Optimum Event Detection in Wireless Sensor Networks

A Thesis
Submitted For the Degree of
Doctor of Philosophy
in the Faculty of Engineering

by

Premkumar Karumbu

Electrical Communication Engineering
Indian Institute of Science
Bangalore – 560 012 (INDIA)

November 2010

Dedicated to

Appa, Amma, and all my Teachers

Abstract

We investigate sequential event detection problems arising in Wireless Sensor Networks (WSNs). A number of battery–powered sensor nodes of the same sensing modality are deployed in a region of interest (ROI). By an event we mean a random time (and, for spatial events, a random location) after which the random process being observed by the sensor field experiences a change in its probability law. The sensors make measurements at periodic time instants, perform some computations, and then communicate the results of their computations to the fusion centre. The decision making algorithm in the fusion centre employs a procedure that makes a decision on whether the event has occurred or not based on the information it has received until the current decision instant. We seek event detection algorithms in various scenarios, that are optimal in the sense that the mean detection delay (delay between the event occurrence time and the alarm time) is minimum under certain detection error constraints.

In the first part of the thesis, we study event detection problems in a small extent network where the sensing coverage of any sensor includes the ROI. In particular, we are interested in the following problems: 1) *quickest event detection with optimal control of the number of sensors that make observations (while the others* sleep*)*, 2) *quickest event detection on wireless ad hoc networks,* and 3) *optimal transient change detection.* In the second part of the thesis, we study the problem of *quickest detection and isolation of an event in a large extent sensor network* where the sensing coverage of any sensor is only a small portion of the ROI.

One of the major applications envisioned for WSNs is detecting any abnormal activity or intrusions in the ROI. An intrusion is typically a rare event, and hence, much of the

energy of sensors gets drained away in the pre–intrusion period. Hence, keeping all the sensors in the awake state is wasteful of resources and reduces the lifetime of the WSN. This motivates us to consider the *problem of* sleep–wake *scheduling of sensors along with quickest event detection.* We formulate the Bayesian quickest event detection problem with the objective of minimising the expected total cost due to i) the detection delay and ii) the usage of sensors, subject to the constraint that the probability of false alarm is upper bounded by α. We obtain optimal event detection procedures, along with optimal closed loop and open loop control for the sleep–wake scheduling of sensors.

In the classical change detection problem, at each sampling instant, a batch of n samples (where n is the number of sensors deployed in the ROI) is generated at the sensors and reaches the fusion centre instantaneously. However, in practice, the communication between the sensors and the fusion centre is facilitated by a wireless ad hoc network based on a random access mechanism such as in IEEE 802.11 or IEEE 802.15.4. Because of the medium access control (MAC) protocol of the wireless network employed, different samples of the same batch reach the fusion centre after random delays. The problem is to detect the occurrence of an event as early as possible subject to a false alarm constraint.

In this more realistic situation, we consider a design in which the fusion centre comprises a *sequencer* followed by a *decision maker.* In earlier work from our research group, a *Network Oblivious Decision Making* (NODM) was considered. In NODM, the decision maker in the fusion centre is presented with complete *batches of observations* as if the network was not present and *makes a decision only at instants at which these batches are presented.* In this thesis, we consider the design in which *the decision maker makes a decision at all time instants* based on the samples of all the complete batches received thus far, and the samples, if any, that it has received from the *next* (partial) batch. We show that for optimal decision making the network–state is required by the decision maker. Hence, we call this setting *Network Aware Decision Making* (NADM). Also, we obtain a mean delay optimal NADM procedure, and show that it is a network–state dependent threshold rule on the a posteriori probability of change.

In the classical change detection problem, the change is persistent, i.e., after the

change–point, the state of nature remains in the in–change state for ever. However, in applications like intrusion detection, the event which causes the change disappears after a finite time, and the system goes to an out–of–change state. The distribution of observations in the out–of–change state is the same as that in the pre–change state. We call this short–lived change a *transient change*. We are interested in detecting whether a change has occurred, even after the change has disappeared at the time of detection.

We model the transient change and formulate the problem of quickest transient change detection under the constraint that the probability of false alarm is bounded by α. We also formulate a change detection problem which maximises the probability of detection (i.e., probability of stopping in the in–change state) subject to the probability of false alarm being bounded by α. We obtain optimal detection rules and show that they are threshold rules on the a posteriori probability of pre–change, where the threshold depends on the a posteriori probabilities of pre–change, in–change, and out–of–change states.

Finally, we consider the problem of detecting an event in a large extent WSN, where the event influences the observations of sensors only in the vicinity of where it occurs. Thus, in addition to the problem of event detection, we are faced with the problem of locating the event, also called the *isolation problem*. Since the distance of the sensor from the event affects the mean signal level that the sensor node senses, we consider a realistic signal propagation model in which the signal strength decays with distance. Thus, the post–change mean of the distribution of observations across sensors is different, and is unknown as the location of the event is unknown, making the problem highly challenging. Also, for a large extent WSN, a distributed solution is desirable. Thus, *we are interested in obtaining distributed detection/isolation procedures which are detection delay optimal subject to false alarm and false isolation constraints.*

For this problem, we propose the following local decision rules, MAX, HALL, and ALL, which are based on the CUSUM statistic, at each of the sensor nodes. We identify corroborating sets of sensor nodes for event location, and propose a global rule for detection/isolation based on the local decisions of sensors in the corroborating sets. Also, we show the minimax detection delay optimality of the procedures HALL and ALL.

Acknowledgments

It gives me immense pleasure to have worked with my PhD advisor Prof. Anurag Kumar. This thesis would not have been possible without the guidance of my advisor. I express my sincere gratitude to you sir. He inspires all of us with his perpetual energy and enthusiasm, and quick–witted understanding. I have been introduced to analytical tools like stochastic processes, queueing theory by him. I am proud to carry with me the wealth of information he provided. Saying a few word of thanks is definitely not sufficient to express my gratitude to you sir.

I would like to express my deep and sincere gratitude to Prof. Joy Kuri who co–advised me on a part of my PhD thesis. His commitment and looking into things at a finer level definitely helped me in shaping my work. I am extremely fortunate to have worked with Prof. Venu Veeravalli. Many thanks to you sir for the discussions we had on the transient change detection problem. I am very much inspired by his acumen in sequential change detection.

I profusely express my sincere gratitude to Prof. Chockalingam for advising me in my MS thesis, and Prof. Vinod Sharma for hiring me in his research group. Thank you sirs. You have a vital role in making me a good researcher.

I owe a great reverence to Prof. Kumar N. Sivarajan and Prof. Utpal Mukherji for having taught me the ever exciting communication networks and random processes. I am indebted to Prof. Shalabh Bhatnagar, Prof. Narahari, and Prof. Rajesh Sundaresan for introducing me to the advanced topics in stochastic optimal control, game theory, and information theory. I am always inspired by the energy and enthusiasm of Prof. Rajesh Sundaresan in whatever he does and his interests in my research and career.

Many thanks are due to MHRD for providing scholarship throughout my stay at IISc. Also, I would like to thank DRDO for supporting the work, and DST, IEEE, IISc, and MSR India for providing me grants to attend conferences. I would be failing in my duty if I do not thank the A–mess for the delicious food (?!?!?!) and a great environment where we discuss quite a lot of intellectual stuff (?!?!?!). Many of my sleepless nights were made less troublesome by the evergreen soul touching music of the legend, Isai Gyani Ilayaraja.

Many thanks are due to Mrs. Chandrika and Savitha of NEL office who took care of all the official things with smile. Many thanks to Anand, Dr. Malati for the discussions we had on the wireless sensor networks project. Thanks to Mahesh, Bore Gowda, and Manjunath for their services. Many thanks to the ECE office for their excellent services; Mr. Srinivasa Murthy deserves a special mention for his quick and perfect services.

Thanks to the folks in the NEL for creating a vibrant environment for research. In particular, I wish to thank Munish, Sayee, Venkat, Manoj, Chandra, Vineeth, Naveen, and Srini. My stay at IISc would not be interesting if I had not met my lab buddies in the PAL and the WRL; thanks to Arun, Kavitha, CT, G. V. V. Sharma. Special thanks to Taposh and CT for accommodating me during my conference travels. The lighter side of life at IISc has never been better for me. My friends at IISc, Rishi, Ananth, RVB, Hari, CV, Srinidhi, Bharath, Pradeep, Nandu, Ram, $\cdots$ (the list is endless) made my stay at IISc more than pleasant and more than interesting.

My deepest gratitude goes to my parents for their unflagging care, love, and support throughout my life; this dissertation is simply impossible without their support. My father Thiru Karumbunathan, and my mother Thirumathi Rajeswari have provided all that they can, and spare no effort to provide the best possible environment for me to grow up. They concealed their problems with an ethereal smile. Thank you appa and amma. I cannot ask for more from my brother Elangovan. You helped me a in a million ways for my conference travels and made them a memorable one. Also, thanks very much for the timely gifts: laptop, pocket hard disc, digital camera, and many more. Thank you annie for all the care and affection when I visited LA. My sisters deserve a special mention. Thank you very much akka for staying close to appa and amma, and taking care of them. The little angels of our family, Abiraami, Bharkavi, Nandini, and Sharanya, brought something pleasant in me, which no words can describe.

Finally, I have no suitable words that can fully or partially describe the everlasting love I have with IISc. IISc gave me all that I wanted and all that I needed. In return, I hope, I will do something good to this academic paradise soon.

Publications based on this Thesis

- **Journal Publications**

  1. K. Premkumar, Anurag Kumar, and Venugopal V. Veeravalli, "Bayesian Quickest Transient Change Detection," *in preparation.*
  2. K. Premkumar, Anurag Kumar, and Joy Kuri, "Distributed Detection/Isolation Procedures for Quickest Event Detection in Large Extent Wireless Sensor Networks," *submitted.*
  3. K. Premkumar and Anurag Kumar, "Optimum Sleep–Wake Scheduling of Sensors for Quickest Event Detection in Small Extent Wireless Sensor Networks," *submitted.*
  4. K. Premkumar, V. K. Prasanthi, and Anurag Kumar, "Delay Optimal Event Detection on Ad Hoc Wireless Sensor Networks," *ACM Transactions on Sensor Networks*, accepted, to appear.

- **Conference Publications**

  1. K. Premkumar, Anurag Kumar, and Joy Kuri, "Distributed Detection and Localization of Events in Large Ad Hoc Wireless Sensor Networks," *Proc.* 47^{th} *Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, 2009 (refereed contribution).
  2. K. Premkumar and Anurag Kumar, "Optimal Sleep–Wake Scheduling for Quickest Intrusion Detection using Wireless Sensor Networks," *Proc. IEEE Infocom*, Phoenix, AZ, USA, 2008.

- **Invited Publications**

  1. Anurag Kumar, *et al.*, “Wireless Sensor Networks for Human Intruder Detection,” *Journal of the Indian Institute of Science*, Special issue on research in the Electrical Sciences, Vol. 90:3, Jul.–Sep. 2010.

  2. K. Premkumar, V. K. Prasanthi, and Anurag Kumar, “Delay Optimal Event Detection on Ad Hoc Wireless Sensor Networks,” *Proc.* 48th *Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, 2010 (abstract).

  3. K. Premkumar, Anurag Kumar, and Venugopal V. Veeravalli, “Bayesian Quickest Transient Change Detection,” *Proc.* 5th *International Workshop on Applied Probability (IWAP)*, Spain, 2010 (abstract).

