x-1)+0.078\delta(x-2)+0.011\delta(x-2)$ and $p_{4|H_0}(x) = 0.977\delta(x)+0.023\delta(x-1)+2.215 \text{ x } 10^{-4}\delta(x-2)+1.210 \text{ x } 10^{-6}\delta(x-3)$. Similar to the above cases, close agreement is shown between numerically computed results and those from Monte Carlo simulation.

We note that this detector structure can be used to analyze discrete hypothesis distributions with any number of non-zero terms. This procedure can therefore be used as an alternative method of analyzing the Bernoulli hypothesis distribution discussed above in Section 3.3.3.



(a) Expected delay

(b) False alarm

Figure 3.5: A comparison between numerical and Monte Carlo results for the change in pmf problem. Missed detection probability, β, is set equal to false alarm probability, α.

3.4 Application to CUSUM

CUSUM Analysis

A CUSUM test has lower and upper log thresholds 0 and h, respectively, where h is called the detection threshold and is set by the system designer. The CUSUM test can be analyzed using a similar procedure as what was shown for the SPRT in Section 3.2. We note that like the SPRT, the pdf of the CUSUM statistic in stage k given that it reaches stage k can be computed using the fact that the pdf of each log likelihood ratio observation is IID.

A CUSUM test will only terminate if the statistic S_k exceeds threshold h. This means the expression for computing the probability a test reaches stage $k+1$ given that stage k is reached in (3.2) changes to

$$\mathrm{P}(\text{test reaches stage } k+1 \mid \text{ test reaches stage } k) = \int_{-\infty}^{h} f_k(x)dx \tag{3.47}$$

where $f_k(x)$ can be found by

$$f_k(x) = \frac{(f \star f_{k-1}^{trunc})(x)}{\int_{-\infty}^{\infty}(f \star f_{k-1}^{trunc})(t)dt} \tag{3.48}$$

and

$$f_{k-1}^{trunc}(x) = \begin{cases} f_{k-1}(x) & 0 \le x < h \\ 0 & \text{otherwise.} \end{cases} \tag{3.49}$$

When a CUSUM test statistic, S_k, falls below the lower threshold 0, the test reinitializes itself by setting $S_k = 0$ and continuing operation. At each CUSUM stage, the

probability that the test will reset is found by

$$P(\text{reset at stage k}) = \int_{-\infty}^{0^-} f_k(x)dx. \tag{3.50}$$

This probability is added to the current stage's truncated pdf at $f_k^{trunc}(0)$ prior to observing the next observation. This allows for the potential of a reset to be accounted for in the subsequent CUSUM stage's pdf. The probability of choosing H_1 in stage $k+1$ given that stage $k+1$ is reached and the expected run length given either hypothesis are found using (3.5) and (3.8) with this new pdf. A modified version of Algorithm 2 is shown below to highlight the differences in this approach versus that of the SPRT analysis.

Algorithm 3 CUSUM Analysis Algorithm

$k = 1$
while true:
    generate pdf $f(x)$
    if $k = 1$:
        $f_1(x) = f(x)$
        P(reach stage 1) = 1
        P(reach stage 2) = $\int_{-\infty}^{h} f_1(x)dx$
        P(accept H_1 at stage 1) = $\int_{b}^{\infty} f_1(x)dx$
        if $\frac{\text{P(accept } H_1 \text{ at stage 1)+P(reach stage 2)}}{\text{P(reach stage 2)}} \geq \epsilon$:
            break
    if $k \geq 2$:
        P(reset at stage k-1) = $\int_{-\infty}^{0^-} f_{k-1}(x)$
        f_{k-1}^{trunc}=truncate($f_{k-1}, 0, h$)
        $f_{k-1}^{trunc}(0) = f_{k-1}^{trunc}(0)$ + P(reset at stage k-1)
        f_k=($f_{k-1}^{trunc} \star f$)(x)
        P(reach stage $k+1$) = $\int_{-\infty}^{h} f_k(x)dx$
        P(accept H_1 at stage k) = $\int_{h}^{\infty} f_k(x)dx$
        if $\frac{\text{P(accept } H_1 \text{ at stage } k\text{)+P(reach stage } k+1\text{)}}{\text{P(reach stage } k+1\text{)}} \geq \epsilon$:
            break
    $k = k + 1$
end while
$E_\theta[T]$=sum(P(reach stage k))

The probability of accepting H_1 unconditional on the delay stage is computed using the proposed procedure and compared to Monte Carlo simulation in the two

figures below. The unconditional probability of accepting H_1 at each delay stage is computed using

$$P(\text{choose } H_1 \text{ at stage } k) \quad = \quad \gamma_k(H_1) r_k(H_1). \tag{3.51}$$

Figure 3.6 shows close agreement between the numerical computations and Monte Carlo simulation at each delay stage when H_1 is true.

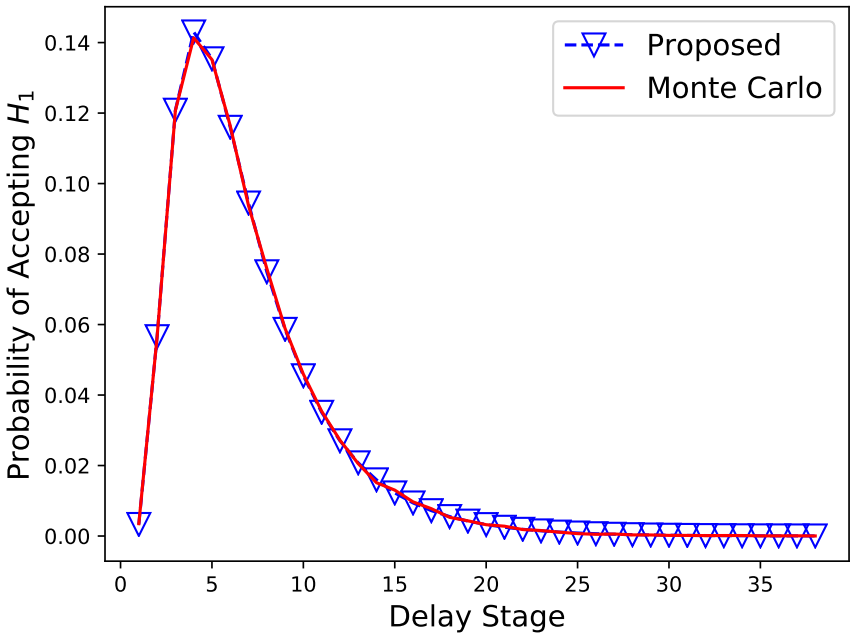


Figure 3.6: Probability of accepting H_1 in a $N(0,1)$ vs $N(1,1)$ test with H_1 true and $h = \ln(25)$. 10^6 Monte Carlo trials are distributed over 38 delay stages to compute the probabilities.

When H_0 is true in Figure 3.7, there is minor discrepancy between Monte Carlo and numerical results. This is because the estimation of low probabilities from Monte Carlo simulations requires a very large number of trials. It was therefore necessary to

verify that the numerically computed results fall within the 90% confidence interval of the probabilities generated by Monte Carlo simulation. The 90% confidence interval is computed using

$$90\% \text{ Confidence Interval} \quad = \quad 1.645\sqrt{\frac{P(1-P)}{W}}, \tag{3.52}$$

where P is the $P(\text{choose } H_1 \text{ at stage } k)$ and W is the number of Monte Carlo trials [9]. In Figure 3.7, 10^6 Monte Carlo trials are distributed over 55 delay stages, resulting in $W = \frac{10^6}{55}$. A value of $P = 0.005$, for example, results in a confidence interval of 8.6 x 10^{-4}. The numerically computed results in Figure 3.7 fall within the 90% confidence interval of those from Monte Carlo and are therefore verified to be accurate.

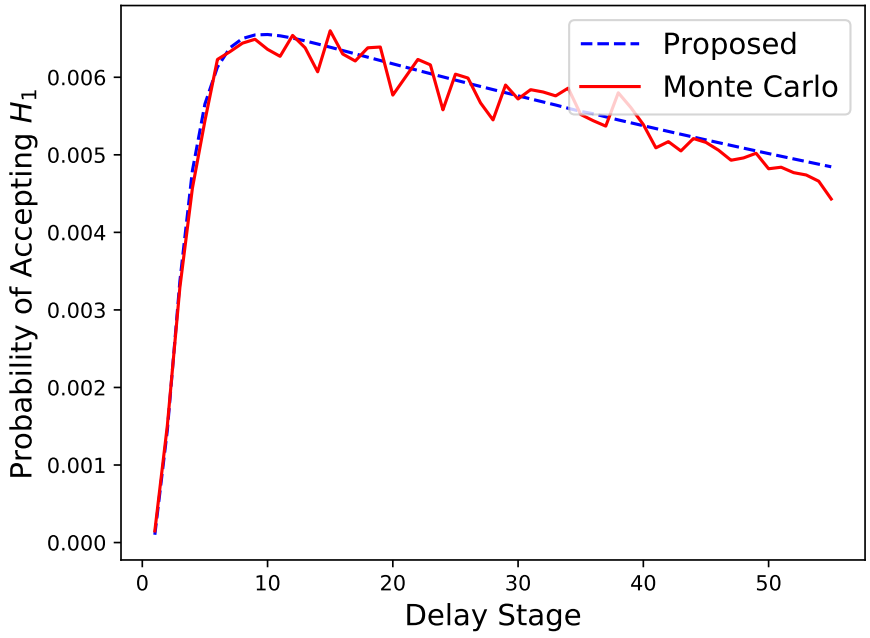


Figure 3.7: Probability of accepting H_1 in a $N(0,1)$ vs $N(1,1)$ test with H_0 true and $h = \ln(25)$. 10^6 Monte Carlo trials are used over 55 delay stages in simulating these probabilities.

Single SPRT Analysis

An alternative method of analyzing the performance of a CUSUM test can also be done through direct analysis of an SPRT.This method is useful specifically for estimating the ARL to false alarm and ARL to detection of a CUSUM test. The application of the analysis of a single SPRT to CUSUM takes advantage of the fact that, as discussed in Chapter 2, CUSUM can be represented as a sequence of repeated SPRTs.

Consider a SPRT with initial statistic $g_k = 0$ and upper and lower log thresholds h and 0, respectively. Let $P(H_1)$ be the probability that an SPRT terminates at upper threshold h. $E_\theta[T_h]$ and $E_\theta[T_0]$ are defined as the average number of samples for the SPRT given H_θ conditional on the test ending on h and 0, respectively. Page showed that the probability that n SPRTs choose H_0 before choosing H_1 is geometrically distributed, $(1 - P(H_1))^n P(H_1)$. The expected number of tests that choose H_0 is therefore

$$E[(1 - P(H_1))^n P(H_1)] = \frac{1 - P(H_1)}{P(H_1)}. \tag{3.53}$$

It follows that the ARL of a CUSUM test given H_θ being the true hypothesis is

$$\begin{aligned} ARL &= \frac{1 - P(H_1)}{P(H_1)} E_\theta[T_0] + E_\theta[T_h] \\ &= \frac{E_\theta[T]}{P(H_1)}, \theta \epsilon \{0, 1\}. \end{aligned} \tag{3.54}$$

where $E_\theta[T]$ is the expected number of samples until the SPRT terminates at either 0 or h, given H_θ.

When H_0 is true, $P(H_1)$ is the probability of false alarm, α, of a SPRT. Equation

(3.54) therefore provides the ARL to false alarm and can be simplified to

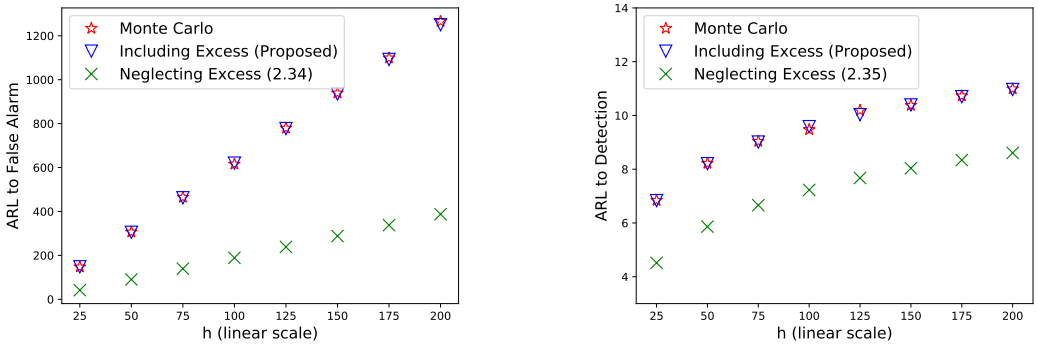
$$\text{ARL to false alarm} \;=\; \frac{E_0[T]}{\alpha}. \tag{3.55}$$

Given H_1, $P(H_1)$ is the probability of correct detection of an SPRT, $1-\beta$. Equation (3.54) then provides the ARL to detection and can similarly be simplified to

$$\text{ARL to detection} \;=\; \frac{E_1[T]}{1-\beta}. \tag{3.56}$$

The numerators of both equations can be computed via (3.8) and the denominator via (3.6). The ARL to false alarm and detection of CUSUM tests are computed for varying detection thresholds and compared to results from Monte Carlo simulation for the same observation distributions considered in Section 3.3 in Figures 3.8, 3.9, 3.10 and 3.11. In each case, the ARL values resulting from the proposed numerical computations show much closer agreement to those obtained from Monte Carlo simulation than those resulting from the use of (2.31) and (2.32), which are rooted in Wald's approximation.

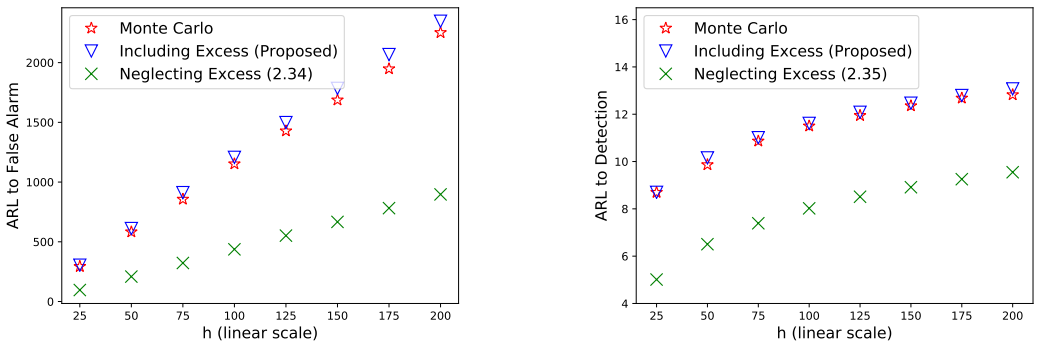
We remark that the staircase effect seen in Figures 3.10 and 3.11 is due to the discretization of the hypothesis distributions in these scenarios. This discretization causes discontinuities in the achievable ARL to false alarm and detection ranges. For these scenarios, randomization at the detector is needed to be able to achieve all possible ARL to false alarm or ARL to detection values when varying h.