Contents

Notation

α	probability of false alarm constraint
Δ_k	network delay of the outstanding samples of the current batch under processing
Θ_k	state of nature at time k
Π_k	a posteriori probability of change having occurred at or before time slot k
τ	stopping time with respect to the information sequence $\mathbf{I}_1, \mathbf{I}_2, \cdots$
γ	minimum between time–to–false alarm and time–to–false isolation constraint
$\mathcal{A}$	action space of the POMDP
$\mathcal{A}_i$	detection subregion corresponding to the set of sensor nodes $\mathcal{N}_i$
$\mathcal{B}_i$	influence subregion corresponding to the set of sensor nodes $\mathcal{N}_i$
b	index of the batch sampled at t_b
$C_k^{(i)}$	CUSUM statistic at sensor node i at time k
$D_k^{(i)}$	local decision at sensor node i at time k
E	the random time at which the event disappears
f_0	pre–change pdf
f_1	in–change pdf
$\mathbf{I}_k$	information that the decision maker has received until time k
k	time index
$k-$	time instant just before k
$k+$	time instant just after k
n	number of sensors deployed in the ROI
N	number of detection subregions in a large extent WSN
$\mathcal{N}_i$	set of sensors that detection cover $\mathcal{A}_i$

P_{FA}	probability of false alarm
$\mathbf{Q}_k$	queueing state of the system at time k
r	sampling rate
$\mathcal{S}$	state space of the POMDP
$t_1, t_2, \cdots$	sampling instants
T	change–point or the time at which an event occurs
T_{FA}	time–to–false alarm
T_{FI}	time–to–false isolation
$(x)^+$	$\max\{0, x\}$
$X_k^{(i)}$	observation of sensor node i at time k
$\mathbf{X}_k$	$[X_k^{(1)}, X_k^{(2)}, \cdots, X_k^{(n)}]$, observation vector of all sensor nodes at time k
$\mathbf{X}_{[k_1:k_2]}$	$[\mathbf{X}_{k_1}, \mathbf{X}_{k_1+1}, \cdots, \mathbf{X}_{k_2}]$
$Z_k^{(i)}$	log–likelihood ratio of the observation $X_k^{(i)}$ between the pdfs $f_1^{(i)}$ and $f_0^{(i)}$

Abbreviations

CUSUM	cumulative sum
DP	dynamic program
FJQ	fork–join queue
GPS	generalized processor sharing
KL	Kullback–Leibler
MAC	medium access control
MDP	Markov decision process
NADM	network aware decision making
NODM	network oblivious decision making
POMDP	partially observable Markov decision process
pdf	probability density function
ROI	region of interest
WSN	wireless sensor network

List of Figures

Chapter 1

Introduction

Reasoning and developing systematic techniques for making *inference* has engaged many a great mind since the age of ancient Greek philosophy (sixth century BC) and Indian philosophy (Nyaya sutras of second century AD). The ancient schools of philosophy make inference by syllogism or logical arguments. Since the advent of probability theory, modelling uncertainty by probability models, building such models from statistical data, and deriving inference procedures from such models has become a very important methodology for a large community of scientists and practising engineers.

The quest for environment and habitat monitoring, industrial automation, intrusion detection, identifying locations of survivors in disasters, etc., has given rise to the field of wireless sensor networks (WSNs) in which sensor devices observe the environment and a wireless ad hoc network communicates the observations from the sensor devices to a decision maker that makes inferences. Dramatic advances in low power microelectronics have made the requisite sensor technology and wireless communication technology feasible. Major advances are required, however, in distributed algorithms for signal processing and networking to realise the potential of WSN technology. In this thesis, we are interested in exploring inference problems that arise in sensor networks.

A wireless sensor network (WSN) is formed by a number of tiny, untethered battery–powered devices (popularly called "motes" anticipating the possibility that one day these devices may be as small and unobtrusive as a speck of dust [Mote]) that can sense,

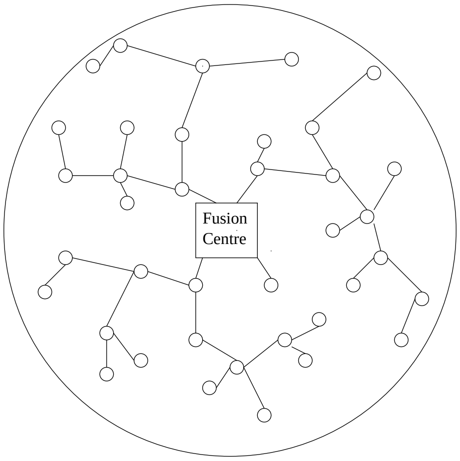


Figure 1.1: An ad hoc wireless sensor network with a fusion centre is shown. The small circles are the sensor nodes (“motes”), and the lines between them indicate wireless links obtained after a self-organization procedure.

compute, and communicate. Figure 1.1 shows a sensor network in which a number of sensor nodes are deployed in the region of interest (ROI) shown by the larger circle. The sensor nodes self–organize to form a network. The observations from the sensor nodes are processed (for e.g., quantized), and the processed data is communicated to the fusion centre through the network. The fusion centre acts as a controller and takes necessary actions based on the application for which it is designed.

Event detection is an important task in many sensor network applications. In general, an event is associated with a change in the distribution of a related quantity that can be sensed. For example, the event of a fire break-out causes a change in the distribution of temperature in that area, and hence, can be detected with the help of temperature sensors. Each sensor node deployed in the ROI, senses and sends some function of its observations (e.g., quantized samples) to the fusion centre at a particular sampling rate. The fusion centre, by appropriately processing the sequence of values it receives, makes a decision regarding the state of nature, i.e., it decides whether an event has occurred or not.

In this thesis, we are interested in obtaining quickest event detection procedures under various scenarios that are detection delay (the delay between the event occurring and the detection decision at the fusion centre) optimal with a constraint on false alarms. We review the literature on change detection in Section 1.1 and identify the shortcomings which we address in this thesis. The main contributions of this thesis is listed in Section 1.2. The organisation of the thesis is given in Section 1.3.

1.1 Literature Survey on Change Detection

Bayesian Change Detection: The classical problem of quickest change detection was formulated and solved in the Bayesian framework by Shiryaev in [Shiryaev, 1963]. In a change detection problem, a stochastic process (describing some aspect of a system) changes from "good" to "bad" state at some unknown time. A decision maker observes the system and needs to infer when the change has occurred based on these observations. Shiryaev assumed the following: i) the observation process is conditionally i.i.d. given the state of nature (i.e., "good" or "bad") and ii) the distribution of the change time is geometric with known mean, and formulated the quickest change detection problem as an optimal stopping problem under the constraint that the probability of false alarm constraint does not exceed α, a parameter of interest. Shiryaev showed that the *optimal stopping rule is a threshold rule on the a posteriori probability of change where the threshold depends on* α.

The conditional i.i.d. assumption of the classical problem is relaxed by Yakir in [Yakir, 1994]. Yakir generalised the classical change detection problem to the case when the pre–change and the post–change processes of observations are finite state Markov chains. Yakir showed that the optimal stopping time is a threshold rule on a posteriori probability of change, and the threshold at time k also depends on the observation at time k.

The geometric distribution assumption on the change time of the classical change detection problem (posed by Shiryaev) was relaxed by Tartakovsky and Veeravalli in [Tartakovsky and Veeravalli, 2005]. Tartakovsky and Veeravalli studied the classical change change detection problem in the Bayesian setting, when the distribution of the change–point is not geometric. In the asymptotic regime, as $\alpha \to 0$, they showed that the optimal detection rule is again a simple threshold rule on the a posteriori probability of change. They also analysed the optimal mean detection delay in the asymptotic setting (i.e., as $\alpha \to 0$).

Non–Bayesian Change Detection: The earliest work on non–Bayesian change detection was by Page in [Page, 1954]. Page proposed CUSUM (CUmulative SUM), a sequential change detection procedure that stops and declares a change at time k, when the CUSUM statistic (a statistic that is recursively computed from the observations) exceeds a certain threshold. The threshold is chosen such that the time–to–false alarm of the CUSUM procedure exceeds γ, a performance objective. It is to be noted that the CUSUM was proposed by Page as a heuristic.

Lorden [Lorden, 1971] showed that Page's CUSUM procedure is asymptotically (as $\gamma \to \infty$) worst–case detection delay optimal, where the worst case is taken over all possible change points and over all possible set of observations before the change point. The optimality of CUSUM for any time–to–false alarm constraint $\gamma > 0$ is shown (in the non–Bayesian framework) by Moustakides in [Moustakides, 1986] and (in the Bayesian framework) by Ritov in [Ritov, 1990].

Shiryaev ([Shiryaev, 1978]), Roberts ([Roberts, 1966], and Pollak ([Pollak, 1985]) independently proposed a non–Bayesian change detection procedure called the Shiryaev–Roberts–Pollak (SRP) test which is obtained as a limit of Bayes rules. Also, it is shown in [Shiryaev, 1978], and [Pollak, 1985] that the SRP procedure is asymptotically average delay optimal as the probability of false alarm goes to zero.

It is to be noted that all the non–Bayesian procedures considered above assume the case of i.i.d. samples before and after the change–point. This condition is relaxed by Lai in [Lai, 1998]. Lai considered stationary and ergodic processes for pre–change and post–change observations, and obtained non–Bayesian minimax delay optimal change detection procedures which are again simple threshold rules.

In [Nikiforov, 1995], Nikiforov proposed a multihypothesis change detection problem, also called a change detection/isolation problem. Nikiforov considered multiple post–change states and that the system after change, enters into one of the post–change states.

Nikiforov proposed a minimax delay optimal solution for the problem under false alarm and false isolation constraints. It is to be noted that the solution proposed by Nikiforov is centralised and the decision statistic can not be computed in a recursive manner, making the procedure computationally expensive.

Decentralised Detection: The problem of decentralised detection was introduced by Tenny and Sandell in [Tenny and Sandell, 1981]. Tenney and Sandell, considered a binary hypothesis testing problem and proposed local decision rules which are threshold rules on likelihood–ratios. In [Aldosari and Moura, 2004], Aldosari and Moura studied the problem of decentralised binary hypothesis testing, where the sensors quantize the observations and the fusion centre makes a binary decision between the two hypotheses.

In [Veeravalli, 2001], Veeravalli considered the problem of decentralised sequential change detection and provided an optimal quantization rule for the sensors and stopping rule for the fusion centre, in the context of conditionally independent sensor observations and a quasi–classical information structure.

In [Tartakovsky and Veeravalli, 2003], Tartakovsky and Veeravalli proposed the following decentralised detection procedures: i) MAX and ii) ALL. Here, each sensor node runs a local change detection procedure (the Shiryaev–Roberts procedure is considered here), which is driven by its own observations only. MAX rule raises an alarm at the time instant when the last local change detection procedure stops, and ALL rule raises an alarm at the time instant when the decision statistic at all the local change detection procedures crosses a threshold. The authors showed that the procedures MAX and ALL are asymptotically optimal as the probability of false alarm constraint $\alpha \to 0$.

In [Mei, 2005], Mei studied the ALL procedure with CUSUM at the sensor nodes for local change detection. He showed that when the time–to–false alarm goes to infinity, the supremum detection delay (in the sense of Lorden's metric [Lorden, 1971]) of ALL is the same as that of centralised CUSUM.

In [Tartakovsky and Veeravalli, 2008], Tartakovsky and Veeravalli studied the MAX procedure with CUSUM at the sensor nodes for local change detection. They showed that when the time–to–false alarm goes to infinity, the supremum detection delay of MAX procedure grows as $\frac{c}{\min_i \mathrm{KL}(f_1^{(i)}, f_0^{(i)})}$, where c is the CUSUM threshold, $\mathrm{KL}(g, h)$ is the Kullback–Leibler divergence between the probability density functions (pdfs) g and h, and $f_0^{(i)}, f_1^{(i)}$s are the pre–change and the post–change pdfs of the observation at sensor node i.

For a large network setting, Niu and Varshney [Niu and Varshney, 2005] studied a simple hypothesis testing problem and proposed a *counting rule* based on the number of alarms. They showed that, for a sufficiently large number of sensors, the detection performance of the counting rule is close to that of the centralised optimal rule.

1.1.1 Limitations of the Classical Change Detection Problem

We note that the classical change detection problem does not address the following issues.

1. Sensor nodes are energy–constrained. Hence, it is important to consider the situation in which the sensor nodes undergo a *sleep–wake cycling*, and thus only the sensor nodes that are in the awake state send their observations to the fusion centre. This problem of optimal stopping with sleep–wake cycling of sensors needs to be studied.

2. In practice, the sensors and the fusion centre are connected by a wireless ad hoc network based on a random access mechanism such as in IEEE 802.11 or IEEE 802.15.4. Hence, the assumption (of the classical change detection problem) that at a sampling instant, the observations from all the sensors reaches the fusion centre instantaneously does not hold true. Hence, the problem of quickest change detection over wireless ad hoc networks remains unanswered in the literature.

3. The classical change detection problem assumes that once the change occurs, it remains there for ever. In some applications, such as structural health monitoring, the model of a permanent change (also called persistent change) might be a reasonable one, but this assumption is not true for many applications like intrusion detection. Thus, the problem of transient change detection is left open in the literature.

4. In the case of a large system, it is not always true that the change affects the statistics of the observations of all the nodes. Hence, the post–change distribution of different nodes can be different. Also, in applications like intrusion detection, the location of the event has a bearing on the mean of the post–change distribution. To the best of our knowledge, detection problems of this kind have not been studied in the literature.

In this thesis, we formulate and solve four change detection problems in each of which one of the limitations mentioned above has been removed.

1.2 Main Contributions of the Thesis

- **Sleep–wake scheduling of sensors for quickest event detection in small extent networks:**

  1. We provide a model for the sleep –wake scheduling of sensors by taking into account the cost per observation (which is the sensing + computation+ communication cost) per sensor in the awake state and formulate the joint sleep –wake scheduling and quickest event detection problem subject to a false alarm constraint, in the Bayesian framework, as an optimal control problem. We show that the problem can be modelled as a partially observable Markov decision process (POMDP).

  2. We obtain an average delay optimum stopping rule for event detection and show that the stopping rule is a threshold rule on the a posteriori probability of change.

  3. Also, at each time slot k, we obtain the optimal strategy for choosing the optimum number of sensors to be in the awake state in time slot $k + 1$ based on the sensor observations until time k, for each of the control strategies described as follows:

(i) control of M_{k+1}, the number of sensors to be in the awake state in time slot $k+1$,

(ii) control of q_{k+1}, the probability of a sensor to be in the awake state in slot $k+1$, and

(iii) constant probability q of a sensor in the awake state in any time slot.

- **Event Detection on a small extent ad hoc wireless network**

  1. We formulate the problem of quickest event detection on ad hoc wireless network.
  2. We propose a class of decision strategies called NADM, in which the decision maker makes a decision based on the samples as and when it comes, but in time–sequence order.
  3. We obtain an optimal change detection procedure the mean detection delay of which is minimal in the class of NADM policies for which $\mathsf{P}_{\mathsf{FA}} \leqslant \alpha$.
  4. We study the tradeoff between the sampling rate, r and the mean detection delay. We also study the detection delay performance as a function of the number of nodes n, for a given *number of observations per unit slot*, i.e., for a fixed nr.

- **Transient change detection**

  1. We provide a model for the transient change and formulate the optimal transient change detection problem.
  2. We obtain the following procedures for detecting a transient change:

     (i) MinD (**Min**imum Detection **D**elay) which minimises the mean detection delay when the probability of false alarm is limited to α

     (ii) $\mathsf{A–MinD}$ (**A**symptotic – **Min**imum Detection **D**elay) which is obtained as a limit of of the MinD procedure when the mean time until the occurrence of change goes to ∞ (i.e., for a rare event)

(iii) MaxP (**Max**imum **P**robability of change) which maximises the probability of stopping when the change is present (which we call the probability of detection) when the probability of false alarm is limited to α.

- **Event detection in large extent wireless sensor networks**

1. We formulate the event detection/isolation problem in a large extent network as a worst case detection delay minimisation problem subject to a mean time to false alarm and mean time to false isolation constraints. Because of the large extent network, the postchange distribution is unknown, and the latter is a novel aspect of our problem formulation.

2. We propose distributed detection/isolation procedures MAX, ALL, and HALL (**H**ysteresis modified **ALL**) for large extent wireless sensor networks. The procedures MAX and ALL are extensions of the decentralised procedures MAX [Tartakovsky and Veeravalli, 2003] and ALL [Mei, 2005], which were developed for small extent networks. The distributed procedures MAX, ALL, and HALL are computationally less complex and more energy–efficient compared to the centralised procedure given by Nikiforov [Nikiforov, 1995] (which can be applied only to the Boolean sensing model).

3. We analyse the supremum worst case detection delay (SADD) of MAX, ALL, and HALL when the mean time to false alarm (T_{FA}) and the mean time to false isolation (T_{FI}) are at least as large as a certain threshold γ. For the case of the Boolean sensing model, we compare the detection delay performance of the these distributed procedures with that of Nikiforov's procedure [Nikiforov, 1995] (a centralised asymptotically optimal procedure) and show that the distributed procedures ALL and HALL are asymptotically order optimal.

1.3 Organisation of the Thesis

In Chapter 2, we introduce the basic change detection problem and define various metrics of interest like the detection delay, probability of detection, etc. We also discuss the various centralised and the decentralised detection procedures available in the literature in this chapter.

In Chapter 3, we study the problem of event detection with minimum number of sensors in the awake state. We formulate the problem and cast it in the framework of Markov Decision Process (MDP) and obtain optimal closed loop and open loop control policies for sleep–wake scheduling of sensor nodes along with the optimum detection rule.

In Chapter 4, we study the problem of detection on ad hoc wireless sensor networks where the reception times of the packets at the fusion centre are not in the same time–order as the sampling times. We provide a decision strategy called Network Aware Decision Making (NADM) and obtain the mean delay optimal NADM procedure.

In Chapter 5, we are interested in detecting a transient–change. We propose a Markov model for transient change, and formulate the optimal transient change detection problem as a Markov Decision Process and obtain various detection procedures for optimality criterion like detection delay and probability of detection.

In Chapter 6, we consider a large extent network where the statistics of the observations are affected by the event only in the vicinity of where it occurs. We formulate the problem in the framework of Nikiforov [Nikiforov, 1995], and propose distributed detection/isolation procedures and discuss their minimax optimality.

In Chapter 7, we conclude the thesis by outlining the list of contributions and the future directions of research in this field.

The proofs of Theorems/Lemmas/Propositions in each chapter are provided in the Appendix of the chapter.

Part I

Event Detection in Small Extent Networks

Chapter 2

The Basic Change Detection Problem

In this chapter, we discuss the basic change detection problem and the performance metrics involved. In Section 2.1, we describe the basic change detection problem. The performance metrics that are typically used in the change detection problem are defined in Section 2.2. In Section 2.3, we describe the Bayesian change detection problem. In Section 2.4, we describe the non–Bayesian change detection problem and discuss the centralised procedure CUSUM, and the decentralised procedures, MAX and ALL.

2.1 Classical Change Detection Problem

Consider a discrete time system with time instants $k \in \mathbb{Z}_+$. At each time instant $k \geqslant 1$, an observation is made by each of n nodes. Let the vector random variable $\mathbf{X}_k = [X_k^{(1)}, X_k^{(2)}, \cdots, X_k^{(n)}]$ represent the observations made by the nodes at time instant k. A change occurs at a random time $T \in \mathbb{Z}_+$. Before the change–point (i.e., for $k < T$), the random variables $X_k^{(i)}$ are i.i.d. across nodes and time, and the distribution of $X_k^{(i)}$ is given by $F_0^{(i)}$. After the change–point (i.e., for $k \geqslant T$), the random variables $X_k^{(i)}$ are i.i.d. across nodes and time, and the distribution of $X_k^{(i)}$ is given by $F_1^{(i)}$. Let the corresponding probability density functions (pdfs) be $f_0^{(i)}$ and $f_1^{(i)}$ respectively (where

$f_1^{(i)} \neq f_0^{(i)}, \forall i)$. The problem is to detect the change as early as possible subject to a constraint on the false alarm. Let τ be the time instant at which the decision maker stops and declares a change (thereby asserting that the change has occurred at or before τ). Since the inference is based on "noisy" observations, it is entirely possible that $\tau < T$, which would be a false alarm. On the other hand, $\tau > T$ would result in detection delay.

2.2 Definitions

Let τ be a stopping time with respect to the observation sequence, $\mathbf{X}_1$, $\mathbf{X}_2$, $\cdots$, i.e., for any $k \in \mathbb{Z}_+$, the occurrence of the event $\{\tau \leqslant k\}$ can be determined by $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_k$. We use the terms stopping time and change detection procedure interchangeably as the stopping time defines a sequential change detection procedure.

Definition 2.1 Probability of False Alarm (P_{FA}) of a procedure τ is defined as the probability of stopping before the change–point T, i.e.,

$$\mathsf{P}_{\mathsf{FA}}(\tau) \ := \ \mathsf{P}\left\{\tau < T\right\}.$$

In many detection problems, a false alarm incurs a cost, and hence, a low P_{FA} is desirable.

Definition 2.2 Mean Detection Delay (ADD) of a procedure τ is defined as the expected number of samples between the change–point, T and the stopping time, τ i.e.,

$$\mathsf{ADD}(\tau) \ := \ \mathsf{E}\left[(\tau - T)^+\right].$$

We note that the notation $(x)^+ := \max\{x, 0\}$. In literature on change detection, there is also a notion of mean detection delay defined as $\mathsf{E}[\tau - T \mid \tau \geqslant T]$.

Consider two stopping times τ_1, τ_2 such that $\tau_1 \leqslant \tau_2$ almost surely. Then, it is easy to see from the definitions of mean detection delay and probability of false alarm that $\mathsf{ADD}(\tau_1) \leqslant \mathsf{ADD}(\tau_2)$ and $\mathsf{P}_{\mathsf{FA}}(\tau_1) \geqslant \mathsf{P}_{\mathsf{FA}}(\tau_2)$. Thus, a lower mean detection delay comes with a price of higher probability of false alarm.

Definition 2.3 Time to False Alarm (T_{FA}) of a procedure τ is defined as the expected number of samples taken by the procedure to stop in the pre–change state.

$$\mathsf{T}_{\mathsf{FA}}(\tau) \quad := \quad \mathsf{E}_{\infty}\left[\tau\right].$$

The notation $\mathsf{E}_{\infty}[\cdot]$ means that the change has not occurred until time τ, and the expectation is taken with respect to the product distribution of $F_0^{(i)}$s.

Definition 2.4 Supremum Average Detection Delay (SADD) of a procedure τ is defined as the worst case expected number of samples between the change–point and the stopping time, where the worst case is over all possible values of change–point t and over all possible set of observations before the change–point, i.e.,

$$\mathsf{SADD}(\tau) \quad := \quad \sup_{t \geqslant 1} \ \text{ess sup } \mathsf{E}_t\left[(\tau - t + 1)^+ \mid \mathbf{X}_{[1:t-1]}\right].$$

The notation $\mathsf{E}_t[\cdot]$ means that the expectation is taken with respect to the distribution when the change–point is t, conditioned on the observations until $t-1$. The distribution of $\mathbf{X}_{[1:k]}$, when the change–point is t, can be described by the following pdf,

$$f(\mathbf{x}_{[1:k]}; t) \quad := \quad \begin{cases} \prod_{k'=1}^{k} \prod_{i=1}^{n} f_0^{(i)}(x_{k'}^{(i)}), & \text{if } k < t \\ \left[\prod_{k'=1}^{t-1} \prod_{i=1}^{n} f_0^{(i)}(x_{k'}^{(i)})\right] \cdot \left[\prod_{k''=t}^{k} \prod_{i=1}^{n} f_1^{(i)}(x_{k''}^{(i)})\right], & \text{if } k \geqslant t. \end{cases}$$

Definition 2.5 Bayesian Detection Procedure: A detection procedure is said to be Bayesian if the procedure uses the distribution of the change–point T. The distribution of the change–point is also called the *prior*.

An example of a Bayesian change detection procedure is Shiryaev's procedure, [Shiryaev, 1978].

Definition 2.6 Non–Bayesian Detection Procedure: A detection procedure is said to be non–Bayesian if no prior distribution of the change–point T is provided during the design of the procedure.

In non–Bayesian change detection problems, the change–point T is typically considered as an unknown constant. An example of a non–Bayesian change detection procedure is Page's CUSUM procedure, [Page, 1954].

Definition 2.7 Centralised Detection Procedure: A detection procedure is said to be *centralised* if at each time instant, the observations from the nodes are passed on to a *centralised decision maker* which makes a decision about whether the change has occurred or not.

Examples of centralised change detection procedures are Shiryaev's procedure and the CUSUM procedure.

Definition 2.8 Decentralised Detection Procedure: A detection procedure is said to be *decentralised* if at each time instant, *each node makes a local decision based on its observations only*, and the local decisions from the nodes are passed on to a *decision maker* which makes a *global decision* about whether the change has occurred or not.

Note that the local decision could be a *quantisation* of the observations into one of several levels or a local change detection based on CUSUM. Examples of decentralised change detection procedures are MAX procedure ([Tartakovsky and Veeravalli, 2003], [Tartakovsky and Veeravalli, 2008]) and ALL procedure ([Tartakovsky and Veeravalli, 2003], [Mei, 2005], [Tartakovsky and Veeravalli, 2008]).

2.3 The Bayesian Change Detection Problem

In Section 2.1, we have discussed the problem of change detection where we have not made any comment about the change–time T. In the Bayesian version of the change detection problem, the distribution of the change–point (called as the *prior*) is known. In the classical Bayesian change detection problem ([Shiryaev, 1978]), the distribution of T is assumed to be geometric and is given by

$$\mathsf{P}\{T=k\} = \begin{cases} \rho, & \text{if } k \leqslant 0 \\ (1-\rho)(1-p)^{k-1}p, & \text{if } k > 0, \end{cases}$$

where $0 < p \leqslant 1$ and $0 \leqslant \rho \leqslant 1$ represents the probability that the event happened even before the observations are made ($k \leqslant 0$).