(a) ARL to FA

(b) Expected Detection Delay

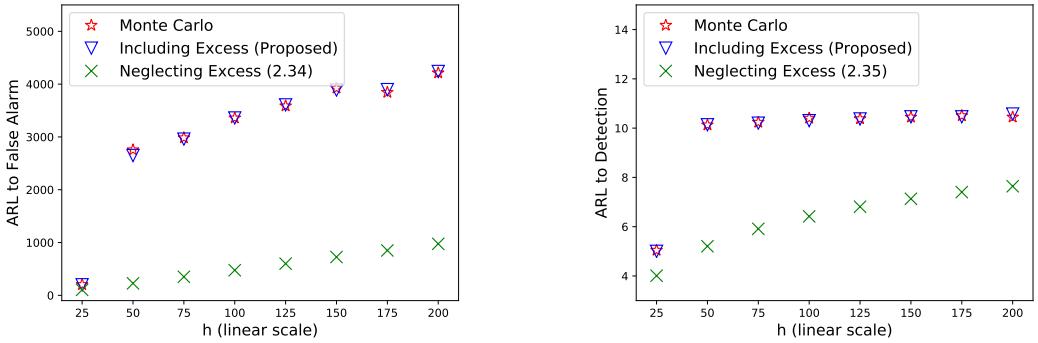
Figure 3.8: A comparison of ARL computed using the proposed method, Monte Carlo, and from (2.31) and (2.32) for a shift in mean N(0,1) vs N(1,1) CUSUM test.



(a) ARL to FA

(b) Expected Detection Delay

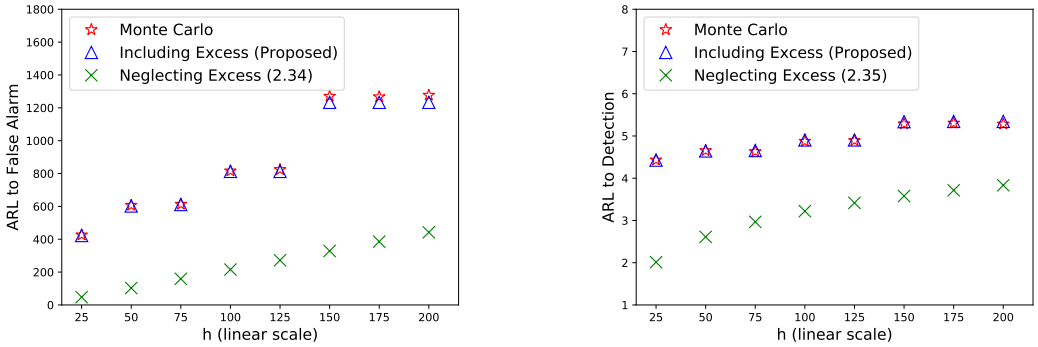
Figure 3.9: A comparison of ARL between the proposed method, Monte Carlo and from (2.31) and (2.32) for a shift in variance N(0,1) vs N(0,3) CUSUM test.



(a) ARL to FA

(b) Expected Detection Delay

Figure 3.10: A comparison of ARL between the proposed method, Monte Carlo, and from (2.31) and (2.32) for a Bernoulli p_1=0.005 vs p_0=0.2 CUSUM test.



(a) ARL to FA

(b) Expected Detection Delay

Figure 3.11: A comparison of ARL between the proposed method, Monte Carlo, and from (2.31) and (2.32) for a discrete pmf where $p_{4|H_1}(x) = 0.587\delta(x) + 0.325\delta(x-1) + 0.078\delta(x-2) + 0.011\delta(x-3)$ and $p_{4|H_0}(x) = 0.977\delta(x) + 0.023\delta(x-1) + 2.215\text{x}10^{-4}\delta(x-2) + 1.210\text{x}10^{-6}\delta(x-3)$.

3.5 Applications to Distributed Quickest Change Detection

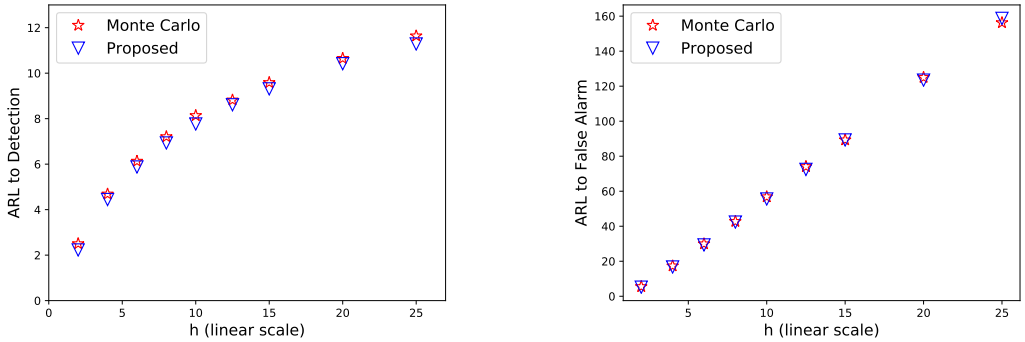
In this section, we will apply the above CUSUM analysis to two types of distributed change detection systems. The first, a centralized system, is useful when communication between local sensors and the FC is not limited. The second example considers a decentralized distributed system. Local sensors implement CUSUM procedures and forward their decisions to a FC at local test termination. We show in this example how our proposed CUSUM analysis can be extended to analyze the performance of CUSUMs that reinitialize themselves after transmitting reports to the FC. This can subsequently be used to approximate the pmfs describing the arrival of reports at the FC.

3.5.1 Centralized, Distributed Change Detection

The first distributed system we will consider is a centralized system. We assume that sensors are each located in a unique geographic location. They observe sequentially in time and immediately transmit each observation directly to the FC. The local sensors do not process these observations, instead forwarding them directly to the FC for processing. Observations at the same sensor are IID and it is assumed that observations between sensors are independent conditioned on H_0 or H_1. All sensor observations are distributed according to the same distribution, either H_0 or H_1. A number of different methods have been proposed to be implemented at the FC to process observations, but we will focus on analyzing the use of a CUSUM at the FC to make a final decision on whether a change has occurred.

Suppose there are M sensors in our distributed system and that the FC cannot take its own observations. This means that instead of the typical CUSUM discussed in

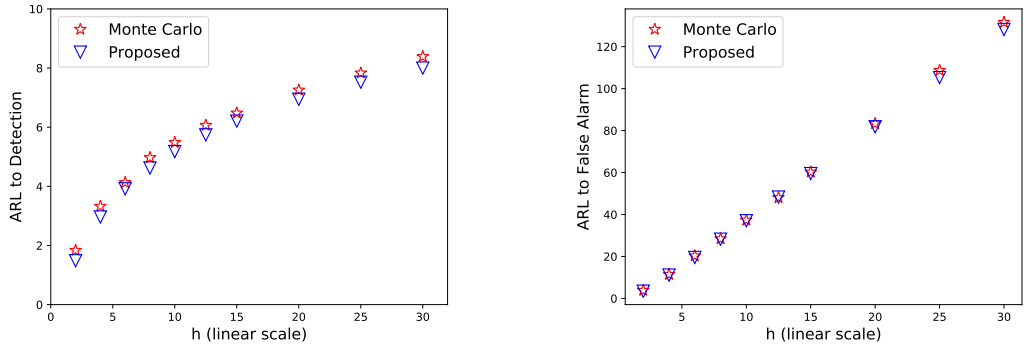
Section 3.4, where observations are observed sequentially in time, we have M observations observed by the FC simultaneously in each time slot. Due to the independence between sensors and of subsequent observations, this makes the pdf of the CUSUM statistic at time $k = 1$ equivalent to a single sensor's CUSUM statistic at time $k = M$. We can then modify the $E_\theta[T]$ terms in Equations (3.55) and (3.56) by dividing by M to obtain an approximation to the ARL of this centralized system. This procedure is performed and compared with results from Monte Carlo simulation in Figures 3.12 and 3.13.



(a) ARL to FA

(b) Expected Detection Delay

Figure 3.12: A comparison of ARL to detection between the proposed method and Monte Carlo for a centralized CUSUM with $M = 2$ under varying detection thresholds.



(a) ARL to FA

(b) Expected Detection Delay

Figure 3.13: A comparison of ARL to false alarm between the proposed method and Monte Carlo for a centralized CUSUM with $M = 3$ under varying detection thresholds.

Using this procedure, we can then numerically compute the performance of a centralized CUSUM for different values of M. Such an analysis is useful as it provides intuition of how, in the best case, the performance of a distributed system improves with increasing M. The worst case expected detection delay, E_{DD}, defined as

$$E_{DD} = E[(\tau_F - \Gamma)^+] \tag{3.57}$$

with $\Gamma = 0$, was computed numerically and plotted as a function of the false alarm rate, R_{FA}, computed by

$$R_{FA} = \frac{1}{\text{ARL to FA}}. \tag{3.58}$$

This is shown for varying number of sensors, M, in the system and and is plotted in Figure 3.14.

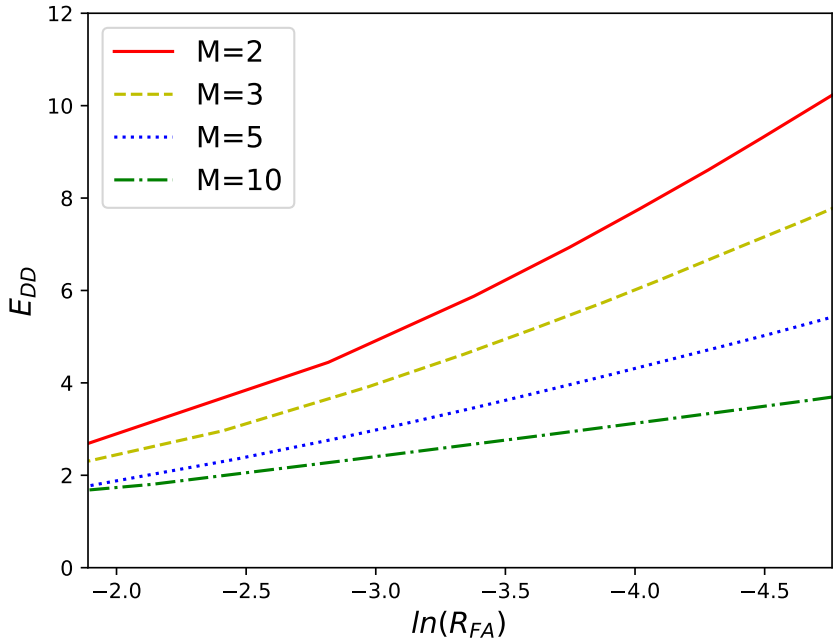


Figure 3.14: Trade-off curve for expected detection delay versus false alarm rate in a N(0,1) vs N(0.5,1) test.

3.5.2 Decentralized, Distributed Change Detection

The CUSUM analysis of Section 3.4 can also be applied to decentralized distributed systems. In such systems, local sensors implement their own detection procedure to offload processing from the FC and summarize their observations by transmitting reports periodically. The FC analyzes these reports and makes a final decision on the state of the system. When a local sensor sends a report, one strategy is for the local sensor to reinitialize itself and continue processing observations. We can extend the above process to analyze local sensor CUSUM procedures that reinitialize themselves after transmitting a report. We will focus on when the CUSUM procedure is reinitialized to zero, but a similar procedure follows for any reset strategy.

We extend the procedure proposed in Section 3.4 by accounting for the probability of reinitializing to zero after each observation. This reinitialization probability is computed by

$$\mathrm{P}(\text{reinitialize to zero at stage } k+1) = \int_b^\infty f_{k+1}(x)dx. \tag{3.59}$$

This probability is then added to the current stages pdf at $f_{k+1}(0)$ prior to observing the next observation. This therefore allows for the possibility of a local sensor's CUSUM terminating and reinitializing itself to zero to be accounted for in the subsequent stages pdf. We can then compute the probability of choosing H_1 in delay stage $k+1$ given that stage $k+1$ is reached, $\gamma_{k+1}(H_\theta)$, using Equation (3.5) for a CUSUM test that after each observation has some probability of transmitting a report and reinitializing itself to zero. This modified CUSUM analysis used to analyze the repeating CUSUM is shown in Algorithm 4.