The Bayesian change detection problem is to detect the change as early as possible subject to the constraint that the probability of false alarm is bounded by α, a parameter of interest. Let τ be the time instant at which the change is detected. Note that τ is a stopping time with respect to the observation sequence $\mathbf{X}_1, \mathbf{X}_2, \cdots$. Then, the optimal change detection problem formulated by Shiryaev is given by

$$\tau^{\mathsf{Shiryaev}} \in \arg \min_{\tau \in \Delta(\alpha)} \mathsf{E}\left[(\tau - T)^+\right]$$

where $\Delta(\alpha) := \{\text{stopping time } \tau : \mathsf{P}\{\tau < T\} \leqslant \alpha\}$. Shiryaev showed that a sufficient statistic for this problem at time k is given by the a posteriori probability of change, $\Pi_k = \mathsf{P}\left\{T \leqslant k \mid \mathbf{X}_{[1:k]}\right\}$. Shiryaev also obtained the optimal Bayesian change detection rule τ^{Shiryaev} which is given by the threshold rule,

$$\tau^{\mathsf{Shiryaev}} = \inf\{k : \Pi_k \geqslant \Gamma\},$$

where the threshold Γ is chosen such that the false alarm criterion is met with equality, i.e., $\mathsf{P}\left\{\tau^{\mathsf{Shiryaev}} < T\right\} = \alpha$.

2.4 Non–Bayesian Change Detection Problem

2.4.1 CUmulative SUM (CUSUM)

In the non–Bayesian problem, the change point T is assumed to be an unknown constant or a random variable whose distribution is unknown. In the non–Bayesian centralised detection procedure, at each time instant k, the decision maker receives the observation vector $\mathbf{X}_k = [X_k^{(1)}, X_k^{(2)}, \cdots, X_k^{(n)}]$, and computes the log–likelihood ratio (LLR) Z_k between the $\mathsf{post\text{–}change}$ and the $\mathsf{pre\text{–}change}$ distributions as follows.

$$\begin{aligned} Z_k &:= \sum_{i=1}^{n} Z_k^{(i)}, \\ \text{where } Z_k^{(i)} &:= \ln\left(\frac{f_1^{(i)}(X_k^{(i)})}{f_0^{(i)}(X_k^{(i)})}\right). \end{aligned}$$

The decision maker then computes the CUSUM statistic C_k as

$$C_k \;:=\; (C_{k-1} + Z_k)^+$$

where $C_0 := 0$. Recall that the notation $(x)^+ := \max\{x, 0\}$. The stopping rule CUSUM is given by Page ([Page, 1954]) as follows.

$$\tau^{\mathsf{CUSUM}} \;=\; \inf\left\{k : C_k \geqslant c\right\},$$

where the threshold c is chosen such that a $\mathsf{time\text{–}to\text{–}false\ alarm}$, $\mathsf{T_{FA}}$ constraint is met, i.e., $\mathsf{E}_\infty\left[\tau^{\mathsf{CUSUM}}\right] = \gamma$. The CUSUM statistic also has a maximum–likelihood interpretation ([Basseville and Nikiforov, 1993]). The optimality of CUSUM is shown by Lorden in [Lorden, 1971]. Lorden showed that CUSUM is asymptotically minimax detection delay optimal, i.e., as the $\mathsf{T_{FA}}$ constraint $\gamma \to \infty$,

$$\tau^{\mathsf{CUSUM}} \;\in\; \arg \inf_{\{\tau : \mathsf{T_{FA}} \geqslant \gamma\}} \mathsf{SADD}(\tau).$$

Also, Lorden showed that the asymptotic SADD of CUSUM is

$$\mathsf{SADD}(\tau^{\mathsf{CUSUM}}) \;\sim\; \frac{\ln\gamma}{\sum\limits_{i=1}^{n} \mathrm{KL}(f_1^{(i)}, f_0^{(i)})}, \quad \text{as } \gamma \to \infty,$$

where $\mathrm{KL}(f, g)$ is the Kullback–Leibler divergence between the pdfs f and g.

In the decentralised approach, each sensor makes a local decision based only on its own observations, and the local decisions are communicated to the global decision maker. The global decision maker makes a decision on the occurrence of change. In the rest of this section, we discuss decentralised non–Bayesian change detection procedures where the local decision rules are based on the CUSUM statistics. Let $Z_k^{(i)}$ be the LLR of the observation $X_k^{(i)}$ between the pdfs $f_1^{(i)}$ and $f_0^{(i)}$. Node i then computes the CUSUM statistic $C_k^{(i)}$ based on its own observations only, i.e.,

$$C_k^{(i)} \quad := \quad \left(C_{k-1}^{(i)} + Z_k^{(i)}\right)^+,$$

where $C_0^{(i)} := 0$. Based on the CUSUM statistic $C_k^{(i)}$, the node i makes a local decision $D_k^{(i)} \in \{0, 1\}$. A number of possibilities arise for the choice of local decision rules. In this chapter, we consider two decentralised procedures i) MAX and ii) ALL.

2.4.2 MAX Procedure

In [Tartakovsky and Veeravalli, 2003], Tartakovsky and Veeravalli proposed MAX rule, a decentralised procedure for change detection. In this procedure, each node i employs CUSUM for change detection. The local CUSUM in sensor node i is driven only by the observations of node i. Let $\tau^{(i),\mathsf{CUSUM}}$ be the time instant at which the CUSUM procedure in node i stops. The global decision rule is given by the following

$$\tau^{\mathsf{MAX}} \quad := \quad \max\left\{\tau^{(1),\mathsf{CUSUM}}, \tau^{(2),\mathsf{CUSUM}}, \cdots, \tau^{(n),\mathsf{CUSUM}}\right\}.$$

In [Tartakovsky and Veeravalli, 2008], Tartakovsky and Veeravalli also studied the asymptotic worst case detection delay of MAX procedure and is given by

$$\mathsf{SADD}\left(\tau^{\mathsf{MAX}}\right) \quad \sim \quad \frac{\ln \gamma}{\min\limits_{1 \leqslant i \leqslant n} \mathrm{KL}(f_0^{(i)}, f_1^{(i)})}, \quad \text{as } \gamma \to \infty.$$

In the special case of $f_0^{(i)} = f_0$ and $f_1^{(i)} = f_1$ for all $1 \leqslant i \leqslant n$, it is easy to see from $\mathsf{SADD}(\tau^{\mathsf{CUSUM}})$ and $\mathsf{SADD}(\tau^{\mathsf{MAX}})$ that as $\gamma \to \infty$, the worst case detection delay of τ^{MAX}

is n times that of the centralised CUSUM procedure.

2.4.3 ALL Procedure

Tartakovsky and Veeravalli ([Tartakovsky and Veeravalli, 2003]), Mei ([Mei, 2005]), and Tartakovsky and Veeravalli ([Tartakovsky and Veeravalli, 2008]) proposed ALL rule, a decentralised change detection procedure based on the CUSUM statistic of each sensor node. In this procedure, the local decision $D_k^{(i)}$ at each sensor node i is obtained using the statistic $C_k^{(i)}$ as follows.

$$D_k^{(i)} := \begin{cases} 0, & \text{if } C_k^{(i)} < c \\ 1, & \text{otherwise,} \end{cases}$$

where c is the CUSUM threshold used at the nodes. In this procedure, the CUSUM at the sensor nodes do not stop even after crossing the threshold. The global decision rule τ^{ALL} is given by

$$\begin{aligned} \tau^{\mathsf{ALL}} &:= \inf\left\{k : D_k^{(i)} = 1,\ \forall\, i = 1, 2, \cdots, n\right\} \\ &= \inf\left\{k : C_k^{(i)} \geqslant c,\ \forall\, i = 1, 2, \cdots, n\right\}. \end{aligned}$$

Choosing the local CUSUM threshold $c = \ln\gamma$ achieves the mean time–to–false alarm larger than γ. For this choice of c, the asymptotic worst case detection delay of ALL procedure is given by

$$\mathsf{SADD}\left(\tau^{\mathsf{ALL}}\right) \sim \frac{\ln\gamma}{\sum\limits_{i=1}^{n} \mathrm{KL}(f_0^{(i)}, f_1^{(i)})}, \quad \text{as } \gamma \to \infty.$$

From $\mathsf{SADD}\left(\tau^{\mathsf{CUSUM}}\right)$ and $\mathsf{SADD}\left(\tau^{\mathsf{ALL}}\right)$, it is easy to see that asymptotically the worst case detection delay performance of ALL, a decentralised procedure, is the same as that of the centralised procedure CUSUM.

Chapter 3

Quickest Event Detection with Sleep–Wake Scheduling

3.1 Introduction

In the previous chapter, we have discussed the classical change detection problem in which the decision maker, after having observed the kth sample, has to make a decision to stop at the kth sample instant, or to continue observing the $k+1$th sample. There, the decision maker is concerned only about minimising the detection delay. However, in many applications, there is a cost associated with generating an observation and communicating it to the decision maker.

When a WSN is used for physical intrusion detection applications (e.g., detection of a human intruder into a secure region), much of the energy of the sensor nodes gets drained away in the *pre–intrusion* period. As sensor nodes are energy–limited devices, this reduces the utility of the sensor network. Thus, *in addition to the problem of quickest event detection, we are also faced with the problem of increasing the lifetime of sensor nodes.* We address this problem in this chapter, by means of optimal sleep–wake scheduling of sensor nodes.

A sensor node can be in one of two states, the sleep state or the awake state. A sensor node in the sleep state conserves energy by switching to a low–power state. In the awake state, a sensor node can make measurements, perform some computations, and then communicate information to the fusion centre. For enhancing the utility and the lifetime of the network, it is essential to have *optimal* sleep–wake *scheduling* for the sensor nodes.

In this chapter, we are interested in the quickest detection of an event with a minimal number of sensors in the awake state. A common approach to this problem is by having a fixed deterministic duty cycle for the sleep–wake activity. However, the duty cycle approach does not make use of the prior information about the event, nor the observations made by the sensors, and hence is not optimal. To the best of our knowledge, our work is the first to look at the problem of joint design of optimal change detection and sleep–wake scheduling.

3.1.1 Summary of Contributions

We summarise the main contributions of this chapter below.

1. We provide a model for the sleep –wake scheduling of sensors by taking into account the cost per observation (which is the sensing + computation + communication cost) per sensor in the awake state and formulate the joint sleep –wake scheduling and quickest event detection problem subject to a false alarm constraint, in the Bayesian framework, as an optimal control problem. We show that the problem can be modelled as a partially observable Markov decision process (POMDP).

2. We obtain an average delay optimum stopping rule for event detection and show that the stopping rule is a threshold rule on the a posteriori probability of change.

3. Also, at each time slot k, we obtain the optimal strategy for choosing the optimum number of sensors to be in the awake state in time slot $k+1$ based on the sensor observations until time k, for each of the control strategies described as follows:

   (i) control of M_{k+1}, the number of sensors to be in the awake state in time slot $k+1$,

(ii) control of q_{k+1}, the probability of a sensor to be in the awake state in slot $k+1$, and

(iii) constant probability q of a sensor in the awake state in any time slot.

3.1.2 Discussion of the Related Literature

In this section, we discuss the most relevant literature on energy–efficient detection. Censoring was proposed by Rago *et al.* in [Rago et al., 1996] as a means to achieve energy–efficiency. *Binary hypothesis testing* with energy constraints was formulated by Appadwedula *et al.* in [Appadwedula et al., 2005]. These schemes find the "information content" in any observation, and uninformative observations are not sent to the fusion centre. Thus, censoring saves only the communication cost of an observation. In our work, by making a sensor go to the sleep state, we save the sensing + computation+ communication cost of making an observation.

In related work [Wu et al., 2007], Wu *et al.* proposed a low duty cycle strategy for sleep –wake scheduling for sensor networks employed for data monitoring (data collection) applications. In the case of sequential event detection, duty cycle strategies are not optimal, and it would be beneficial to adaptively turn the sensor nodes to the sleep or awake state based on the prior information, and the observations made during the decision process, which is the focus of this chapter.

In [Zacharias and Sundaresan, 2007], Zacharias and Sundaresan studied the problem of event detection in a WSN with physical layer fusion and power control at the sensors for energy–efficiency. Their Markov decision process (MDP) framework is similar to ours. However, in [Zacharias and Sundaresan, 2007], all the sensor nodes are in the awake state at all time. In our work, we seek an optimal state dependent policy for determining how many sensors to be kept in the awake state, while achieving the inference objectives (detection delay and false alarm).

3.1.3 Outline of the Chapter

The rest of the chapter is organised as follows. In Section 3.2, we formulate the sleep –wake scheduling problem for quickest event detection. We describe various costs associated with the event detection problem. Also, we outline various control strategies for sleep –wake scheduling of sensor nodes. In Section 3.3, we discuss the optimal sleep –wake scheduling problem that minimises the detection delay when there is a feedback from the decision maker (in this case, the fusion centre) to the sensors. In particular, the feedback could be the number of sensors to be in the awake state or the probability of a sensor to be in the awake state in the next time slot. We show that the a posteriori probability of change is sufficient for stopping and for controlling the number of sensors to be in the awake state. In Section 3.4, we discuss an optimal open loop sleep –wake scheduler that minimises the detection delay where there is no feedback from the fusion centre and the sensor nodes. We obtain the optimal probability with which a sensor node is in the awake state at any time slot. In Section 3.5, we provide numerical results for the sleep –wake scheduling algorithms we obtain. Section 3.6 summarises the results in this chapter.

3.2 Problem Formulation

In this section, we describe the problem of *quickest event detection with a cost for taking observations* and set up the model. We consider a WSN comprising n unimodal sensors (i.e., all the sensors have the same sensing modality, e.g., acoustic, vibration, passive infrared (PIR), or magnetic) deployed in a region $\mathcal{A}$ for an intrusion detection application. We consider a small extent network, i.e., the region $\mathcal{A}$ is covered by the *sensing coverage* of each of the sensors. An event (for example, a human "intruder" entering a secure space) happens at a random time. The problem is to detect the event as early as possible with an optimal sleep –wake scheduling of sensors subject to a false alarm constraint.

We consider a discrete time system and the basic unit of time is one slot. The slots are indexed by non–negative integers. A time slot is assumed to be of unit length, and hence, slot k refers to the time interval $[k, k+1)$. We assume that the sensor network is time synchronised (see, [Solis et al., 2006] for achieving time synchrony). An event occurs at a random time $T \in \mathbb{Z}_+$ and persists from there on for all $k \geqslant T$. The prior distribution of T (the time slot at which the event happens) is given by

$$\mathsf{P}\{T = k\} = \begin{cases} \rho, & \text{if } k = 0 \\ (1-\rho)(1-p)^{k-1}p, & \text{if } k > 0, \end{cases}$$

where $0 < p \leqslant 1$ and $0 \leqslant \rho \leqslant 1$ represents the probability that the event happened even before the observations are made. We say that the state of nature, Θ_k is 0 before the occurrence of the event (i.e., $\Theta_k = 0$ for $k < T$) and 1 after the occurrence of the event (i.e., $\Theta_k = 1$ for $k \geqslant T$).

At any time $k \in \mathbb{Z}_+$, the state of nature Θ_k can not be observed directly and can be observed only partially through the sensor observations. The observations are obtained sequentially starting from time slot $k = 1$ onwards. Before the event takes place, i.e., for $1 \leqslant k < T$, sensor i observes $X_k^{(i)} \in \mathbb{R}$ the distribution of which is given by $F_0(\cdot)$, and after the event takes place, i.e., for $k \geqslant T$, sensor i observes $X_k^{(i)} \in \mathbb{R}$ the distribution of which is given by $F_1(\cdot) \neq F_0(\cdot)$ (because of the small extent network, at time T, the observations of all the sensors switch their distribution to the postchange distribution $F_1(\cdot)$). The corresponding probability density functions (pdfs) are given by $f_0(\cdot)$ and $f_1(\cdot) \neq f_0(\cdot)$[1]. Conditioned on the state of nature, i.e., given the change point T, the observations $X_k^{(i)}$s are independent across sensor nodes and across time. The event and the observation models are essentially the same as in the classical change detection problem, [Shiryaev, 1978] and [Veeravalli, 2001].

The observations are transmitted to a fusion centre. The communication between the sensors and the fusion centre is assumed to be error–free and completes before the next measurements are taken[2]. At time k, let $\mathcal{M}_k = \{i_{k,1}, i_{k,2}, \cdots, i_{k,M_k}\} \subseteq \{1, 2, \cdots, n\}$

[1] If the observations are quantised, one can work with probability mass functions instead of pdfs.

[2] This could be achieved by synchronous time division multiple access, with robust modulation and coding. For a formulation that incorporates a random access network (but not sleep –wake scheduling), see [Prasanthi and Kumar, 2006] and [Premkumar et al.].

be the set of sensor nodes that are in the awake state, and the fusion centre receives a vector of M_k observations $\mathbf{Y}_k = \mathbf{X}_k^{\mathcal{M}_k} := \left[X_k^{(i_{k,1})}, X_k^{(i_{k,2})}, \cdots, X_k^{(i_{k,M_k})}\right]$. At time slot k, based on the observations so far $\mathbf{Y}_{[1:k]}$,[3] the distribution of T, $f_0(\cdot)$, and $f_1(\cdot)$, the fusion centre

1. makes a decision on whether to raise an alarm or to continue sampling, and

2. if it decides to continue sampling, it determines the number of sensors that must be in the awake state in time slot $k+1$.

Let $D_k \in \{0,1\}$ be the decision made by the fusion centre to "continue sampling" in time slot $k+1$ (denoted by 0) or "stop and raise an alarm" (denoted by 1). If $D_k = 0$, the fusion centre controls the set of sensors to be in the awake state in time slot $k+1$, and if $D_k = 1$, the fusion centre chooses $\mathcal{M}_{k+1} = \emptyset$. Let $A_k \in \mathcal{A}$ be the decision (or control or action) made by the fusion centre after having observed $\mathbf{Y}_k$ at time k. We note that A_k also includes the decision D_k. Also, the action space $\mathcal{A}$ depends on the feedback strategy between the fusion centre and the sensor nodes which we discuss in detail in Section 3.3. Let $\mathbf{I}_k := [\mathbf{Y}_{[1:k]}, A_{[0,k-1]}]$ be the information available to the decision maker at the beginning of slot k. The action or control A_k chosen at time k depends on the information $\mathbf{I}_k$ (i.e., A_k is $\mathbf{I}_k$ measurable).

The costs involved are i) λ_s, the cost due to (sampling + computation + communication) per observation per sensor, ii) λ_f, the cost of false alarm, and iii) the detection delay, defined as the delay between the occurrence of the event and the detection, i.e., $(\tau - T)^+$, where τ is the time instant at which the decision maker stops sampling and raises an alarm[4]. Let $c_k : \{0,1\} \times \{(0,0),(0,1),\cdots,(0,n),(1,0)\} \to \mathbb{R}_+$ be the cost incurred at time slot k. For $k \leqslant \tau$, the one step cost function is defined (when the state of nature is Θ_k, the decision made is D_k, and the number of sensors in the awake state in the next

[3] The notation $Y_{[k_1:k_2]}$ defined for $k_1 \le k_2$ means the vector $[Y_{k_1}, Y_{k_1+1}, \cdots, Y_{k_2}]$.

[4] We note here that the event $\{\tau = k\}$ is completely determined by the information $\mathbf{I}_k$, and hence, τ is a stopping time with respect to the sequence of random variables $\mathbf{I}_1, \mathbf{I}_2, \cdots$.

time slot is M_{k+1}) as

$$c_k(\Theta_k, D_k, M_{k+1}) \;\; := \;\; \begin{cases} \lambda_s M_{k+1}, & \text{if } \Theta_k = 0, D_k = 0 \\ \lambda_f, & \text{if } \Theta_k = 0, D_k = 1 \\ 1 + \lambda_s M_{k+1}, & \text{if } \Theta_k = 1, D_k = 0 \\ 0, & \text{if } \Theta_k = 1, D_k = 1 \end{cases} \tag{3.1}$$

and for $k > \tau$, $c_k(\cdot, \cdot, \cdot) := 0$. Note that in the above definition of the cost function, if the decision D_k is 1, then M_{k+1} is always 0. The cost $c_k(\Theta_k, D_k, M_{k+1})$ can be written as

$$\begin{aligned} & c_k(\Theta_k, D_k, M_{k+1}) \\ = & \begin{cases} \lambda_f \cdot \mathbf{1}_{\{\Theta_k=0\}} \mathbf{1}_{\{D_k=1\}} + \big(\mathbf{1}_{\{\Theta_k=1\}} + \lambda_s M_{k+1}\big) \mathbf{1}_{\{D_k=0\}}, & \text{if } k \leqslant \tau \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{3.2}$$

We are interested in obtaining a quickest detection procedure that minimises the mean detection delay and the cost of observations by sensor nodes in the awake state subject to the constraint that the probability of false alarm is bounded by α, a desired quantity. We thus have a constrained optimization problem,

$$\begin{aligned} \text{minimise} \quad & \mathsf{E}\left[(\tau - T)^+ + \lambda_s \sum_{k=1}^{\tau} M_k\right] \\ \text{subject to} \quad & \mathsf{P}\left\{\tau < T\right\} \leqslant \alpha \end{aligned} \tag{3.3}$$

where τ is a stopping time with respect to the sequence $\mathbf{I}_1, \mathbf{I}_2, \cdots$ (i.e., $\tau \in \sigma(\{\mathbf{I}_1, \mathbf{I}_2, \cdots\})$, the stopping time τ is measurable with respect to the sigma field generated by the random variables, $\mathbf{I}_1, \mathbf{I}_2, \cdots$). The above problem would also arise if we imposed a total energy constraint on the sensors until the stopping time (in which case, λ_s can be thought of as the *Lagrange multiplier* that relaxes the energy constraint). Let λ_f be the cost of false alarm. The expected total cost (or the Bayes risk) when the stopping time is τ is given by

$$\begin{aligned}
R(\tau) &= \lambda_f \mathsf{P}\left\{\tau < T\right\} + \mathsf{E}\left[(\tau - T)^+ + \lambda_s \sum_{k=1}^{\tau} M_k\right] \\
&= \mathsf{E}\left[\lambda_f \mathbf{1}_{\{\Theta_\tau = 0\}} + \sum_{k=0}^{\tau-1} \left(\mathbf{1}_{\{\Theta_k = 1\}} + \lambda_s M_{k+1}\right)\right] \\
&= \mathsf{E}\left[c_\tau(\Theta_\tau, 1, 0) + \sum_{k=0}^{\tau-1} c_k(\Theta_k, 0, M_{k+1})\right] \\
&= \mathsf{E}\left[\sum_{k=0}^{\tau} c_k(\Theta_k, D_k, M_{k+1})\right] \\
&\overset{(a)}{=} \mathsf{E}\left[\sum_{k=0}^{\infty} c_k(\Theta_k, D_k, M_{k+1})\right] \\
&\overset{(b)}{=} \sum_{k=0}^{\infty} \mathsf{E}[c_k(\Theta_k, D_k, M_{k+1})] \qquad (3.4)
\end{aligned}$$

where step (a) follows from $c_k(\cdot, \cdot, \cdot) = 0$ for $k > \tau$, and step (b) follows from the monotone convergence theorem. Note that λ_f is a Lagrange multiplier and is chosen such that the false alarm constraint is satisfied with equality, i.e., $\mathsf{P}_{\mathsf{FA}} = \alpha$ (see [Shiryaev, 1978]).