We can then use this procedure to compute the probability of receiving reports at the FC. We define $\mathrm{P}(r_{k+1} = n|H_{\theta,k+1})$ and $\mathrm{P}(r_{k+1} \geq L|H_{\theta,k+1})$, $\theta\epsilon\{0,1\}$, as the probability of receiving n reports and the probability of receiving greater than or equal to L reports at the fusion center given hypothesis θ at time slot $k+1$, respectively. These probabilities can be found using

$$\mathrm{P}(r_{k+1} \geq L|H_{\theta,k+1}) = \sum_{n=L}^{M} \mathrm{P}(r_{k+1} = n|H_{\theta,k+1}) \tag{3.60}$$

where

$$\mathrm{P}(r_{k+1} = n|H_{\theta,k+1}) = \binom{M}{n} \gamma_{k+1}(H_\theta)^n (1-\gamma_{k+1}(H_\theta))^{M-n}. \tag{3.61}$$

Algorithm 4 Repeating CUSUM Analysis with Reinitialization to Zero

$k = 1$
while true:
    generate pdf $f(x)$
    if $k = 1$:
        $f_1(x) = f(x)$
        P(reach stage 1) = 1
        P(reach stage 2) = $\int_{-\infty}^{h} f_1(x)dx$
        P(accept H_1 at stage 1) = $\int_{b}^{\infty} f_1(x)dx$
        if $\frac{\text{P(accept } H_1 \text{ at stage 1)+P(reach stage 2)}}{\text{P(reach stage 2)}} \geq \epsilon$:
            break
    if $k \geq 2$:
        P(reset at stage k-1) = $\int_{-\infty}^{0^-} f_{k-1}(x)$
        P(reinitialize to zero at stage $k-1$)= $\int_{b}^{\infty} f_{k-1}(x)dx$
        f_{k-1}^{trunc}=truncate($f_{k-1}, 0, h$)
        $f_{k-1}^{trunc}(0) = f_{k-1}^{trunc}(0)$ + P(reset at stage k-1) + P(reinitialize to zero at stage $k-1$)
        f_k=($f_{k-1}^{trunc} \star f$)(x)
        P(reach stage $k+1$) = $\int_{-\infty}^{h} f_k(x)dx$
        P(accept H_1 at stage k) = $\int_{h}^{\infty} f_k(x)dx$
        if $\frac{\text{P(accept } H_1 \text{ at stage } k\text{)+P(reach stage } k+1\text{)}}{\text{P(reach stage } k+1\text{)}} \geq \epsilon$:
            break
    $k = k + 1$
end while
$E_\theta[T]$=sum(P(reach stage k))

Equation (3.61) can be interpreted as follows. In a system with M sensors, there is a possibility of receiving n reports simultaneously, where $n \leq M$. For a single set of sensors, the probability of receiving n reports at time k is geometrically distributed, characterized by the probability that n independent sensors choose H_1, multiplied by the probability $(M - n)$ sensors do not choose H_1. There is, however, a total of M choose n different combinations of local sensors that can provide n reports to the fusion center. Equation (3.61) accounts for all these different possible combinations. Using the fact that receiving $1, 2, ..., M$ reports are independent events, Equation (3.60) includes these terms in a summation to determine the probability of the number of received reports exceeding a specified amount, L, that is between 1 and M.

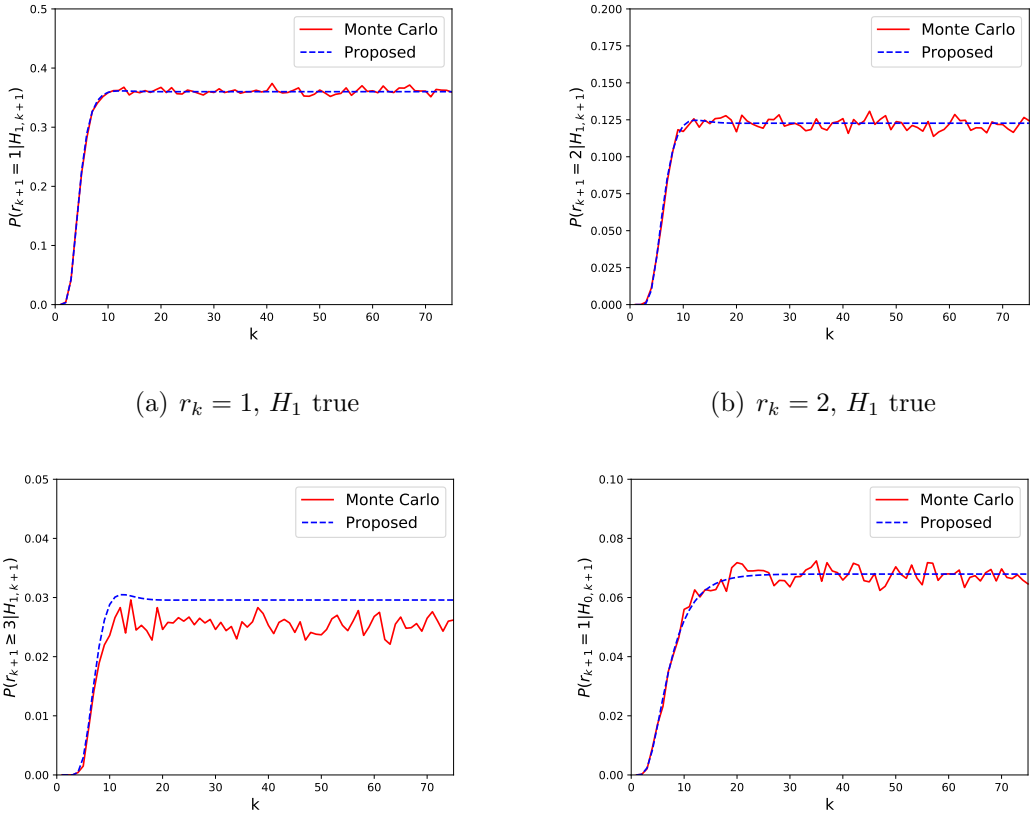
Figure 3.15 shows the accuracy of this procedure by computing the probability of receiving one, two and greater than or equal to three reports at the FC given H_1, and

the probability of receiving one report given H_0 between the proposed procedure and Monte Carlo simulation of the distributed system. These probabilities are plotted as a function of absolute time from the start of system operation at $k = 0$. The change is assumed to have happened at time $k = 0$ when H_1 is true, or to have never happened when H_0 is true. We remark that there is an initial transient period where the value of $\mathrm{P}(r_{k+1} = n|H_\theta)$ and $\mathrm{P}(r_{k+1} \geq L|H_\theta)$ is time varying, but reaches a steady state value after a short period of time. This transient period occurs both at the start of system operation and immediately after a change between H_0 and H_1 occurs. These two periods overlap in Figure 3.15 because it is assumed that the change time $\Gamma = 0$.

In general, our proposed numerical computations closely approximate the reporting probabilities of local sensors found from Monte Carlo simulation. Minor discrepancy is seen in Figure 3.15(c) because of time dependency inherent in the reporting sequence of local sensors that is being neglected in our computations. The origin of this time dependence is from the transient period at the start of local sensor operation, or immediately after reinitialization, where the probability of a local sensor transmitting a report is smallest and time varying. Although our proposed procedure captures this period at the start of system operation at $k = 0$, it does not consider the transients that occur when local sensors reinitialize mid-operation. These transients lower the probability of receiving a report immediately after one or more local sensors transmit a report until the reinitialized local sensor reaches this steady state region again.

The effect of this time dependence is greatest when considering the probability of receiving multiple simultaneous reports. To illustrate this, consider $P(r_{k+1} = 3|H_{1,k+1})$ in a system with $M = 3$. When a local sensor transmits a report, it's

test statistic is reinitialized to zero, thereby making the probability that the same local sensor immediately transmits another report very small. In this example, the probability of receiving three simultaneous reports is unlikely if one of the local sensors has recently reported and been reinitialized. The probability of receiving one report is impacted less because although one sensor may have been reinitialized, two others are still operating in their steady state region and are likely to report. We will consider conditions on when this time dependence may be smallest in Chapter 5.



(a) $r_k = 1$, H_1 true

(b) $r_k = 2$, H_1 true

(c) $r_k = 3$, H_1 true

(d) $r_k = 1$, H_0 true

Figure 3.15: A comparison between numerically computed and Monte Carlo generated $P(r_{k+1} = 1|H_{1,k+1})$, $P(r_{k+1} = 2|H_{1,k+1})$, $P(r_{k+1} \geq 3|H_{1,k+1})$ and $P(r_{k+1} = 1|H_{0,k+1})$ for a $N(0, 1)$ vs $N(0.5, 1)$ test with $h_l = \ln(15)$, $M = 13$, and $L = 3$.

3.6 Conclusion

In this chapter, a procedure for analyzing SPRT performance, originally developed in [7] for the Gaussian shift in mean problem, was extended to shift in variance, Bernoulli, and arbitrary pmf tests. Two methods of adapting this procedure to analyze the CUSUM test were proposed. These strategies allow for more accurate test

thresholds to be designed in comparison to those obtained via Wald's approximation by accounting for the overshoot that occurs within sequential tests. They also provide insight into the operation of the SPRT and CUSUM after each observation. Two applications of these procedures to the distributed change detection problem were presented.

Chapter 4

A Bayesian Approach to Quickest Change Detection in Large Sensor Networks under Bandwidth Constraints

4.1 Introduction

In this chapter, a Bayesian formulation to the distributed quickest change detection problem is investigated. Existing literature on the Bayesian distributed quickest detection problem often treat communication between local sensors and the FC at one of its two extremes. In [44], communication is highly restricted. A single report can be received at the FC in each time slot. If local sensors transmit more than one report, these additional reports are ignored by the FC. In these situations, valuable information collected by local sensors is being discarded. In large sensor networks, time slots with multiple reports may often occur, resulting in a significant waste of resources within the network. There is also potential for performance improvement by considering this additional information.

The other extreme of this problem is considered in [40]. In this system design,

local sensors immediately transmit observations to the FC for processing. Although this procedure is optimal for the FC, the requirement of all observations being visible to the FC is often too resource demanding to implement in practice within large wireless sensor networks.

We therefore investigate a decentralized solution to this problem, where a bandwidth constraint is imposed at the FC that limits the number of local sensor reports that can be received in a single time slot. Multiple reports may be received and uniquely considered in each time slot as long as they do not exceed the bandwidth constraint of the system.

The CUSUM procedure is used at local sensors to quantize local sensor observations into binary reports that are transmitted to the FC, indicating the outcome of each local sensor's CUSUM in each time slot. The numerical procedures of Chapter 3 are used to analyze the local sensor CUSUMs and generate pmfs describing the probability of the FC receiving one or more reports in each time slot.

The FC treats the discrete stream of reports as an optimal stopping problem, which, as described in Chapter 2, it solves by tracking the posterior probability of a change occurring after each observation and comparing to a pre-computed threshold. The pmfs describing the arrival of reports at the FC under hypotheses H_0 and H_1 allow for up to a limit of $L \geq 1$ different posterior update functions to be used, where L is the bandwidth constraint restricting the number of simultaneous reports received.

We investigate the performance of the proposed system when design parameters required within such a Bayesian formulation are varied. We also discuss limitations of this formulation in addressing research objectives and towards solving the distributed

change detection problem in large sensor networks.

4.2 System Model

We consider a distributed system of M sensors. These sensors observe a sequence of observations X_1^i,X_2^i,..., where i denotes the local sensor, $i\epsilon\{1, 2, .., M\}$. Each sensor is located in a unique geographic location, and the distances between sensors are large enough that no sensors observe the same sequence of observations. Observations between each sensor can therefore be considered as independent conditioned on H_0 or H_1. Observations at the same sensor are also IID in time. At some point in time, the observations at all sensors change from being generated by H_0 to H_1. It is assumed that this change occurs simultaneously for all sensors.

This change can be described for sensor i, $1 \leq i \leq M$, as

$$H_0 : \quad X_1^i, X_2^i, ..., X_{\Gamma-1}^i, \quad \sim f_0 \tag{4.1}$$

$$H_1 : \quad X_\Gamma^i, X_{\Gamma+1}^i, ..., X_k^i, \quad \sim f_1 \tag{4.2}$$

where Γ is the change time.

This change time is assumed to be random and unknown. For this formulation, we will consider Γ to be geometrically distributed because of the memoryless property of this distribution. Γ is therefore described by π_0, the probability that the change occurs before the observations are observed, and by ρ, the prior probability that a change occurs at each time slot regardless of the observation observed. The parameter ρ is specified by the physical environment, and thus the system designer has no control over it. Often this parameter can be difficult to accurately estimate ahead of time,

and therefore designing a system that is robust to different values of this parameter is desirable. Using notation from Chapter 2, the change time distribution can be expressed as

$$P(\Gamma = k) = \pi_0 I_{k=0} + \rho(1-\rho)^{k-1} I_{k\geq 1}, \tag{4.3}$$

where I_k denotes the indicator function of an event at time k and is zero otherwise.

When a local sensor decides that there has been a change in distribution, it sends a binary-valued report to the FC. The FC is considered as the global decision maker. It is in charge of making a final decision on whether a change has occurred, using information collected from local sensors. The sequence of received reports at the FC can be considered as follows: the FC observes a value of r_k equal to $1, 2, ..., M$, where a value of $r_k = j$ means that j sensors have made a decision and sent a report to the FC in the same time slot, where $0 \leq j \leq M$. If a report is not received, the value of r_k is zero. This sequence of the numbers of received reports, r_k, at times $k = 0, 1, 2...$, is used to update a cumulative statistic at the FC at each time slot, which is updated based on how many reports are received from local sensors.

Let the fusion center's expected detection delay and rate at which false alarms are generated be defined, respectively, as

$$E_{DD}^{FC} = E[(\tau_F - \Gamma)^+] \tag{4.4}$$

$$R_{FA}^{FC} = \frac{1}{\text{ARL to FA}}, \tag{4.5}$$

where τ_F is the FC's decision time, Γ is the time at which a change occurs, and $(x)^+ = \max(x, 0)$. The goal of the FC is to obtain a decision policy, δ, with minimum

expected delay subject to the following false alarm constraint:

$$\min_{\delta} \qquad E_{DD}^{FC} \tag{4.6}$$

$$\text{subject to} \quad R_{FA}^{FC} \leq \alpha_{FC}. \tag{4.7}$$

In this formulation, α_{FC} is the desired maximum rate of false alarms.

In our analysis of the proposed systems, we will assume the change time occurs at $\Gamma = 0$ under H_1 and does not occur under H_0. We also assume that local sensors are only able to have one-way communication with the fusion center. Communications among local sensors is also not possible. The communication between local sensors and the fusion center is assumed to be error-free unless specified otherwise.

4.3 Local Sensing Algorithm

In order to design distributed, decentralized systems, we must consider the design of both the FC and of local sensors to ensure our false alarm rate constraint is satisfied. Local sensors observe observations sequentially in time. In order for the global system to minimize the delay in detecting the change in distributions, local sensors must also utilize an optimal decision policy.

Due to its optimality properties discussed in [24], and its simplicity by possessing a single parameter, the CUSUM algorithm is proposed to be used at local sensors. This local strategy involves tracking the likelihood ratio for all observations using

$$S_k = \max_{1 \leq j \leq k} \sum_{i=j}^{k} \ln(\frac{f_1(X_i)}{f_0(X_i)}), \tag{4.8}$$

until the test statistic S_k exceeds local CUSUM threshold h_l. If the test statistic

exceeds this threshold at time k, a binary report is generated by local sensor i and transmitted to the fusion center,

$$r_k^i = \begin{cases} 1 & S_k \geq h_l \\ 0 & S_k < h_l. \end{cases} \tag{4.9}$$

The generation of one bit local sensor reports acts as a method of quantizing the observations accumulated by local sensors into indicators of whether a local sensor CUSUM has terminated at each time k.

When a local sensor's CUSUM terminates and a report is transmitted, the local sensor statistic, S_k, is reinitialized to zero. We remark that reinitialization to $S_k = 0$ implies that H_0 is favoured a priori, while reinitialization to $S_k = h_l$ implies that H_1 is favoured a priori. We justify reinitializing to $S_k = 0$ because it is the most conservative strategy for detecting a change among all other possible values. This will result in the smallest possible FC false alarm rate and the largest FC detection delay among all other possible resetting strategies.

We also note that reinitialization to $S_k = 0$ creates a union of likelihood ratio tests for all possible change times, and by extension, a union of CUSUM tests. This allows us to guarantee a lower bound on the ARL to FA when designing local sensors.

4.4 Fusion Center Algorithm

At the fusion center, local sensor reports are received sequentially in time. At each time slot, the fusion center can receive one of {0,1,2..,M} local sensor reports, where M is the number of local sensors in the system. When the report generation policy at local sensors is fixed, the number of reports arriving at the FC can be described

by distinct pmfs under H_1 and H_0.

For this problem, we consider the case where there are bandwidth constraints preventing $r_{k+1} \geq L$ reports from being received at the FC in a single time slot, thus requiring the simultaneous arrival of L to M reports to be treated as a single event. This type of bandwidth constraint is realistic for large sensor networks, where the number of reports received in a single time slot could be very large, exceeding the communication bandwidth to the FC. In the case where bandwidth is not the limiting factor, the value of L can be set equal to M, so that every possible report quantity can be considered.

In Section 3.5.2, a repeating CUSUM with reinitializion to zero after choosing H_1 was analyzed. A procedure was proposed that allows us to compute the probability of receiving n reports and the probability of receiving greater than or equal to L reports in a distributed system at the FC for a given time slot. It was however noted that these probabilities are time varying both at the start of system operation and immediately after a change in distribution occurs, but reach steady state values after a short period of time.

We propose that the procedure of Section 3.5.2 is used to compute the pmfs describing the arrival of reports at the FC and assume that the steady state probabilities computed using (3.60) and (3.61) are valid in describing the local sensor reporting probability of the system being considered. This assumption is applicable when the system is operating before, or a period of time after, a change in distribution has occurred, but not during the time period encompassing the change time. This allows us to avoid the possibility of a set of time varying report pmfs and assume constant reporting probabilities. We can then drop time k from (3.60) and (3.61) in computing

the report arrival pmfs under H_1 and H_0.

By fixing our local sensor strategy and associating them with time invariant reporting pmfs, we can follow existing literature in treating the quickest change detection problem at the FC as an optimal stopping problem. As discussed in Chapter 2.3.2, this can be solved by tracking the posterior probability that a change has occurred at each time slot [31]. We now generalize this procedure to account for multiple simultaneous reports with bandwidth constraints at the FC.

Let π_{k+1} denote the posterior probability that a change occurred at the $(k+1)^{th}$ time slot, given by

$$\begin{aligned} \pi_{k+1} &= P(H_1|r_{k+1}) && (4.10) \\ &= P(\Gamma \leq k+1|F_{k+1}), && (4.11) \end{aligned}$$

where F_{k+1} denotes the history of the reporting sequence up to the $(k+1)^{th}$ time slot. Based on Bayes rule, π_{k+1} can be computed recursively by the following formula:

$$\pi_{k+1} = \begin{cases} \Phi^0(\pi_k) & \text{if } r_{k+1} = 0 \\ \Phi^n(\pi_k, r_{k+1}) & \text{if } r_{k+1} = n,\ 1 \leq n \leq L-1 \\ \Phi^L(\pi_k, r_{k+1}) & \text{if } r_{k+1} \geq L,\ L \leq M. \end{cases} \quad (4.12)$$

Update function $\Phi^0(\pi_k)$ is used when a report is not received at the FC in a time slot. It updates the posterior probability based on the assumed prior probability that a change occurs in each time slot, ρ. It can be computed by

$$\Phi^0(\pi_k) = \pi_k + (1-\pi_k)\rho. \quad (4.13)$$

Functions $\Phi^n(\pi_k, r_{k+1})$ and $\Phi^L(\pi_k, r_{k+1})$ are used to update the posterior probabilities when $1 \leq n < L$ and $n \geq L$ reports are received in a time slot, respectively. They are computed using Bayes theorem as follows:

$$\begin{aligned}\Phi^n(\pi_k, r_{k+1}) &= \frac{P(H_1)P(r_{k+1}=n|H_1)}{P(r_{k+1}=n)} \\ &= \frac{P(H_1)P(r_{k+1}=n|H_1)}{P(r_{k+1}=n, H_1)+P(r_{k+1}=n, H_0)} \\ &= \frac{\Phi^0(\pi_k)P(r_{k+1}=n|H_1)}{\Phi^0(\pi_k)P(r_{k+1}=n|H_1)+(1-\Phi^0(\pi_k))P(r_{k+1}=n|H_0)}\end{aligned} \tag{4.14}$$

and

$$\Phi^L(\pi_k, r_{k+1}) = \frac{\Phi^0(\pi_k)P(r_{k+1}\geq L|H_1)}{\Phi^0(\pi_k)P(r_{k+1}\geq L|H_1)+(1-\Phi^0(\pi_k))P(r_{k+1}\geq L|H_0)} \tag{4.15}$$

respectively, where $P(H_\theta) = \Phi^0(\pi_k)$, $\theta \in \{0,1\}$, is the prior probability of H_θ at the $(k+1)^{th}$ time slot, and $P(r_{k+1} \geq L|H_\theta)$ and $P(r_{k+1}=n|H_\theta)$ are given by the steady state values of Equations (3.60) and (3.61),respectively. The procedure used by local sensors and the FC is summarized in Algorithm 5.

Algorithm 5 Bayesian FC and Local Sensor Algorithm

```
while π_k < π*:
    for i=1 to M:
        g^i_{k+1}=g^i_k+ln(f_1(X_k)/f_0(X_k))
        if g^i_{k+1} ≥ h_l:
            r_{k+1} = r_{k+1} + 1
        if g^i_{k+1} < 0:
            g^i_{k+1} = 0
    end for
    π_{k+1} = Φ^{r_{k+1}}(π_k, r_{k+1})
    k = k + 1
end while
```

FC Threshold Design Procedure (Offline)

An offline procedure is used by the FC to determine the best global threshold, π^*, to use for a given number of local sensors M, detection threshold h_l, and false alarm rate constraint α_{FC}. Without loss of generality, for this analysis we will consider the case where $L = M$ reports can be uniquely received and considered at the fusion center at any time.

Consider again the FC design goal given by (4.6). For a given α_{FC}, the total expected cost of this problem can be expressed as

$$R(\delta) = R_{FA}^{FC} + c_d E_{DD}^{FC}, \tag{4.16}$$

where c_d is a non negative constant that denotes the relative cost of a delay in correct detection compared to a false alarm. The value of this constant is set to achieve a desired system false alarm rate. Since the Bayes risk $R(\delta)$ is additive over time, minimization of (4.16) can be done using the following approach:

At each time slot, we define the optimal cost $J_{opt}^*(F_k)$, where $J_{opt}^*(F_k)$ is determined by choosing the action that provides the minimum cost at each time slot. In our model, the decision the fusion center can make at each time slot is to continue or stop the test. We define this optimal cost as

$$J_{opt}^*(F_k) = \min(J_{stop}(F_k), J_{cont}(F_k)). \tag{4.17}$$

For each time k, this optimal cost $J_{opt}^*(F_k)$ can be written as a function of only π_k, $J_{opt}^*(\pi_k)$, and the optimal stopping policy can be restricted to a class of decision functions that only depend on π_k [17].

We define the cost incurred by stopping, $J_{stop}(\pi_k)$, as the uncertainty in deciding that a change has occurred. This is expressed as

$$J_{stop}(\pi_k) = 1 - \pi_k. \tag{4.18}$$

The cost of continuing is defined as the cost of taking an additional sample and then choosing the optimal decision. This is expressed as

$$J_{cont}(\pi_k) = c_d \pi_k + E[J^*_{opt}(\pi_{k+1})], \tag{4.19}$$

where

$$E[J^*_{opt}(\pi_{k+1})] = P(r_{k+1} = 0) J^*_{opt}(\Phi^0(\pi_k)) + \sum_{q=1}^{M} P(r_{k+1} = q) J^*_{opt}(\Phi^q(\pi_k, r_{k+1})), \tag{4.20}$$

and

$$P(r_{k+1} = q) = \pi_k P(r_{k+1} = q | H_{1,k+1}) + (1 - \pi_k) P(r_{k+1} = q | H_{0,k+1}). \tag{4.21}$$

In (4.19), $c_d \pi_k$ is the detection delay cost associated with taking another observation. $E[J^*_{opt}(\pi_{k+1})]$ is the expected cost of choosing the optimal decision at the next time slot, where the possibility of receiving $r_k = 0, 1, 2, ..., M$ reports is considered with their associated posterior update functions from Equation (4.12).

It can be shown that both $J_{stop}(\pi_k)$ and $J_{cont}(\pi_k)$ are nonnegative concave functions on the interval [0,1], with $J_{stop}(1)$=$J_{cont}(1)$=0 [40]. Under these assumptions, there is an optimal strategy at the fusion center which solves the above minimization

problem by applying the decision rule

$$\phi^B(\pi_k) = \begin{cases} \geq \pi^* & \text{decide change occurs at time } k \\ \text{else} & k \leftarrow k+1. \end{cases} \tag{4.22}$$

where the optimal threshold at the fusion center, π^*, is found from the solution to

$$J_{stop}(\pi_k) = J_{cont}(\pi_k). \tag{4.23}$$

In order to compute the optimal threshold for this problem, the cost functions J_{stop} and J_{cont} must be computed for different possible π_k values. This was done by partitioning the range of the π_k into 1000 points, ie. $\{0, 0.001, 0.002, ..., 1\}$. For each π_k value, the cost of stopping was computed using Equation (4.18) . The cost of continuing was computed by recursively computing (4.19) and (4.20) until a maximum of 1000 iterations had been performed. Due to the infinite horizon nature of this problem, it is necessary to truncate the number of delay stages considered. This truncation point was chosen to be 1000 based on experimental results for the scenarios considered. The iterations terminate and set a value for $E[J^*_{opt}(\pi_{k+1})]$ when the posterior $\pi_k = 1$, or if all 1000 iterations have been performed [6]. In this case, $1 - \pi_k$ is returned. The optimum cost for each π_k was then computed by (4.17). This strategy is summarized in Algorithm 6.

Algorithm 6 Dynamic Programming Algorithm for determining π^*

```
for all π_k:
    if E[J*_opt(π_{k+1})] has recursively been called > 1000 times:
        return 1-π_k
    if π_k = 1:
        return 1-π_k
    J*_opt(π_k)=min(1-π_k, c_d π_k + E[J*_opt(π_{k+1})])
end for
```

Once the optimal threshold π^* is determined from the intersection of $J_{stop}(\pi_k)$ and $J_{cont}(\pi_k)$, the performance metrics R_{FA}^{FC} and E_{DD}^{FC} can be computed for given constant c_d. These quantities may be obtained for each value of c_d via Monte Carlo simulation. Pairs of $(R_{FA}^{FC}, E_{DD}^{FC})$ for this optimal strategy can then be obtained by varying c_d, obtaining the corresponding optimal threshold π^*, and performing these Monte Carlo simulations, thus generating a trade-off curve for the optimal policy. This strategy is described in [39],[40].

4.5 Results

The performance of the system proposed in this chapter is evaluated under varying conditions. In doing this, the effect of design variables on system performance is investigated. System performance is defined to be the expected detection delay achieved given a guaranteed FC false alarm rate.

First, the effect of the number of sensors, M, on system performance is considered. Figure 4.1 shows a $N(0,1)$ vs $N(0.5,1)$ test with $L = 3$, $\rho = 0.001$ and $h_l = \ln(10)$. M is increased by increments of four, from $M = 3$ to $M = 15$. Monte Carlo simulations with 10^5 trials are performed using Algorithm 5 with varying π^* between $(0,1)$ to determine $(R_{FA}^{FC}, E_{DD}^{FC})$ pairs. As is expected in distributed systems, increasing M results in improved system performance. We remark though that this improvement is smaller than what would be expected from adding redundancy in the from of additional sensors in a distributed system.

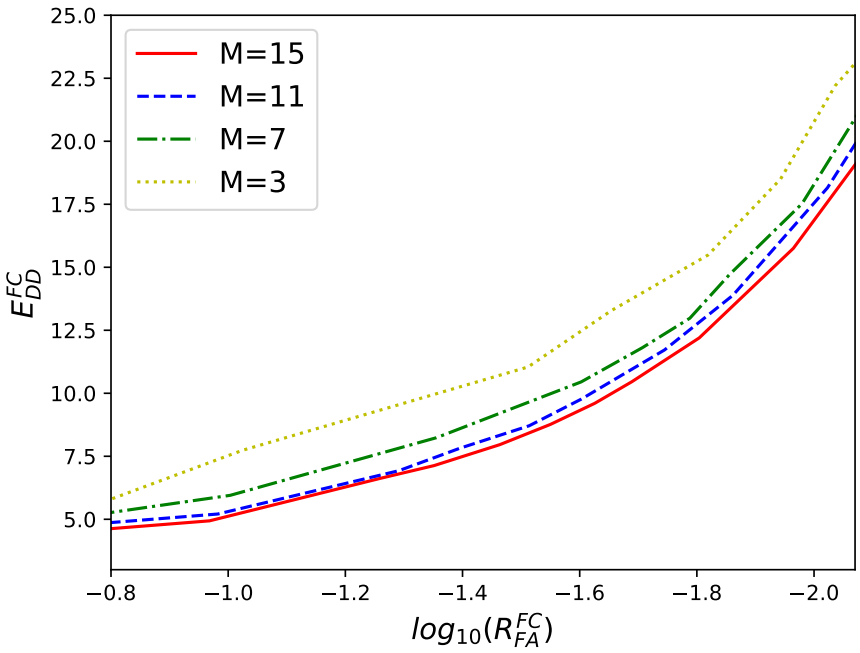


Figure 4.1: Performance of the proposed distributed system observing a $N(0,1)$ vs $N(0.5,1)$ test with $h_l = \ln(10)$, $\rho = 0.001$, $L = 3$ and 10^5 Monte Carlo trials. The number of systems in the system, M, is varied between $M = 3$ to $M = 15$.