We note that the stopping time τ is related to the control sequence $\{A_k\}$ in the following manner. For any stopping time τ, there exists a sequence of functions (also called a policy) $\nu = (\nu_1, \nu_2, \cdots)$ such that for any k, when $\tau = k$, $D_{k'} = \nu_{k'}(\mathbf{I}_{k'}) = 0$ for all $k' < k$ and $D_{k'} = \nu_{k'}(\mathbf{I}_{k'}) = 1$ for all $k' \geqslant k$. Thus, the unconstrained expected cost given by Eqn. 3.4 is

$$\begin{aligned}
R(\tau) = \sum_{k=0}^{\infty} \mathsf{E}[c_k(\Theta_k, D_k, M_{k+1})] &= \sum_{k=0}^{\infty} \mathsf{E}[c_k(\Theta_k, \nu_k(\mathbf{I}_k), M_{k+1})] \\
&= \sum_{k=0}^{\infty} \mathsf{E}[\mathsf{E}[c_k(\Theta_k, \nu_k(\mathbf{I}_k), M_{k+1}) \mid \mathbf{I}_k]] \\
&\overset{(a)}{=} \mathsf{E}\left[\sum_{k=0}^{\infty} \mathsf{E}[c_k(\Theta_k, \nu_k(\mathbf{I}_k), M_{k+1}) \mid \mathbf{I}_k]\right] \qquad (3.5) \\
&= \mathsf{E}\left[\sum_{k=0}^{\tau} \mathsf{E}[c_k(\Theta_k, \nu_k(\mathbf{I}_k), M_{k+1}) \mid \mathbf{I}_k]\right]
\end{aligned}$$

where step (a) above follows from the monotone convergence theorem. From Eqn. 3.2, it is clear that for $k \leqslant \tau$

$$\begin{aligned}
& \mathsf{E}[c_k(\Theta_k, \nu_k(\mathbf{I}_k), M_{k+1}) \mid \mathbf{I}_k] \\
= \;& \mathsf{E}\big[\lambda_f \cdot \mathbf{1}_{\{\Theta_k=0\}} \cdot \mathbf{1}_{\{\nu_k(\mathbf{I}_k)=1\}}\big] + \mathsf{E}\big[\big(\mathbf{1}_{\{\Theta_k=1\}} + \lambda_s M_{k+1}\big) \cdot \mathbf{1}_{\{\nu_k(\mathbf{I}_k)=0\}} \mid \mathbf{I}_k\big] \\
= \;& \lambda_f \cdot \mathsf{E}\big[\mathbf{1}_{\{\Theta_k=0\}} \mid \mathbf{I}_k\big] \cdot \mathbf{1}_{\{\nu_k(\mathbf{I}_k)=1\}} + \big(\mathsf{E}\big[\mathbf{1}_{\{\Theta_k=1\}} \mid \mathbf{I}_k\big] + \lambda_s \cdot \mathsf{E}[M_{k+1} \mid \mathbf{I}_k]\big) \cdot \mathbf{1}_{\{\nu_k(\mathbf{I}_k)=0\}}
\end{aligned}$$

For $k \leqslant \tau$, define the a posteriori probability of the change having occurred at or before time slot k, $\Pi_k := \mathsf{E}\big[\mathbf{1}_{\{\Theta_k=1\}} \,\big|\, \mathbf{I}_k\big]$, and hence, we have

$$\mathsf{E}[c_k(\Theta_k, \nu_k(\mathbf{I}_k), M_{k+1}) \mid \mathbf{I}_k] = \lambda_f \cdot (1 - \Pi_k)\mathbf{1}_{\{\nu_k(\mathbf{I}_k)=1\}} + (\Pi_k + \lambda_s \cdot \mathsf{E}[M_{k+1} \mid \mathbf{I}_k])\, \mathbf{1}_{\{\nu_k(\mathbf{I}_k)=0\}} \tag{3.6}$$

Thus, we can write the Bayesian risk given in Eqn. 3.5 as

$$R(\tau) = \mathsf{E}\left[\lambda_f \cdot (1 - \Pi_\tau) + \sum_{k=0}^{\tau-1} (\Pi_k + \lambda_s \mathsf{E}[M_{k+1} \mid \mathbf{I}_k])\right] \tag{3.7}$$

We are interested in obtaining an optimal stopping time τ and an optimal control of the number of sensors in the awake state. Thus, we have the following problem,

$$\text{minimise } \mathsf{E}\left[\lambda_f \cdot (1 - \Pi_\tau) + \sum_{k=0}^{\tau-1} (\Pi_k + \lambda_s \mathsf{E}[M_{k+1} \mid \mathbf{I}_k])\right] \tag{3.8}$$

We consider the following possibilities for the problem defined in Eqn. 3.8.

1. **Closed loop control on M_{k+1}**: At time slot k, the fusion centre makes a decision on M_{k+1}, the number of sensors in the awake state in time slot $k+1$, based on the information available (at the fusion centre) up to time slot k. The decision is then fed back to the sensors via a feedback channel. Thus, the problem becomes

$$\min_{\tau, M_1, M_2, \cdots, M_\tau} \mathsf{E}\left[\lambda_f(1 - \Pi_\tau) + \sum_{k=0}^{\tau-1} (\Pi_k + \lambda_s M_{k+1})\right] \tag{3.9}$$

2. **Closed loop control on** q_{k+1}: At time slot k, the fusion centre makes a decision on q_{k+1}, the probability that a sensor is in the awake state at time slot $k+1$ based on the information $\mathbf{I}_k$. q_{k+1} is then broadcast via a feedback channel to the sensors. Thus, given $\mathbf{I}_k$, the number of sensors in the awake state M_{k+1}, at time slot $k+1$, is Bernoulli distributed with parameters (n, q_{k+1}) and $\mathsf{E}[M_{k+1} \mid \mathbf{I}_k] = nq_{k+1}$. Thus, the problem defined in Eqn. 3.8 becomes

$$\min_{\tau, q_1, q_2 \cdots, q_\tau} \mathsf{E}\left[\lambda_f(1-\Pi_\tau) + \sum_{k=0}^{\tau-1}(\Pi_k + \lambda_s n q_{k+1})\right] \tag{3.10}$$

3. **Open loop control on** q: Here, there is no feedback between fusion centre and the sensor nodes. At time slot k, each sensor node is in the awake state with probability q. Note that M_k, the number of sensors in the awake state at time slot k is Bernoulli distributed with parameters (n, q). Also note that $\{M_k\}$ process is i.i.d. and $\mathsf{E}[M_{k+1} \mid \mathbf{I}_k] = nq$ (also, M_{k+1} is independent of the information vector $\mathbf{I}_k$). Note that *the probability q is constant over time.* Thus, the problem defined in Eqn. 3.8 becomes

$$\min_{\tau} \mathsf{E}\left[\lambda_f(1-\Pi_\tau) + \sum_{k=0}^{\tau-1}(\Pi_k + \lambda_s n q)\right] \tag{3.11}$$

Here, q is chosen (at time $k = 0$) such that it minimises the above cost.

Note that the first two scenarios require a feedback channel between the fusion centre and the sensors whereas the last scenario does not require a feedback channel.

In Section 3.3, we formulate the optimization problem defined in Eqns. 3.9 and 3.10 in the framework of MDP and study the optimal closed loop sleep –wake scheduling policies. In Section 3.4, we formulate the optimization problem defined in Eqn. 3.11 in the MDP framework and obtain the optimal probability q of a sensor in the awake state.

3.3 Quickest Change Detection with Feedback

In this section, we study the sleep –wake scheduling problem when there is feedback from the fusion centre to the sensors.

At time slot k, the fusion centre receives a M_k–vector of observations $\mathbf{Y}_k$, and computes Π_k. Recall that $\Pi_k = \mathsf{P}\left\{T \leqslant k \,\middle|\, \mathbf{I}_k\right\}$ is the a posteriori probability of the event having occurred at or before time slot k. For the event detection problem, a sufficient statistic for the sensor observations at time slot k is given by Π_k (see [Lehmann and Casella, 1998] and page 244, [Bertsekas, 2000a]). When an *alarm* is raised, the system enters into a terminal state 't'. Thus, the state space of the $\{\Pi_k\}$ process is $\mathcal{S} = [0,1] \cup \{\mathsf{t}\}$. Note that Π_k is also called the *information state* of the system.

In the rest of the section, we explain the MDP formulation that yields the closed loop sleep –wake scheduling algorithms.

3.3.1 Control on the number of sensors in the awake state

In this subsection, we are interested in obtaining an optimal control on M_{k+1}, the number of sensors in the awake state, based on the information we have at time slot k.

At time slot k, after having observed $\mathbf{X}_k^{\mathcal{M}_k}$, the fusion centre computes the sufficient statistic Π_k. Based on Π_k, the fusion centre makes a decision to stop or to continue sampling. If the decision is to continue at time slot $k+1$, the fusion centre (which also acts as a controller) chooses M_{k+1}, the number of sensors to be in the awake state at time slot $k+1$. The fusion centre also keeps track of the residual energy in the sensor nodes, based on which it chooses the set of sensor nodes $\mathcal{M}_{k+1}$ that must be in the awake state in time slot $k+1$. Since, the prechange and the postchange pdfs of the observations are the same for all the sensor nodes and at any time, the sensor observations are conditionally independent across sensors, any observation vector of size m has the same pdf and hence, for decision making, it is sufficient to look at only the number of sensors in the awake state M_{k+1}, i.e., the indices of the sensor nodes that are in the awake state are not required for detection (we assume that the fusion centre chooses the sequence $\mathcal{M}_1, \mathcal{M}_2, \cdots$ in such a way that the rate at which the sensor nodes drain their energy is the same). Thus, the set of controls at time slot k is given by $\mathcal{A} = \left\{(\mathsf{stop}, 0), \bigcup_{m\in\{0,1,\cdots,n\}}(\mathsf{continue}, m)\right\} = \big\{(1,0),(0,0),(0,1),\cdots,(0,n)\big\}$.

We show that Π_k can be computed in a recursive manner from the previous state Π_{k-1}, the previous action A_{k-1}, and the current observation $\mathbf{X}_k^{\mathcal{M}_k}$ as,

$$\begin{aligned}\Pi_k &= \Phi(\Pi_{k-1}, A_{k-1}, \mathbf{X}_k^{\mathcal{M}_k}) \\ &:= \begin{cases} \mathsf{t}, & \text{if } \Pi_{k-1} = \mathsf{t} \\ \mathsf{t}, & \text{if } A_{k-1} = 1 \\ \frac{\widetilde{\Pi}_{k-1}\phi_1\left(\mathbf{X}_k^{\mathcal{M}_k}\right)}{\phi_2\left(\mathbf{X}_k^{\mathcal{M}_k};\widetilde{\Pi}_{k-1}\right)}, & \text{if } \Pi_{k-1}\in[0,1], A_{k-1} = (0, M_k) \end{cases}\end{aligned} \tag{3.12}$$

where

$$\begin{aligned}\widetilde{\Pi}_k &:= \Pi_k + (1-\Pi_k)p, \\ \phi_0\left(\mathbf{X}_k^{\mathcal{M}_k}\right) &:= \prod_{i\in\mathcal{M}_k} f_0(X_k^{(i)}), \\ \phi_1\left(\mathbf{X}_k^{\mathcal{M}_k}\right) &:= \prod_{i\in\mathcal{M}_k} f_1(X_k^{(i)}), \\ \phi_2\left(\mathbf{X}_k^{\mathcal{M}_k};\widetilde{\Pi}\right) &:= \widetilde{\Pi}\phi_1\left(\mathbf{X}_k^{\mathcal{M}_k}\right) + (1-\widetilde{\Pi})\phi_0\left(\mathbf{X}_k^{\mathcal{M}_k}\right)\end{aligned} \tag{3.13}$$

Thus, the a posteriori probability process $\{\Pi_k\}$ is a controlled Markov process. Note that $\widetilde{\Pi}_k = \Pi_k + (1-\Pi_k)p = \mathsf{E}[\Pi_{k+1}]$ before $\mathbf{X}_{k+1}^{\mathcal{M}_{k+1}}$ is observed. Motivated by the structure of the cost given in Eqn. 3.6, we define the one stage cost function $\widetilde{c} : \mathcal{S}\times\mathcal{A}\to\mathbb{R}_+$ when the (state, action) pair is (s,a) as

$$\widetilde{c}(s,a) = \begin{cases} \lambda_f\left(1-\pi\right), & \text{if } s = \pi\in[0,1], a = (1,0) \\ \pi + \lambda_s m, & \text{if } s = \pi\in[0,1], a = (0,m) \\ 0, & \text{if } s = \mathsf{t}. \end{cases}$$

Since M_{k+1} is chosen based on the information $\mathbf{I}_k$, there exists a function ν_k' such that $M_{k+1} = \nu_k'(\mathbf{I}_k)$. Thus, the action or control at time k is given by $\mu_k(\mathbf{I}_k) = (\nu_k(\mathbf{I}_k), \nu_k'(\mathbf{I}_k))$.

Hence, we can write the Bayesian risk given in Eqn. 3.4 for a policy $\mu = (\mu_1, \mu_2, \cdots)$ as

$$\begin{aligned} R(\tau) &= \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{c}\,(\Pi_k, \mu_k(\mathbf{I}_k))\right] \\ &= \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{c}\,(\Pi_k, \widetilde{\mu}_k(\Pi_k))\right] \qquad (3.14) \end{aligned}$$

Since Π_k is a sufficient statistic for $\mathbf{I}_k$, for any policy μ_k there exists a corresponding policy $\widetilde{\mu}_k$ such that $\widetilde{\mu}_k(\Pi_k) = \mu_k(\mathbf{I}_k)$, and hence, the last step in the above equation follows (see page 244, [Bertsekas, 2000a]) Since, the one stage cost and the density function $\phi_2(\mathbf{y}; \widetilde{\Pi}_{k-1})$ are time invariant, it is sufficient to consider the class of stationary policies (see Proposition 2.2.2 of [Bertsekas, 2007]). Let $\widetilde{\mu} : \mathcal{S} \to \mathcal{A}$ be a stationary policy. Hence, the cost of using the policy $\widetilde{\mu}$ is given by

$$J_{\widetilde{\mu}}(\pi_0) = \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{c}(\Pi_k, \widetilde{\mu}(\Pi_k)) \,\middle|\, \Pi_0 = \pi_0\right],$$

and hence, the minimal cost among the class of stationary policies is given by

$$J^*(\pi_0) = \min_{\widetilde{\mu}} \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{c}(\Pi_k, \widetilde{\mu}(\Pi_k)) \,\middle|\, \Pi_0 = \pi_0\right].$$

The dynamic program (DP) that solves the above problem is given by the Bellman's equation,

$$J^*(\pi) = \min\left\{\widetilde{c}(\pi, 1), H_{J^*}(\pi)\right\} \qquad (3.15)$$

where the function $H_{J^*} : [0,1] \to \mathbb{R}_+$ is defined as

$$H_{J^*}(\pi) := \min_{0 \leqslant m \leqslant n} \left\{\widetilde{c}(\pi, (0,m)) + \mathsf{E}_{\phi_2(\mathbf{y};\tilde{\pi})}\left[J^*\left(\Phi(\pi, (0,m), \mathbf{Y})\right)\right]\right\} \qquad (3.16)$$

where $\mathbf{Y}$ and $\mathbf{y}$ are m–vectors. The notation $\mathsf{E}_{\phi_2(\mathbf{y};\tilde{\pi})}[\cdot]$ means that the expectation is taken with respect to the pdf $\phi_2(\mathbf{y}; \tilde{\pi})$ (recall Eqn. 3.13 for the definition of $\phi_2(\mathbf{y}; \tilde{\pi})$).