Figure 4.2 shows the effect of bandwidth constraint, L, on system performance. To exemplify this, the mean of H_1 in the shift in mean Gaussian test is increased to $N(0,1)$ vs $N(1,1)$. This is done to examine the case where reports are sent frequently under H_1, thereby increasing the likelihood of receiving multiple reports simultaneously. The system design variables are chosen to be $M = 11$, $h_l = \ln(10)$, $\rho = 0.001$ and L varied from $L = 1$ to $L = 3$. Similar to the previous example, Algorithm 5 is used within a Monte Carlo simulation of 10^5 trials to generate the $(R_{FA}^{FC}, E_{DD}^{FC})$ pairs for each π^* tested.

The performance of the distributed system improves with increasing L. The improvement from $L = 1$ to $L = 2$ is significant because of the high likelihood of receiving more than one report in each time slot for this example. Further improvement can be seen from $L = 2$ to $L = 3$. We note that this improvement is smaller because there is a lower likelihood of receiving three reports simultaneously and taking advantage of this additional bandwidth.

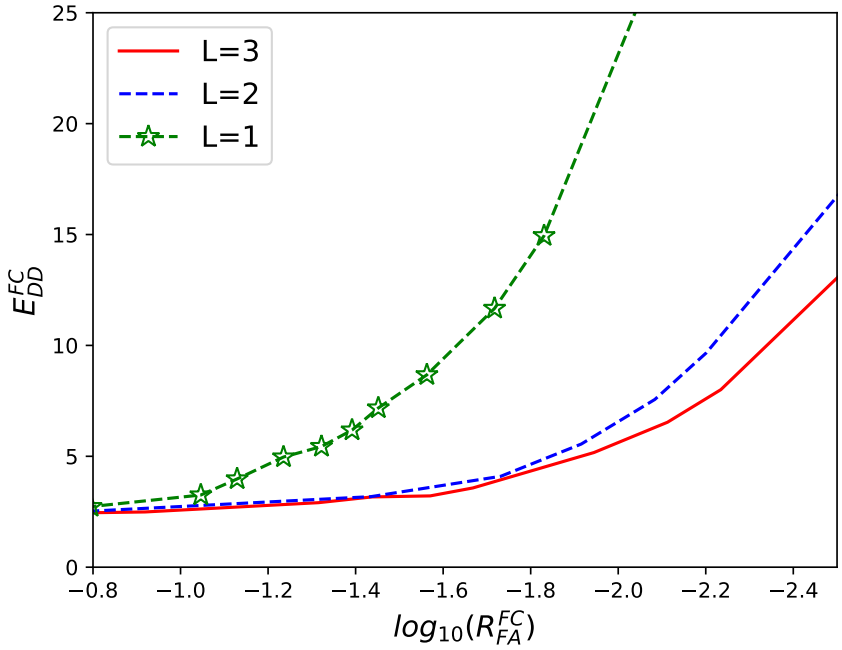


Figure 4.2: Performance of the proposed distributed system observing a $N(0, 1)$ vs $N(1, 1)$ test with $h_l = \ln(10)$, $\rho = 0.001$, $M = 11$ and 10^5 Monte Carlo trials. The bandwidth constraint, L, is varied from $L = 1$ to $L = 3$.

Next, the performance of the system is evaluated under different values of ρ. As discussed in Section 4.2, estimating ρ can often be difficult in practice. A system designer may determine local sensor and FC thresholds to maintain a desired error rate under an assumed value of ρ that may differ when operating under real world

conditions. We therefore explore the impact of ρ on the false alarm rate and expected detection delay of the distributed system for fixed threshold pairs (h_l, π^*).

Table 4.1 shows R_{FA}^{FC} and E_{DD}^{FC} for a $N(0,1)$ vs $N(1,1)$ test with $M = 11$, $L = 3$ and $h_l = \ln(10)$ with varying assumed ρ values. The FC threshold, π^*, is held fixed for each possible value of ρ. Both R_{FA}^{FC} and E_{DD}^{FC} are affected by the assumed value of ρ. In particular, the effect of ρ on R_{FA}^{FC} makes it difficult for a system designed to guarantee a specific false alarm rate unless near exact knowledge of ρ is known prior to designing FC and local sensor thresholds.

	R_{FA}^{FC}			E_{DD}^{FC}		
π^*	$\rho = 0.01$	$\rho = 0.001$	$\rho = 0.0005$	$\rho = 0.01$	$\rho = 0.001$	$\rho = 0.0005$
0.05	0.194	0.0484	0.0367	2.429	2.909	3.157
0.25	0.0624	0.0269	0.0230	2.5997	3.2144	3.271
0.5	0.0400	0.0215	0.0186	3.1413	3.574	4.0667
0.75	0.0295	0.0177	0.0158	3.234	4.2235	4.291
0.99	0.0159	0.0114	0.0106	4.332	5.1707	5.1976

Table 4.1: Values of R_{FA}^{FC} and E_{DD}^{FC} computed for a $N(0,1)$ vs $N(1,1)$ test using fixed FC threshold π^*, M, h_l and L. The change parameter, ρ, is varied to show the difference in false alarm rate and expected detection delay performance caused by different values of ρ.

4.6 Limitations

Although this chapter proposes aspects of a process by which we can extend the optimal stopping approach to quickest change detection explored in literature to distributed, decentralized systems, a number of design challenges limit the ability of using this system to meet the objectives that motivate this work. We recall these goals from Section 1.3; we seek to propose a design methodology by which we can determine local sensor and FC test thresholds in order to satisfy desired error constraints, and to be capable of analyzing the performance of our distributed, decentralized system. The limitations discussed next limit the feasibility of using a Bayesian formulation towards, in practice, addressing the distributed quickest change detection problem.

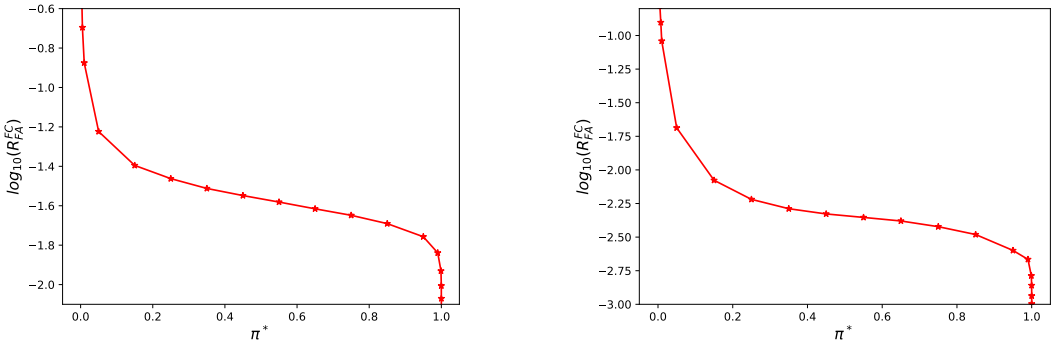
Operating Range of a Set of Thresholds

The interdependence between local sensor false alarm rate and FC false alarm rate creates one challenge in proposing design methodologies for this system. Our choice of local sensor threshold h_l in this system dictates which FC false alarm rates can be physically realized. In other words, for a given value of h_l, we are limited in what R_{FA}^{FC} we can design our system to achieve with any choice of π^*.

This can be illustrated by investigating the global FC false alarm rate as a function of π^* with fixed h_l. Figure 4.3 shows a $N(0,1)$ vs $N(0.5,1)$ test with $\rho = 0.001$, $M = 3$, and $L = 3$. In Figure 4.3(a), the value of h_l is fixed to $h_l = \ln(3)$ and in Figure 4.3(b) to $h_l = \ln(47)$. The offline procedure of Algorithm 6 is performed for values of π^* between $[0,1]$ and then Algorithm 5 is performed to determine the corresponding value of R_{FA}^{FC} for each value of π^*.

As seen in these figures, an operating range exists of potential values of R_{FA}^{FC}

that be physically specified by varying π^* between $[0, 1]$ for any fixed value of h_l. In general, increasing h_l results in an operating range that contains smaller values of R_{FA}^{FC}. Given a value of h_l, we can therefore only design the FC to achieve a fixed range of R_{FA}^{FC} values. This relationship makes an appropriate choice of h_l, prior to designing the FC, a necessity when seeking to propose a distributed system design that will satisfy false alarm requirements.



(a) $h_l = \ln(3)$ (b) $h_l = \ln(47)$

Figure 4.3: Fusion Center false alarm rate as a function of π^* for fixed local sensor threshold h_l.

Computational Complexity in Large Sensor Networks

Significant computational resource involved in treating the arrival of reports at the FC as an optimal stopping problem presents another limitation to this formulation. This is especially evident when designing large sensor networks and can be seen through investigation of Equation (4.20). The ability to determine an optimal FC threshold, π^*, relies on the recursive computation of Equations (4.17), (4.19) and (4.20), where the optimal cost function of (4.17) has real valued arguments. This requires $J_{cont}(\pi_k)$ and $J_{stop}(\pi_k)$ to be computed for every possible value of π_k. We also remark that the

number of terms in (4.20) is equal to $(M+1)$, which results in the number of function calls needed to compute $J_{opt}(\pi_k)$ to increase exponentially as a function of $(M+1)$, as seen in line 6 of Algorithm 6. This makes designing a suitable π^* for distributed systems with large M challenging in practice.

Performance Sensitivity to ρ

We also recall that, as investigated in Table 4.1 of Section 4.5, the performance of our system is sensitive to the assumed value of change parameter ρ. This sensitivity is rooted in the recursive use of ρ in the updating of π_k after each observation in (4.12). Given that ρ is often difficult to accurately estimate, the physically realized performance of our system could significantly change from what is simulated based on mismatch between the value of ρ that we expect and that which occurs, in practice, in our physical environment.

Necessity of Monte Carlo Simulations

Finally, we emphasize the necessity of Monte Carlo simulation in order to generate $(R_{FA}^{FC}, E_{DD}^{FC})$ trade-off curves that describe the performance of the optimal decision policy of the proposed system. Without doing this, it is not possible to associate a value of π^* with its accompanying R_{FA}^{FC} under the given system conditions. This presents limitations on the flexibility of designing this system to meet low false alarm rates.

4.7 Conclusion

In this chapter, we investigated a Bayesian system design to the distributed, decentralized quickest change detection problem. Local sensors implemented CUSUMs that reinitialize to zero after transmitting reports. These repeating CUSUMs were analyzed using the procedures developed in Chapter 3, to characterize pmfs describing the arrival of reports at the FC. These pmfs were subsequently used to implement a Bayesian form of decision making at the FC.

Although using an optimal stopping approach to processing the reporting sequence from local sensors is optimal, in practice, the system designer is faced with a number of limitations and challenges when implementing this type of system. The design challenges discussed within this chapter motivate the investigation of an alternative formulation in Chapter 5 to this problem in order to address our research objectives.

Chapter 5

A Minimax Approach to Quickest Change Detection in Large Sensor Networks under Bandwidth Constraints

5.1 Introduction

Motivated by the limitations of the system discussed in Chapter 4, we propose an alternative distributed, decentralized system design based on the minimax formulation to the quickest change detection problem.

A number of distributed system designs have been proposed in literature based on a minimax formulation to the quickest change detection problem, however many of these designs do not scale well to large wireless sensor networks. In [34], examples of different system designs are summarized and their performance compared. A centralized CUSUM is used as an upper-bound on the performance that can be achieved in these distributed systems. Another design considered has local sensors forward their quantized observations to a FC, which runs a global CUSUM. Local sensors transmit all observations in this design, quantized to a binary value. A limitation of these two

designs are that they may be too resource demanding for implementation in large wireless sensor networks because of restrictions on sensor battery power and available wireless spectrum. It is therefore desirable for significant offloading of processing to local sensors.

Decentralized system designs are also considered in this comparative work. These designs use CUSUMs at local sensors and different fusion strategies at the FC based on minimal and maximal combining of decision reports, such as those in [36] and [23]. These designs however tend to not be very efficient or do not perform as well for moderate false alarm rates typical of practical applications [34] [2].

The design of local sensors in these systems also do not consider overshoot that occurs within these tests. In large systems, the accumulated impact of overshoot on system performance can be significant. In [47], overshoot is addressed in the distributed hypothesis testing problem by encoding it into a time delay between the sampling time of local sensors and the transmission time. The FC decodes this time delay to determine the amount of overshoot in each transmission. Such a procedure, however, will inevitably impact system delay performance. Quantization strategies, similar to those considered in [35], could be used to include overshoot in local sensor decision reports, however the trade-off between improved performance and increased power consumption of such strategies has to be carefully managed. Novel ways of accounting for overshoot that occurs in local sensor tests is therefore an active research area.

Another limitation of existing work is a lack of design methodologies for determining appropriate FC and local sensor thresholds, where needed. Given a desirable tolerance for errors, it is often unclear how one can design a system to meet such a

constraint.

The system proposed in this chapter therefore seeks to address some of these limitations in existing literature. The CUSUM procedure is used at local sensors to quantize local sensor observations into binary reports that are transmitted to the FC, indicating the outcome of each local sensor's CUSUM in each time slot. The CUSUM analysis procedures proposed in Chapter 3 are used to generate pmfs describing the probability of the FC receiving one or more reports from local sensors in each time slot. The FC uses the report arrival pmfs to implement a globally run CUSUM.

Motivated by our intended application to large wireless sensor networks, we allow multiple reports to be received at the FC simultaneously and consider the case of bandwidth restrictions that limit the number of reports that can be received and considered uniquely in each time slot. We also propose a methodology by which global and local thresholds can be chosen to meet a desired false alarm rate constraint. The performance of the proposed system is evaluated for different numbers of sensors, M, and under varying bandwidth constraints, L, $1 \leq L \leq M$.

5.2 System Model

We consider a system of M sensors within our distributed system. These sensors observe a sequence of observations X_1^i,X_2^i,..., where i denotes the local sensor, $i\epsilon\{1, 2, .., M\}$. Each sensor is located in a unique geographic location, and the distance between sensors is large enough that no sensors observe the same sequence of observations. Observations between each sensor can therefore be considered as independent conditioned on H_0 or H_1. Observations at the same sensor are IID. At some point in time, the observations at all sensors change from being generated by H_0 to

H_1. It is assumed that this change occurs simultaneously for all sensors.

This change can be described as

$$H_0 : \quad X_1^i, X_2^i, ..., X_{\Gamma-1}^i, \quad \sim f_0 \tag{5.1}$$

$$H_1 : \quad X_\Gamma^i, X_{\Gamma+1}^i, ..., X_k^i, \quad \sim f_1 \tag{5.2}$$

where Γ is the change time. Γ is considered as an unknown but deterministic quantity. Unlike in Chapter 4, there is no prior assumption needed to characterize the change time distribution.

When a local sensor decides that there has been a change in distribution, it sends a binary-valued report to the FC. The FC is considered as the global decision maker. It is in charge of making a final decision on whether a change has occurred, using information collected from local sensors. The sequence of received reports at the FC can be considered as follows: the FC observes a value of r_k equal to $1, 2, ..., M$, where a value of $r_k = j$ means that j sensors have made a decision and sent a report to the FC in the same time slot, where $0 \leq j \leq M$. If a report is not received, the value of r_k is zero. This sequence of the numbers of received reports, r_k, at times $k = 0, 1, 2...$, is used to update a cumulative statistic at the FC at each time slot, which is updated based on how many reports are received from local sensors.

Let the fusion center's expected detection delay and rate at which false alarms are generated be defined, respectively, as

$$E_{DD}^{FC} = E[(\tau_F - \Gamma)^+] \tag{5.3}$$

$$R_{FA}^{FC} = \frac{1}{\text{ARL to FA}}, \tag{5.4}$$

where τ_F is the FC's decision time, Γ is the time at which a change occurs, and $(x)^+ = \max(x, 0)$. The goal of the FC is to obtain a decision policy, δ, with minimum delay subject to the following false alarm constraint:

$$\min_{\delta} \quad E_{DD}^{FC} \tag{5.5}$$

$$\text{subject to} \quad R_{FA}^{FC} \leq \alpha_{FC}, \tag{5.6}$$

where α_{FC} is the desired rate of false alarms.

In our analysis, we will assume the change time occurs at $\Gamma = 0$ under H_1 and does not occur under H_0. We also assume that local sensors are only able to have one-way communication with the fusion center. Communications among local sensors is also not possible. The communication between local sensors and the fusion center is assumed to be error-free unless specified otherwise.

5.3 Local Sensing Algorithm

The procedure implemented by local sensors will follow that of the system discussed in Chapter 4. We refer to Section 4.3 for details on local sensor operation and justification for its reinitialization strategy. In summary, local sensors will each employ a CUSUM procedure. A binary report, r_k, is generated by a local sensor and transmitted to the FC when test statistic, S_k, exceeds its local detection threshold. Local sensors will be reinitialized to zero after transmitting a report and continue to process observations until a final decision is made by the FC.

5.4 Fusion Center Algorithm

At the fusion center, local sensor reports are received sequentially in time. At each time slot, the fusion center can receive one of {0,1,2..,M} local sensor reports, where M is the number of local sensors in the system. When the report generation policy at local sensors is fixed, the numbers of reports arriving at the FC can be described by distinct pmfs under H_1 and H_0.

For this problem, we consider the case where there are bandwidth constraints preventing $r_{k+1} \geq L$ reports from being received at the FC in a single time slot, thus requiring the simultaneous arrival of L to M reports to be treated as a single event. This type of bandwidth constraint is realistic for large sensor networks, where the number of reports received in a single time slot could be very large, exceeding the communication bandwidth to the FC. In the case where bandwidth is not the limiting factor, the value of L can be set equal to M, so that every possible report quantity can be considered.

Similar to the system of Chapter 4, we refer back to the procedure discussed in Section 3.5.2 of analyzing a repeating CUSUM that reinitializes to zero after choosing H_1. We again use this procedure to compute the probability of receiving n reports and the probability of receiving greater than or equal to L reports in each time slot. This allows us to determine the pmf describing the arrival of reports at the FC, defined as $R_\theta(X_k)$, under H_1 and H_0.

We also refer back to the system operation assumptions discussed in Section 4.4. In summary, the steady state reporting probabilities computed for a system design using Equations (3.60) and (3.61) can be considered valid a short period of time after the start of system operation, and either before, or some time after, a change in

distribution has occurred. Immediately following a change, the report arrival pmfs, $R_0(X_k)$ and $R_1(X_k)$, become time varying for a short period of time. This contradicts the abrupt change model between H_0 and H_1 used in the classic CUSUM problem. Although the change between H_0 and H_1 in this problem is abrupt at local sensors, the change between $R_0(X_k)$ and $R_1(X_k)$ at the FC is gradual after the change point. In order to avoid time varying report pmfs within our FC CUSUM procedure, we assume that the FC is always operating in this steady state region, before or a period of time after a change has occurred.

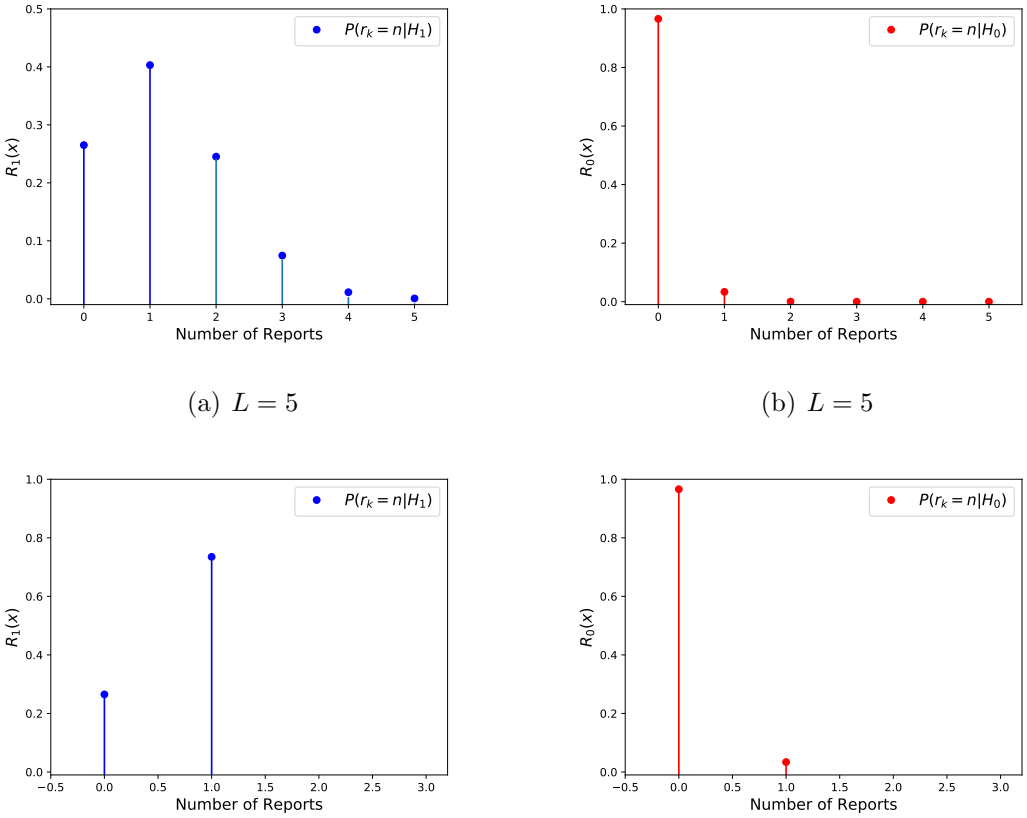
This allows us to bypass this transient period and neglect the time dependence in Equations (3.60) and (3.61), allowing us to assume constant pmfs under H_1 and H_0. We can therefore express the report arrival pmfs as

$$R_\theta(X_k) = \begin{cases} P(r_{k+1} = 0|H_\theta) & \text{if } X_k = 0 \\ P(r_{k+1} = n|H_\theta) & \text{if } X_k = n, \text{ for } n = 1, 2, .., L-1 \\ P(r_{k+1} \geq L|H_\theta) & \text{if } X_k = L, \text{ for } n = L, L+1, .., M. \end{cases} \tag{5.7}$$

When there is a bandwidth constraint of $L = 1$ on our system, this simplifies to a Bernoulli distribution

$$R_\theta(X_k) = \begin{cases} P(r_{k+1} = 0|H_\theta) & \text{if } X_k = 0 \\ P(r_{k+1} \geq 1|H_\theta) & \text{if } X_k = 1. \end{cases} \tag{5.8}$$

An example of a set of report arrival distributions for a system with $M = 5$ observing a $N(0, 1)$ vs $N(1, 1)$ test is shown in Figure 5.1 for $L = 1$ and $L = 5$.



(a) $L = 5$ (b) $L = 5$

(c) $L = 1$ (d) $L = 1$

Figure 5.1: Numerically computed pdfs given H_1 and H_0 characterizing the arrival of reports for a $N(0, 1)$ vs $N(1, 1)$ test with $h_l = \ln(25)$ and $M = 5$. A bandwidth constraint of $L = 5$ and $L = 1$ is used.

We can use these pmfs to implement a CUSUM at the FC, where the global statistic is denoted

$$z_k = \max_{1 \le j \le k} \sum_{i=j}^{k} \ln\left(\frac{R_1(X_i)}{R_0(X_i)}\right), \tag{5.9}$$

and the FC makes the decision that a change has occurred when this statistic exceeds FC detection threshold h_f. This procedure is shown in Algorithm 7.

Algorithm 7 Minimax FC and Local Sensor Algorithm

```
while z_k < h_f:
    for i = 1 to M:
        g_k^i = g_k^i + ln(f_1(X_k) / f_0(X_k))
        if g_k^i ≥ h_l:
            r_k = r_k + 1
        if g_k^i < 0:
            g_k^i = 0
    end for
    z_k = z_k + ln(R_1(r_k) / R_0(r_k))
    k = k + 1
end while
```

FC Threshold Design Procedure (Offline)

Using the CUSUM procedure at the FC avoids the need for constants ρ and c_d to be specified. We still, however, need to compute an appropriate global threshold, h_f, for the FC. It is desirable to set this threshold such that our system meets the FC false alarm rate constraint with equality to minimize FC detection delay.

Using the shift in pmf analysis, developed for the SPRT in Section 3.3.4 and extended to CUSUM in Section 3.4, we can compute the FC expected detection delay and false alarm rate given our report arrival pmf under H_1 and H_0, and a value of h_f. For a distributed system with M sensors and local sensor thresholds set to h_l, we can iteratively compute the FC false alarm rate of the system under different potential values of h_f. This can be done until a (h_l, h_f) pair is found that designs the system to satisfy the false alarm rate constraint. We will describe in the following section the full system design methodology by which we can determine (h_l, h_f) pairs that can achieve the desired false alarm rate given system parameters and estimate the performance of each of these designs.

5.5 System Design Procedure

Consider a distributed system with M sensors observing a shift in mean Gaussian test. We wish to design this system such that $R_{FA}^{FC} \leq \alpha_{FC}$, while minimizing E_{DD}^{FC}. This problem can be formulated as

$$\min_{h_l, h_f} \quad E_{DD}^{FC} \tag{5.10}$$

$$\text{subject to} \quad R_{FA}^{FC} \leq \alpha_{FC}. \tag{5.11}$$

We can convert α_{FC} to a local sensor minimum false alarm rate constraint, α_l, using

$$\frac{\alpha_{FC}}{M} = \alpha_l. \tag{5.12}$$

We then seek to design local sensors to achieve a local false alarm rate, defined as R_{FA}^{l}, of

$$R_{FA}^{l} \geq \alpha_l. \tag{5.13}$$

When a value of h_l is chosen for the distributed system of M sensors that results in $R_{FA}^{l} = \alpha_l$, the FC should terminate and choose H_1 after only one report in order to satisfy $R_{FA}^{FC} = \alpha_{FC}$. This value of h_l is an upper bound on our potential h_l values. For h_l that result in $R_{FA}^{l} > \alpha_l$, we will need to process more than one report before choosing H_1 at the FC in order to satisfy $R_{FA}^{FC} = \alpha_{FC}$. If we select a value of h_l that causes local sensors to violate (5.13), then for any choice of h_f we will always over design our distributed system in terms of false alarm rate, resulting in greater delay.

We therefore propose that, given the value of α_l from (5.12), we compute the

maximum possible value of h_l that satisfies (5.13) using our CUSUM analysis of Section 3.4 within a search procedure. Once this value of h_l is determined, we can compute its associated reporting pmfs using Algorithm 4 and determine the h_f that designs the system to achieve the closest R_{FA}^{FC} to α_{FC} as possible by treating it as a shift in pmf test, as described in the offline FC threshold design procedure of Section 5.4. We then compute and store the E_{DD}^{FC} and R_{FA}^{FC} of this (h_l, h_f) pair, decrement h_l and repeat this process, storing the E_{DD}^{FC} and R_{FA}^{FC} values for each (h_l, h_f) pair.

We can choose a final (h_l, h_f) pair to design local sensors and the FC based on system requirements. For example, if our only design goal is to minimize expected detection delay while meeting a maximum false alarm rate, we can choose the threshold pair with minimum E_{DD}^{FC}. If, for example, power consumption is a secondary concern within our network, it may be desirable to reduce the number of report transmissions between local sensors and the FC. In this case, we can choose the largest possible h_l among these threshold pairs so that our reports are most informative, and thus less reports are needed for a decision.

We summarize the implementation of this design procedure below:

1. Identify system design and environment variables M, L, and α_{FC}.

2. Using Equation (5.12), convert α_{FC} to a lower bound on the local sensor false alarm rate constraint, α_l.

3. Iteratively compute R_{FA}^{l} for different local sensor thresholds, h_l, using Algorithm 2 and Equation (3.55). Stop the search procedure when R_{FA}^{l} is sufficiently close to α_l. This is the maximum value of h_l that should be used in the distributed system.