Thus, Eqn. 3.15 can be written as

$$J^*(\pi) \quad = \quad \min\left\{\lambda_f \cdot \big(1-\pi\big), \pi + A_{J^*}\big(\pi\big)\right\} \tag{3.17}$$

where the function $A_{J^*} : [0,1] \to \mathbb{R}_+$ is defined as

$$A_{J^*}(\pi) = \min_{0 \leqslant m \leqslant n} \left\{\lambda_s m + \mathsf{E}_{\phi_2(\mathbf{y};\tilde{\pi})}\left[J^*\left(\frac{\tilde{\pi}\cdot\phi_1(\mathbf{Y})}{\phi_2(\mathbf{Y};\tilde{\pi})}\right)\right]\right\} \tag{3.18}$$

The optimal policy μ^* that achieves J^* gives the optimal stopping rule, τ^*, and the optimal number of sensors in the awake state, $M_1^*, M_2^*, \cdots, M_{\tau^*}^*$.

We now establish some properties of the *minimum* total cost function J^*.

Theorem 3.1 *The total cost function $J^*(\pi)$ is concave in π.*

Also, we establish some properties of the optimal policy μ^* (which maps the a posteriori probability of change Π_k to the action space $\mathcal{A}$) in the next theorem.

Theorem 3.2 *The optimal stopping rule is given by the following threshold rule where the threshold is on the a posteriori probability of change,*

$$\tau^* \quad = \quad \inf\{k : \Pi_k \geqslant \Gamma\}, \tag{3.19}$$

for some $\Gamma \in [0,1]$. The threshold Γ depends on the probability of false alarm constraint, α (among other parameters like the distribution of T, f_0, f_1).

Theorem 3.2 addresses only the *stopping time* part of the optimal policy μ^*. We now explore the structure of the optimal closed loop control policy for $M^* : [0,1] \to \mathbb{Z}_+$, the optimal number of sensors in the awake state in the *next* time slot. At time k, based on the (sufficient) statistic Π_k, the fusion centre chooses $M_{k+1}^* = M^*(\Pi_k)$ number of sensor nodes in the awake state. For each $0 \leqslant m \leqslant n$, we define the functions $B_{J^*}^{(m)} : [0,1] \to \mathbb{R}_+$

and $A_{J^*}^{(m)} : [0,1] \to \mathbb{R}_+$ as

$$\begin{aligned} B_{J^*}^{(m)}(\pi) &:= \mathsf{E}_{\phi_2(\mathbf{y};\tilde{\pi})}\left[J^*\left(\frac{\tilde{\pi}\cdot\phi_1(\mathbf{Y})}{\phi_2(\mathbf{Y};\tilde{\pi})}\right)\right], \\ \text{and } A_{J^*}^{(m)}(\pi) &:= \lambda_s m + B_{J^*}^{(m)}(\pi). \end{aligned}$$

We have shown in the proof of Theorem 3.1 that for any $m = 0, 1, 2, \cdots, n$, the functions $B_{J^*}^{(m)}(\pi)$ and $A_{J^*}^{(m)}(\pi)$ are concave in π.

Theorem 3.3 *For any $\pi \in [0,1]$, the functions $B_{J^*}^{(m)}(\pi)$ monotonically decrease with m.*

Remark: By increasing the number of sensor nodes in the awake state, i.e., by increasing m, we expect that the a posteriori probability of change will get closer to 1 or closer to 0 (depending on whether the change has occurred or not). In either case, the one stage cost decreases, and hence, we expect that the functions $B_{J^*}^{(m)}(\pi)$ monotonically decrease with m.

At time k, $B_{J^*}^{(m)}(\Pi_k)$ can be thought of as the cost–to–go function from slot $k+1$ onwards (having used m sensor nodes at time $k+1$). Note that $A_{J^*}^{(m)}(\pi)$ has two components, the first component $\lambda_s m$ increases with m and (from Theorem 3.3) the second component decreases with m. As m takes values in a finite set $\{0, 1, 2, \cdots, n\}$, for each π, there exists an optimal $M^*(\pi)$ for which $A_{J^*}^{(M^*(\pi))}(\pi)$ is minimum. For any given $\pi \in [0,1]$, we define the differential cost $d : \{1, 2, \cdots, n\} \to \mathbb{R}_+$ as

$$d(m;\pi) = B_{J^*}^{(m-1)}(\pi) - B_{J^*}^{(m)}(\pi). \tag{3.20}$$

Note that for any $1 \leqslant m \leqslant n$, $d(m;\pi)$ is bounded and continuous in π (as $B_{J^*}^{(m)}$s are bounded and concave in π). Also note that $d(m;1) = 0$ as $B_{J^*}^{(m-1)}(1) = B_{J^*}^{(m)}(1) = 0$. We are interested in $d(m;\pi)$ for $\pi \in [0,\ \Gamma)$. In Figure 3.1, we plot $d(m;\pi)$ against π for $m = 1, 2,$ and 3 (for the set of parameters $n = 10$, $\lambda_f = 100$, $\lambda_s = 0.5$, and f_0 and f_1 are unit variance Gaussian pdfs with means 0 and 1 respectively; also $d(m;\pi)$ is computed from $B_{J^*}^{(m)}(\pi)$ which in turn is obtained by averaging through 100 simulation runs using $J^*(\cdot)$ which we will obtain in Section 3.5). We observe that $d(m;\pi)$ monotonically

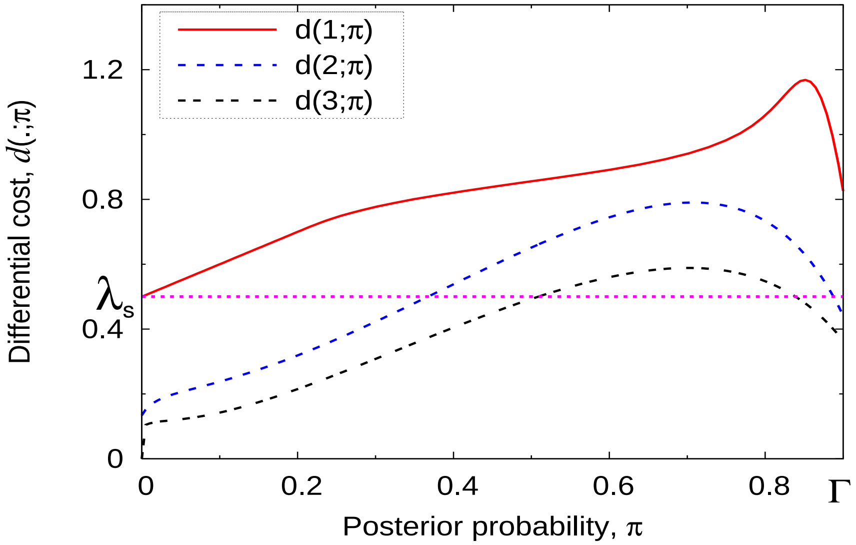


Figure 3.1: Differential costs, $d(\cdot;\pi)$, for $n = 10$ sensors, $\lambda_f = 100.0$, $\lambda_s = 0.5$, $f_0 \sim \mathcal{N}(0,1)$ and $f_1 \sim \mathcal{N}(1,1)$.

decreases in m, for each $\pi \in [0,\Gamma)$ (i.e., $d(1;\pi) \geqslant d(2;\pi) \geqslant d(3;\pi)$). We have observed this monotonicity property for different sets of experiments for the case when f_0 and f_1 belong to the Gaussian class of distributions. We conjecture that this monotonicity property of d holds and state the following theorem which gives a *structure* for M^*, the optimal number of sensors in the awake state.

Theorem 3.4 *If for each $\pi \in [0,\Gamma)$, $d(m;\pi)$ decreases monotonically in m, then the optimal number of sensors in the* awake *state, $M^* : [0,1] \to \{0,1,\cdots,n\}$ is given by*

$$M^*(\pi) \quad = \quad \max\left\{m : d(m;\pi) \geqslant \lambda_s\right\}.$$

3.3.2 Control on the probability of a sensor in the awake state

In this subsection, we are interested in obtaining an optimal control on q_{k+1}, the probability of a sensor in the awake state, based on the information we have at time slot k, instead of determining the number of sensors that must be in the awake state in the next slot.

At time slot k, after having observed $\mathbf{X}_k^{\mathcal{M}_k}$, the fusion centre computes the sufficient statistic Π_k, based on which it makes a decision to $\textsf{stop}$ or to $\textsf{continue}$ sampling. If the decision is to $\textsf{continue}$ at time slot $k+1$, the fusion centre (also acts as a controller) chooses q_{k+1}, the probability of a sensor to be in the $\textsf{awake}$ state at time slot $k+1$. Thus, the set of controls at time slot k is given by $\mathcal{A} = \left\{(\textsf{stop}, 0), \cup_{q\in[0,1]}(\textsf{continue}, q)\right\}$ $= \left\{1, \cup_{q\in[0,1]}(0,q)\right\} = \{(1,0), \{0\}\times[0,1]\}$.

When the control $A_k = (0, q_{k+1})$ is chosen, M_{k+1}, the number of sensors in the $\textsf{awake}$ state at time slot $k+1$ is *Bernoulli* distributed with parameters (n, q_{k+1}). Let $\gamma_m(q_{k+1})$ be the probability that m sensors are in the $\textsf{awake}$ state at time slot $k+1$. $\gamma_m(q_{k+1})$ is given by

$$\gamma_m(q_{k+1}) = \binom{n}{m} q_{k+1}^m (1-q_{k+1})^{n-m}. \tag{3.21}$$

The information state at time slot k Π_k, can be computed in a recursive manner from Π_{k-1}, A_{k-1} and $\mathbf{X}_k^{\mathcal{M}_k}$ using Eqn. 3.12. Thus, it is clear that the $\{\Pi_k\}$ process is a controlled Markov process, the state space of the process being $\mathcal{S} = [0,1]\cup\{\mathsf{t}\}$. Motivated by the cost function given in Eqn. 3.6, define the one stage cost function $\widetilde{c}(s,a)$ when the (state,action) pair is (s,a) as

$$\widetilde{c}(s,a) = \begin{cases} \lambda_f(1-\pi), & \text{if } s=\pi\in[0,1], a=(1,0) \\ \pi + \lambda_s nq, & \text{if } s=\pi\in[0,1], a=(0,q) \\ 0, & \text{if } s=\mathsf{t}. \end{cases}$$

Since, the one stage cost and the density function $\phi_2(\mathbf{y}; \widetilde{\Pi}_{k-1})$ are time invariant, it is sufficient to consider the class of stationary policies (see Proposition 2.2.2 of [Bertsekas, 2007]). Let $\widetilde{\mu} : \mathcal{S} \to \mathcal{A}$ be a stationary policy. Hence, the cost of using the policy $\widetilde{\mu}$ is given by

$$J_{\widetilde{\mu}}(\pi_0) = \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{c}(\Pi_k, \widetilde{\mu}(\Pi_k)) \,\middle|\, \Pi_0 = \pi_0\right],$$

and hence the minimal cost among the class of stationary policies is given by

$$J^*(\pi_0) \quad = \quad \min_{\widetilde{\mu}} \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{c}(\Pi_k, \widetilde{\mu}(\Pi_k)) \,\middle|\, \Pi_0 = \pi_0\right].$$

The DP that solves the above problem is given by the Bellman's equation,

$$J^*(\pi) \quad = \quad \min\left\{\widetilde{c}(\pi,1), H_{J^*}(\pi)\right\}$$

where $H_{J^*} : [0,1] \to \mathbb{R}_+$ is defined as

$$H_{J^*}(\pi) \quad := \quad \min_{0 \leqslant q \leqslant 1} \left\{\widetilde{c}(\pi,(0,q)) + \sum_{m=0}^{n} \gamma_m(q) \mathsf{E}_{\phi_2(\mathbf{y};\tilde{\pi})}\left[J^*\big(\Phi(\pi,(0,m),\mathbf{Y})\big]\right\}$$

where $\mathbf{Y}$ and $\mathbf{y}$ are m–vectors. Recall that the expectation is taken with respect to the pdf $\phi_2(\mathbf{y};\tilde{\pi})$. The Bellman's equation can be written as

$$J^*(\pi) \quad = \quad \min\left\{\lambda_f \cdot \big(1-\pi\big), \pi + A_{J^*}\big(\pi\big)\right\} \tag{3.22}$$

where the function $A_{J^*} : [0,1] \to \mathbb{R}_+$ is defined as

$$A_{J^*}(\pi) \quad = \quad \min_{q \in [0,1]} \left\{\lambda_s n q + \sum_{m=0}^{n} \gamma_m(q) \mathsf{E}_{\phi_2(\mathbf{y};\tilde{\pi})}\left[J^*\left(\frac{\tilde{\pi} \cdot \phi_1(\mathbf{Y})}{\phi_2(\mathbf{Y};\tilde{\pi})}\right)\right]\right\}.$$

The optimal policy μ^* gives the optimal stopping time τ^*, and the optimal probabilities, $q_k^*,\ k \ = \ 1,2,\cdots,\tau^*$. The structure of the optimal policy is shown in the following theorems.