4. Using this value of h_l, perform Algorithm 4 to determine the reporting pmf at the FC, $R_1(X_k)$ and $R_0(X_k)$.

5. The arrival of reports at the FC can now be treated as a discrete shift in pmf problem. Iteratively compute R_{FA}^{FC} for different values of h_f using $R_1(X_k)$ and $R_0(X_k)$ as the observation distribution to Algorithm 2 and using Equation (3.55). Stop the search procedure when h_f is sufficiently close to α_{FC}.

6. Record $(E_{DD}^{FC}, R_{FA}^{FC})$ for this threshold pair (h_l, h_f).

7. Decrement h_l and repeat procedure from 4.

5.6 Results

Scalability of System Design

The system design proposed in this chapter, and the accompanying methodology of computing appropriate thresholds, are both better equipped to scale to large sensor networks than existing approaches in literature. The reasons for this are twofold.

First, by using the CUSUM analysis procedures of Chapter 3, we are able to account for sequential test overshoot in the design of local sensors. This allows us to accurately design each local sensor to achieve the desired error metric. This contrasts existing approaches that often neglect sequential test overshoot in the design of local sensor thresholds, which results in more conservative thresholds than needed to meet design specifications. The accumulated impact of over designing many local sensors in large systems will be significant on system wide performance.

Second, the value of M does not result in a significant increase of computational complexity in the design procedure of Section 5.5. The value of M is only used in

the design procedure in Equations (5.12) and (3.61). Of these, increasing M only increases the complexity of computing Equation (3.61), which can be approximated for large M using well known methods such as Stirling's formula.

Accuracy of Proposed Design Approach

We investigate the accuracy of our proposed design procedure by considering a $N(0,1)$ vs $N(0.5,1)$ test under different parameters h_l, M, and L.

Each combination of M, L and h_l resulted in unique report arrival pmfs, that were computed using Algorithm 4 at the local sensor. Algorithm 3 was then used with these pmfs to compute E_{DD}^{FC} and R_{FA}^{FC} for each potential h_f. Monte Carlo trials simulating the entire distributed system were performed to evaluate the accuracy of our proposed procedure in predicting E_{DD}^{FC} and R_{FA}^{FC} given these parameters. This comparison is shown in Figures 5.2, 5.3 and 5.4.

Close agreement is seen between our proposed procedure and Monte Carlo. We remark that the source of error between approaches is likely a result of slight time dependence in the report arrival process in Monte Carlo trials. This dependence was originally highlighted in Section 3.5.2. When a report is received at the FC, it is less likely to receive another report in the immediately following time slots because one or more local sensors have been reinitialized to zero as part of their reporting procedure. It is unlikely that these sensors transmit another report in next few time slots and their reporting probability is time varying for a short time until they reach the assumed steady state pmfs. This time interval is not accounted for in our proposed procedure.

In Section 3.5.2, it was shown that this time dependence has the most effect on

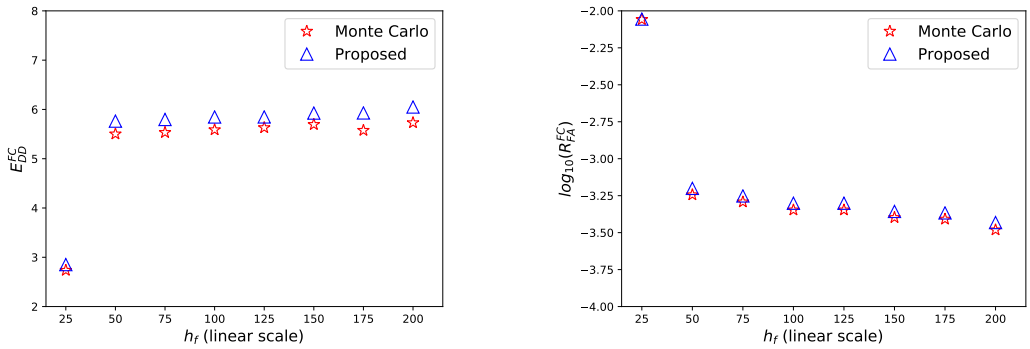
estimating the probability of receiving multiple simultaneous reports. Based on this discussion, we reason that as M gets large and L remains fixed, the time dependence of the reporting sequence will decrease. In such cases, the test termination and reinitialization of a small subset of local sensors will have minor impact on the system-wide reporting probabilities. Given this, and that we are operating away from the change time, we can assume that the reporting sequence will be approximately IID. When these conditions hold, the accuracy of our design procedure in estimating E_{DD}^{FC} and R_{FA}^{FC} for threshold pairs (h_l, h_f) will increase, and our system design will be asymptotically optimal for fixed L as $M \to \infty$ in Lorden's sense.

The time dependence in the arrival of reports also becomes vanishingly small in cases where the difference between hypotheses as observed by local sensors before and after the change gets large. In such conditions, local sensors reach their steady state reporting probabilities more quickly, reducing the transient period between the two reporting pdfs, which impacts the accuracy of computed reporting probabilities.

We also note the staircase phenomenon prominent in each figure. This effect occurs because the report arrival pmfs are discrete distributions with $L + 1$ non-zero terms. A similar effect was seen and discussed when analyzing discrete distributions in Section 3.4. When L is small, the discretization of the observation distributions causes discontinuities in the achievable false alarm rates the FC can be designed for under different values of h_f. In these cases, it is necessary to use randomization at the FC detector to achieve all possible false alarm rates.

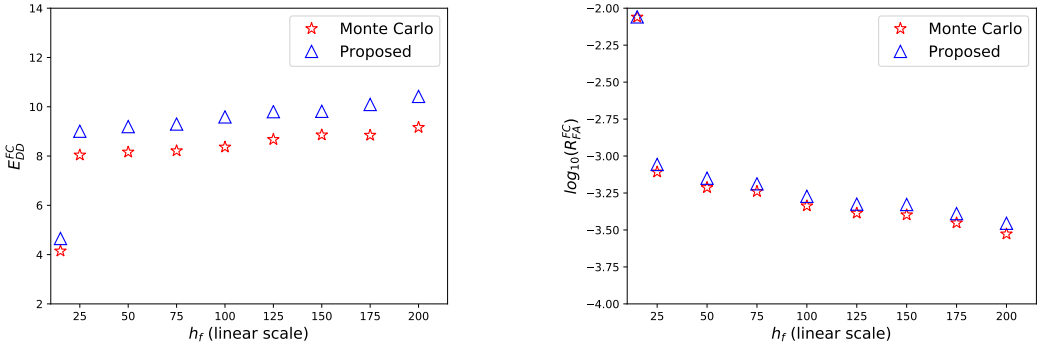
The largest discontinuity in these scenarios occurs at small h_f. This discontinuity is due to the performance difference between designing the FC to always terminate after one report and to always terminate after more than one report. We remark that

this test design is therefore not a Neyman-Pearson test because some α_{FC} are not achievable without randomization. As L increases and the probability distribution of the number of simultaneously received reports is more uniformly distributed over n, $0 \leq n \leq L$, this staircase effect decreases because the discrete report pmfs more closely approximate continuous distributions.



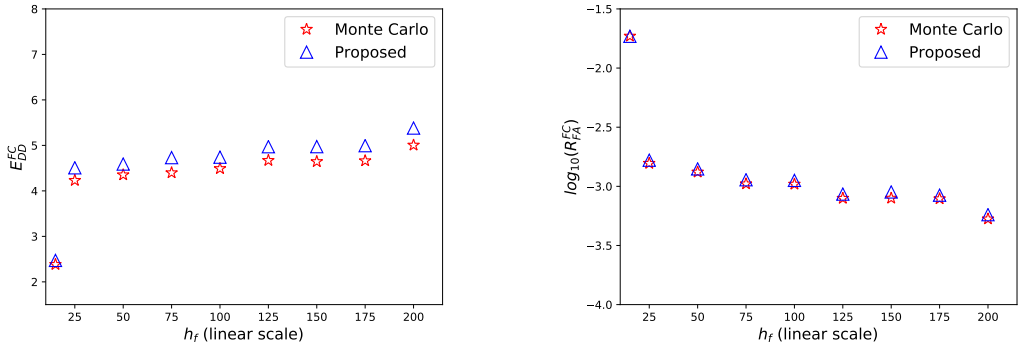
(a) H_1 true (b) H_0 true

Figure 5.2: Expected detection delay and false alarm rate as a function of FC threshold h_f for a $N(0,1)$ vs $N(0.5,1)$ test with $M = 15$, $h_l = \ln(125)$, $L = 1$.



(a) H_1 true (b) H_0 true

Figure 5.3: Expected detection delay and false alarm rate as a function of FC threshold h_f for a $N(0,1)$ vs $N(0.5,1)$ with $M = 7$, $h_l = \ln(60)$, $L = 3$.



(a) H_1 true (b) H_0 true

Figure 5.4: Expected detection delay and false alarm rate as a function of FC threshold h_f for a $N(0,1)$ vs $N(0.5,1)$ with $M = 15$, $h_l = \ln(60)$, $L = 3$.

System Performance

The performance of our proposed distributed system in terms of the expected detection delay when achieving a given false alarm rate is investigated in Figure 5.5. A $N(0,1)$ vs $N(0.5,1)$ test is considered with $M = 13$ and varying L. The difference in system performance between $L = 1$ and $L = 2$ is significant. Two reports are often received simultaneously at the FC in this scenario, therefore it is beneficial to be able to distinguish between one and two reports being received. The difference in using $L = 2$ and $L = 3$ is marginal because of the rarity of receiving three reports simultaneously for this scenario. We can therefore use this analysis to conclude that for this scenario a bandwidth constraint of $L = 2$ is sufficient. The performance improvement from $L = 2$ to $L = 3$ does not merit the increase in resources needed to maintain the higher bandwidth limit.

We also note that the improvement between $L = 2$ and $L = 3$ is more significant as R_{FA}^{FC} decreases. A lower false alarm rate results in, on average, more reports being

needed prior to the FC choosing H_1. This therefore makes it more likely to receive three reports within a test.

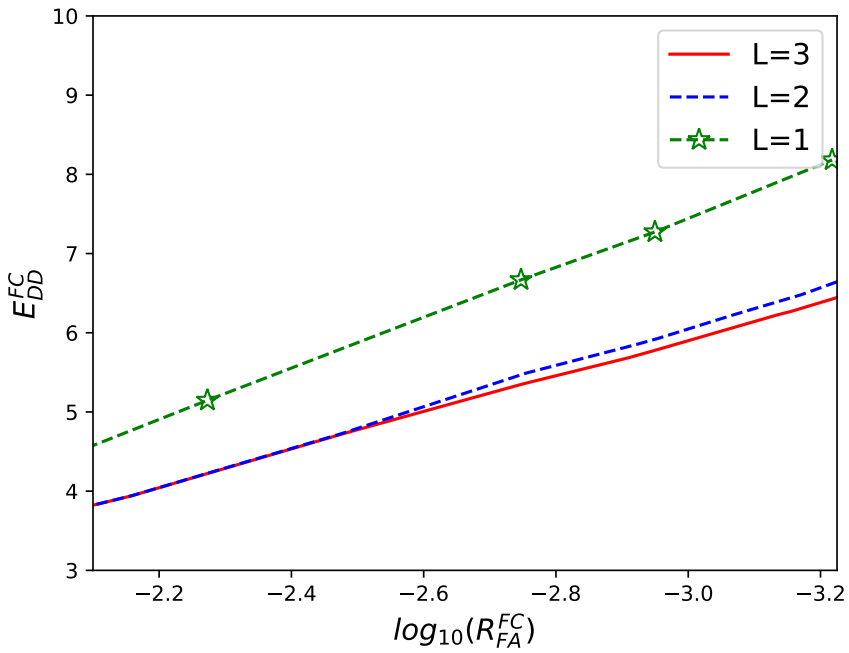


Figure 5.5: Performance of the decentralized CUSUM system observing a $N(0,1)$ vs $N(0.5,1)$ test with $h_l = \ln(15)$ and $M = 13$ under varying bandwidth constraint L.

Figure 5.6 shows the same scenario described above but with fixed $L = 3$. The number of sensors is then varied and system performance is compared. As expected, increasing the number of sensors improves system redundancy, allowing for smaller E_{DD}^{FC} to be achieved for a given false alarm rate. We remark that greater performance improvement is seen by increasing M in this system than in that of Chapter 4. The computational complexity of the design procedure also does not change significantly with M. This system and its design methodology therefore scale more readily to large sensor networks.

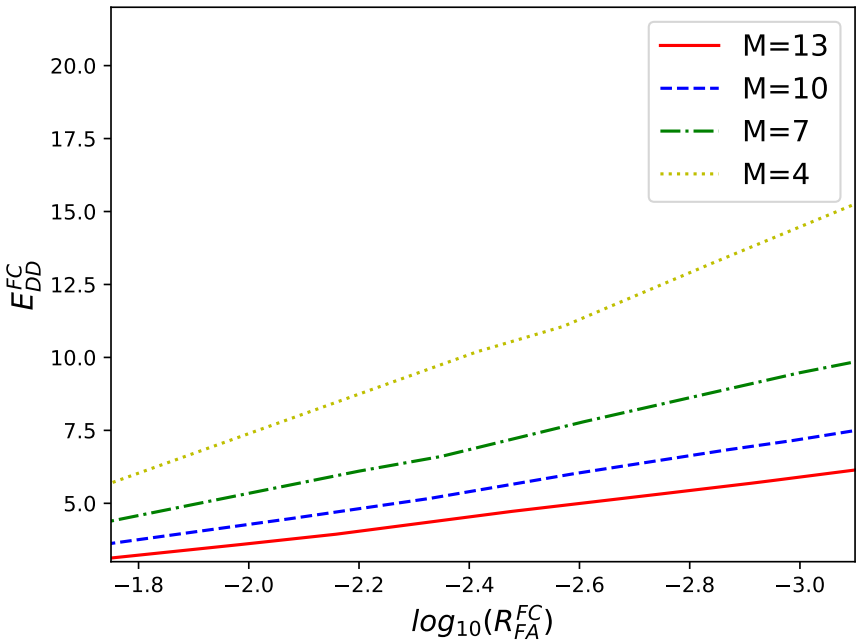


Figure 5.6: Performance of the decentralized CUSUM system observing a $N(0,1)$ vs $N(0.5,1)$ test with $h_l = \ln(15)$ with varying number of sensors, M

5.7 Conclusion

Motivated by the limitations of the Bayesian formulated system design of Chapter 4, a minimax formulation to the distributed quickest change detection problem was investigated in this chapter. A number of distributed system designs using this formulation have been explored in literature, but with the emergence of IoT, there is significant demand for designs that are scalable to large wireless sensor networks. Existing designs are often not capable, or suffer significant performance degradation, when scaling to large networks because of issues such as power consumption, limits on the available wireless spectrum and neglecting sequential test overshoot. Existing designs also do not consider methodologies for designing sensor thresholds within

these systems to meet acceptable error rates.

To address these limitations in existing works, we propose a decentralized, distributed system design that considers bandwidth restrictions on the number of simultaneous summary reports that can be received by the FC from local sensors in any time slot. The proposed design uses the CUSUM procedure at local sensors and at the FC. The numerical procedures of Chapter 3 are used to analyze local sensor operation and compute pmfs describing the arrival of summary reports at the FC. This information is used in the online procedure of the FC, and in the design of local sensor and FC thresholds. Through this, we are able to propose a methodology for designing the thresholds of local sensors and the FC according to a desired limit on the false alarm rate of our system.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis, the distributed quickest change detection problem was investigated. Although this problem has been well explored in existing literature, the vast scale of future IoT networks has highlighted limitations of existing approaches in their application to large sensor networks. These limitations are rooted in simplification and resource management issues such as ignoring local sensor sequential test overshoot, high power consumption and reliance on an abundance of unused wireless spectrum. Existing works also do not consider practical issues with the real-world implementation of these systems. For example, guidelines are needed that govern the design of sensor thresholds in these systems to meet error specifications, and methods of numerically analyzing the performance of a system design prior to implementation hold significant promise in making planning and decision making for system designers easier. This thesis explored these limitations in the context of different formulations of solutions to addressing the distributed quickest change detection problem.

In Chapter 3, an algorithm was introduced that allows for sequential test overshoot

to be taken into account in the design of SPRT and CUSUM thresholds. These procedures provide an improvement in the accuracy of test designs with respect to desired error specifications in comparison to using Wald's approximations. To put this in perspective, in the Gaussian shift in mean example considered in Section 3.3.1, using Wald's approximations to design a SPRT was shown to provide thresholds that resulted in tests that were up to twice as conservative as desired. This in turn approximately doubled the expected test delay at high error probabilities.

It was shown that these procedures had significant potential for application to distributed quickest change detection. First and foremost, designing local sensors to account for sequential test overshoot removes a major obstacle in the scalability of any distributed system design. In large networks, it was expected that the accumulated performance loss resulting from neglecting local sensor overshoot was significant. These procedures, however, also provided insight into the operation of local sensors using any reinitialization strategy within distributed systems. This allowed us to accurately estimate the arrival of summary reports at the FC from local sensors that utilize the CUSUM procedure in distributed systems of different sizes.

Two system designs addressing the distributed quickest detection problem based on Bayesian and minimax formulations were then proposed that take advantage of this algorithm. Insight into the operation of local sensors allowed multiple simultaneous transmissions within the network to be permitted and analyzed uniquely by the FC. The potential restriction of limited bandwidth at the FC was also incorporated and addressed in both proposed solutions.

In Chapter 4, the Bayesian system design was discussed. A decentralized system was proposed that uses the CUSUM procedure at local sensors. The local sensor

CUSUMs were analyzed using the algorithms of Chapter 3 to determine the probability of receiving one or more report in each time slot at the FC. The arrival of reports at the FC was treated as an optimal stopping problem, an approach that has been extensively studied in literature in single sensor systems. Analysis of the performance of this system highlighted implementation challenges and limitations on the scalability of systems using a Bayesian formulation. It was found that incorrect specification of ρ could lead to a change in the false alarm rate of the distributed system in tested scenarios by a factor of five, and that the complexity involved in computing a suitable threshold for the FC increased exponentially as a function of $(M+1)$, where M is the number of local sensors in the system. An operating range was also found to exist in the false alarm rate that could be achieved by the FC for any choice of h_l and ρ, further complicating any threshold design procedure. This system was therefore considered to not be capable of addressing research objectives and have limited potential within large sensor networks.

Such limitations motivated the minimax formulated system design in Chapter 5. This design similarly used the CUSUM procedure at local sensors to quantize local sensor observations into binary reports that were transmitted to the FC. Probability mass functions describing the probability of the FC receiving one or more reports from local sensors in each time slot were computed, and were used by the fusion center to implement a globally run CUSUM. This design was shown to not have limitations in its scalability to large networks and a methodology by which global and local thresholds could be chosen to meet a desired false alarm rate constraint was proposed. It was also shown that the performance of a system design could be numerically computed for different choices of system design variables. The application

of such a performance analysis was showcased in Figure 5.5, where it was concluded that increasing the system bandwidth constraint from $L = 2$ to $L = 3$ only resulted in a less than 5% improvement in performance at $R_{FA}^{FC} = 0.001$. This kind of insight can assist system designers in making important decisions pertaining to the allocation of resources in potential deployments of the proposed system.

6.2 Future Work

- In our current formulation to the distributed quickest change detection problem, it is assumed that local sensors are reliable indefinitely. All summary reports are considered to be equally trustworthy by the FC in determining whether a change has occurred. In practice however, sensors have finite life expectancies. They may eventually break down, causing erroneous reports, or a lack thereof, to be delivered to the FC. It is therefore important to investigate procedures that the FC can implement to guarantee some performance against the potential of faulty sensors. Solutions to this problem could also find application within security critical contexts where one or more sensors may be malicious. The background behind this problem was considered in [18], and extensions of this problem to distributed hypothesis testing and change detection are seen in [15],[33],[10]. These problems consider the concern that one or more sensors may fall under the control of an adversary that tries to ruin the integrity of the system. The distributed system design must therefore be robust against incorrect data transmitted by a subset of local sensors.

- Binary quantization of local sensor observations into summary reports is used

by local sensors in each distributed design in this thesis. This kind of quantization strategy is well motivated in that it is the strategy that requires the least amount of power for local sensors to transmit reports to the FC. However, it also introduces some performance loss within our proposed systems, which can be seen by comparing to the performance of a centralized CUSUM. We therefore propose investigating the scenario where local sensors transmit to the FC their unquantized log-likelihood ratio statistic at test termination. This transmission strategy should allow for the design of a system that more closely approximates that of the centralized CUSUM in terms of performance. Such a design however requires knowledge of the pdf of local sensor test overshoot under H_1 and H_0 to implement similar CUSUM and Bayesian procedures at the FC as was considered in this thesis. These overshoot pdfs are in general difficult to estimate.

- The algorithms proposed and discussed in Chapter 3 can also be applied to analyzing the distributed hypothesis testing problem. Similar system designs that were considered in this thesis could be applied to this problem.

Bibliography

[1] S. Anand and S. K. Routray. Issues and challenges in healthcare narrowband IoT. In *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 486–489, March 2017.

[2] S. Banerjee and G. Fellouris. Decentralized sequential change detection with ordered CUSUMs. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 36–40, July 2016.

[3] T. Banerjee and V. V. Veeravalli. Data-efficient quickest change detection with on-off observation control. *Sequential Analysis*, 31, 05 2011.

[4] T. Banerjee and V. V. Veeravalli. Data-efficient minimax quickest change detection. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3937–3940, March 2012.

[5] T. Banerjee and V. V. Veeravalli. Data-efficient quickest change detection in sensor networks. *IEEE Transactions on Signal Processing*, 63(14):3727–3735, July 2015.

[6] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume I and II. Athena Scientific, 2007.

[7] S. D. Blostein and T. S. Huang. Detecting small, moving objects in image sequences using sequential hypothesis testing. *IEEE Transactions on Signal Processing*, 39(7):1611–1629, July 1991.

[8] R. W. Crow and S. C. Schwartz. Quickest detection for sequential decentralized decision systems. *IEEE Transactions on Aerospace and Electronic Systems*, 32(1):267–283, Jan 1996.

[9] E. J. Dudewicz and S. N. Mishra. *Modern Mathematical Statistics*. New York: Wiley, 1988.

[10] G. Fellouris, E. Bayraktar, and L. Lai. Efficient byzantine sequential change detection. *IEEE Transactions on Information Theory*, 64(5):3346–3360, May 2018.

[11] J. Geng and L. Lai. Non-Bayesian quickest detection with a stochastic energy constraint. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6372–6376, May 2013.

[12] J. Geng and L. Lai. Bayesian quickest detection with stochastic energy constraint. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1846–1850, May 2014.

[13] A. M. Hussain. Multisensor distributed sequential detection. *IEEE Transactions on Aerospace and Electronic Systems*, 30(3):698–708, July 1994.

[14] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. Kwak. The internet of things for health care: A comprehensive survey. *IEEE Access*, 3:678–708, 2015.

[15] B. Kailkhura, Y. S. Han, S. Brahma, and P. K. Varshney. Distributed Bayesian detection in the presence of byzantine data. *IEEE Transactions on Signal Processing*, 63(19):5250–5263, Oct 2015.

[16] M. N. Kurt and X. Wang. Multi-sensor sequential change detection with unknown change propagation pattern. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–1, 2018.

[17] L. Lai, H. V. Poor, Y. Xin, and G. Georgiadis. Quickest search over multiple sequences. *IEEE Transactions on Information Theory*, 57(8):5375–5386, Aug 2011.

[18] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[19] P. A. Laplante and N. Laplante. The internet of things in healthcare: Potential applications and challenges. *IT Professional*, 18(3):2–4, May 2016.

[20] T. S. Lee, S. Ghosh, and A. Nerode. A mathematical framework for asynchronous, distributed, decision-making systems with semi-autonomous entities: algorithm synthesis, simulation, and evaluation. In *Proceedings. Fourth International Symposium on Autonomous Decentralized Systems. - Integration of Heterogeneous Systems* -, pages 206–212, March 1999.

[21] G. Lorden. On excess over the boundary. *The Annals of Mathematical Statistics*, 41(2):520–527, 1970.

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

[23] Y. Mei. Information bounds and quickest change detection in decentralized decision systems. *IEEE Transactions on Information Theory*, 51(7):2669–2681, July 2005.

[24] G. V. Moustakides. Optimal stopping times for detecting changes in distributions. *Ann. Statist.*, 14(4):1379–1387, 12 1986.

[25] G. V. Moustakides. Decentralized CUSUM change detection. In *2006 9th International Conference on Information Fusion*, pages 1–6, July 2006.

[26] H. Zhang O. Hadjiliadis and H. V. Poor. One shot schemes for decentralized quickest change detection. In *2008 11th International Conference on Information Fusion*, pages 1–8, June 2008.

[27] E. S. Page. Continuous Inspection Schemes. *Biometrika*, 41(1-2):100–115, 06 1954.

[28] C. Roff, J. R. Henderson, D. Clarke, M. C. Walden, and S. Fitz. Low-cost millimeter-wave radio-frequency sensors: New applications enabled by developments in low cost chipsets. In *2017 IEEE SENSORS*, pages 1–3, Oct 2017.

[29] K. Routh and T. Pal. A survey on technological, business and societal aspects of internet of things by q3, 2017. In *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pages 1–4, Feb 2018.

[30] Y. Shaikh, V. K. Parvati, and S. R. Biradar. Survey of smart healthcare systems using internet of things (IoT) : (invited paper). In *2018 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, pages 508–513, Feb 2018.

[31] A. N. Shiryaev. On optimum methods in quickest detection problems. *Theory Prob. and Applic.*, 8(1):22–46, 1963.

[32] D. Siegmund. *Sequential analysis: Tests and confidence intervals*. New York: Springer-Verlag, 1985.

[33] E. Soltanmohammadi, M. Orooji, and M. Naraghi-Pour. Decentralized hypothesis testing in wireless sensor networks in the presence of misbehaving nodes. *IEEE Transactions on Information Forensics and Security*, 8(1):205–215, Jan 2013.

[34] A. G. Tartakovsky and H. Kim. Performance of certain decentralized distributed change detection procedures. In *2006 9th International Conference on Information Fusion*, pages 1–8, July 2006.

[35] A. G. Tartakovsky and V. V. Veeravalli. An efficient sequential procedure for detecting changes in multichannel and distributed systems. In *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997)*, volume 1, pages 41–48 vol.1, July 2002.

[36] Alexander G. Tartakovsky and Venugopal V. Veeravalli. Asymptotically optimal quickest change detection in distributed sensor systems. *Sequential Analysis*, 27(4):441–475, 2008.

[37] D. Teneketzis and Y. Ho. The decentralized wald problem. *Information and Computation*, 73(1):23 – 44, 1987.

[38] R. R. Tenney and N. R. Sandell. Detection with distributed sensors. In *1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pages 433–437, Dec 1980.

[39] V. V. Veeravalli. Sequential decision fusion: Theory and applications. *J. Franklin Inst.*, pages 302–322, Feb 1999.

[40] V. V. Veeravalli. Decentralized quickest change detection. *IEEE Transactions on Information Theory*, 47(4):1657–1665, May 2001.

[41] V. V. Veeravalli, T. Basar, and H. V. Poor. Decentralized sequential detection with a fusion center performing the sequential test. *IEEE Transactions on Information Theory*, 39(2):433–442, March 1993.

[42] A. Wald. Sequential tests of statistical hypothesis. *Annals of Mathematical Statistics*, 16:117–186, 1945.

[43] A. Wald and J. Wolfowitz. Optimum character of the sequential probability ratio test. *Annals of Mathematical Statistics*, 19, 11 1947.

[44] H. Wang and S. D. Blostein. Communication-efficient decentralized change detection for cognitive wireless networks. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2015.

[45] Y. Yilmaz, G. V. Moustakides, and X. Wang. Cooperative sequential spectrum sensing based on level-triggered sampling. *IEEE Transactions on Signal Processing*, 60(9):4509–4524, Sep. 2012.

[46] Y. Yilmaz, G. V. Moustakides, and X. Wang. Channel-aware decentralized detection via level-triggered sampling. *IEEE Transactions on Signal Processing*, 61(2):300–315, Jan 2013.

[47] Y. Yilmaz and X. Wang. Sequential distributed detection in energy-constrained wireless sensor networks. *IEEE Transactions on Signal Processing*, 62(12):3180–3193, June 2014.

[48] L. J. Young. Computation of some exact properties of Wald's SPRT when sampling from a class of discrete distributions. *Biometrical Journal*, 36(5):627–637.