Theorem 3.5 *The total cost function $J^*(\pi)$ is concave in π.*

Theorem 3.6 *The optimal stopping rule is a threshold rule where the threshold is on the a posteriori probability of change,*

$$\tau^* \;=\; \inf\{k : \Pi_k \geqslant \Gamma\},$$

for some $\Gamma \in [0,1]$*. The threshold* Γ *depends on the probability of false alarm constraint,* α *(among other parameters like the distribution of* T*,* f_0*,* f_1*).*

3.4 Quickest Change Detection without Feedback

In this section, we study the sleep –wake scheduling problem defined in Eqn. 3.11. Open loop control is applicable to the systems in which there is no feedback channel from the fusion centre (controller) to the sensors. Here, at any time slot k, a sensor chooses to be in the awake state with probability q independent of other sensors. Hence, $\{M_k\}$, the number of sensors in the awake state at time slot k is i.i.d. *Bernoulli distributed* with parameters (n,q). Let γ_m be the probability that m sensors are in the awake state. γ_m is given by

$$\gamma_m \;=\; \binom{n}{m} q^m (1-q)^{n-m} \tag{3.23}$$

We choose q that minimises the Bayesian cost given by Eqn. 3.11.

At time slot k, the fusion centre receives a vector of observation $\mathbf{X}_k^{\mathcal{M}_k}$ and computes Π_k. In the open loop scenario, the state space is $\mathcal{S} = \big\{[0,1] \cup \{\mathsf{t}\}\big\}$. The set of actions is given by $\mathcal{A} = \{\mathsf{stop}, \mathsf{continue}\} = \{1, 0\}$ where '1' represents stop and '0' represents continue. Note that Π_k can be computed from Π_{k-1}, A_{k-1}, and $\mathbf{X}_k^{\mathcal{M}_k}$ in the same way as shown in Eqn. 3.12. Thus, $\{\Pi_k\}$, $k \in \mathbb{Z}_+$ is a controlled Markov process. Motivated by the structure of the cost given in Eqn. 3.6, we define the one stage cost function

$\widetilde{c} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$ when the (state, action) pair is (s, a) as

$$\widetilde{c}(s,a) \quad = \quad \begin{cases} \lambda_f(1-\pi), & \text{if } s = \pi \in [0,1], a = 1 \\ \pi + \lambda_s n q, & \text{if } s = \pi \in [0,1], a = 0 \\ 0, & \text{if } s = \mathsf{t}. \end{cases}$$

Since, the one stage cost and the density function $\phi_2(\mathbf{y}; \widetilde{\Pi}_{k-1})$ are time invariant, it is sufficient to consider the class of stationary policies (see Proposition 2.2.2 of [Bertsekas, 2007]). Let $\widetilde{\mu} : \mathcal{S} \to \mathcal{A}$ be a stationary policy. Hence, the cost of using the policy $\widetilde{\mu}$ is given by

$$J_{\widetilde{\mu}}(\pi_0) \quad = \quad \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{c}(\Pi_k, \widetilde{\mu}(\Pi_k)) \,\middle|\, \Pi_0 = \pi_0\right],$$

and the optimal cost under the class of stationary policies is given by

$$J^*(\pi_0) \quad = \quad \min_{\widetilde{\mu}} \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{c}(\Pi_k, \widetilde{\mu}(\Pi_k)) \,\middle|\, \Pi_0 = \pi_0\right]$$

The DP that solves the above equation is given by the Bellman's equation,

$$J^*(\pi) \quad = \quad \min\left\{\widetilde{c}(\pi, 1), H_{J^*}(\pi)\right\}$$

where $H_{J^*} : [0,1] \to \mathbb{R}_+$ is defined as

$$H_{J^*}(\pi) := \widetilde{c}(\pi, 0) + \sum_{m=0}^{n} \gamma_m \mathsf{E}_{\phi_2(\mathbf{y};\tilde{\pi})}\left[J^*\Big(\Phi(\pi, (0,m), \mathbf{Y})\Big)\right]$$

where $\mathbf{Y}$ and $\mathbf{y}$ are m–vectors. The above equation can be written as

$$J^*(\pi) \quad = \quad \min\left\{\lambda_f \cdot \big(1-\pi\big), \pi + A_{J^*}\big(\pi\big)\right\}. \tag{3.24}$$

where the function $A_{J^*} : [0,1] \to \mathbb{R}_+$ is defined as

$$A_{J^*}(\pi) \;\; = \;\; \lambda_s nq + \sum_{m=0}^{n} \gamma_m \mathsf{E}_{\phi_2(\mathbf{y};\tilde{\pi})}\left[J^*\left(\frac{\tilde{\pi} \cdot \phi_1(\mathbf{Y})}{\phi_2(\mathbf{Y};\tilde{\pi})} \right) \right].$$

The optimal policy μ^* that achieves J^* gives the optimal stopping rule, τ^*. We now prove some properties of the optimal policy.

Theorem 3.7 *The optimal total cost function $J^*(\pi)$ is concave in π.*

Theorem 3.8 *The optimal stopping rule is a threshold rule where the threshold is on the a posteriori probability of change,*

$$\tau^* \;\; = \;\; \inf\{k : \Pi_k \geqslant \Gamma\},$$

for some $\Gamma \in [0,1]$. The threshold Γ depends on the probability of false alarm constraint, α (among other parameters like the distribution of T, f_0, f_1).

For each $q \in [0,1]$, we compute the optimal mean detection delay ADD (as a function of q), and then find the optimal q^* for which the optimal mean detection delay is minimum.

3.5 Numerical Results

We compute the optimal policy for each of the sleep –wake scheduling strategies given in Eqns. 3.17, 3.22, 3.24 using value–iteration technique (see [Bertsekas, 2000a]) for 1000 iterations. We consider $n = 10$ sensors. The distributions of change–time T is taken to be geometric (0.01) (and $\pi_0 = 0$). Also, the prechange and the postchange distributions of the sensor observations are taken to be $\mathcal{N}(0,1)$ and $\mathcal{N}(1,1)$. We set the cost per observation per sensor, λ_s to 0.5 and the cost of false alarm, λ_f to 100.0 (this corresponds to $\alpha = 0.04$).

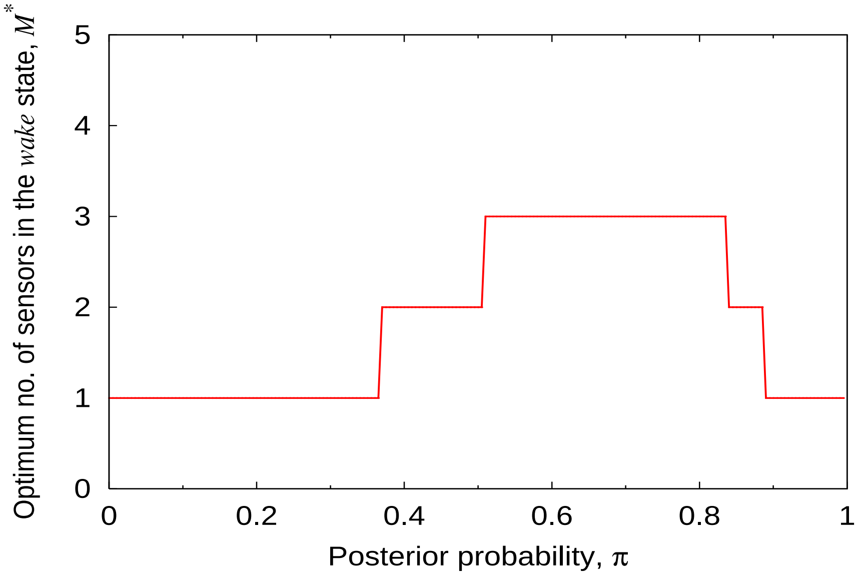


Figure 3.2: Optimum number of sensors in the awake state M^* for $n = 10$ sensors, $\lambda_f = 100.0$, $\lambda_s = 0.5$, $f_0 \sim \mathcal{N}(0,1)$ and $f_1 \sim \mathcal{N}(1,1)$. Note that $\Gamma = 0.9$ corresponds to the threshold.

- **Optimal control of M_{k+1}:**

  We compute M^* the optimal number of sensors to be in the awake state in time slot $k+1$ as a function of the a posteriori probability of change π (from the optimal policy μ^* given by Eqn.3.17) by the *value iteration* algorithm [Bertsekas, 2007], [Hernández-Lerma and Lasserre, 1996], the number of iterations taken being 1000, and plot in Figure 3.2. We note that in any time slot, it is not economical to use more than 3 sensors (though we have 10 sensors). Also, from Figure 3.2, it is clear that M^* increases monotonically for $\pi < 0.6$ and then decreases monotonically for $\pi \geqslant 0.6$. Note that, the region $\pi \in [0.5,\ 0.82]$ requires many sensors for optimal detection whereas the region $[0.0,\ 0.3] \cup [0.9,\ 1.0]$ requires the least number of sensors. This is due to the fact that *uncertainty* (about whether an event has occurred or not) is more in the region $\pi \in [0.5,\ 0.82]$ whereas it is less in the region $[0.0,\ 0.3] \cup [0.9,\ 1.0]$.

  In Figure 3.3, we plot the trajectory of a sample path of Π_k versus the time slot k. In our numerical experiment, the event occurs at $T = 152$. When the number of sensors to be in the awake state M_{k+1} is $M^*(\pi_k)$ (taken from Figure 3.2), for a

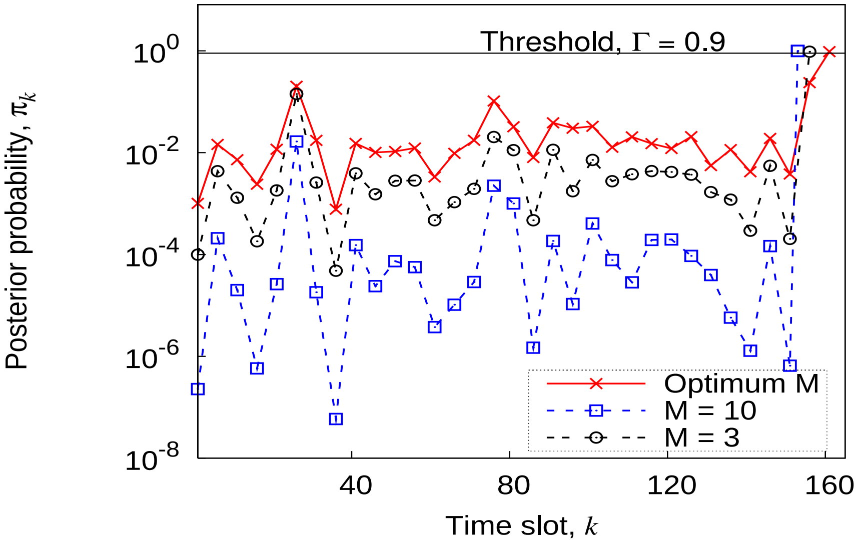


Figure 3.3: A sample run of *event detection* with $n = 10$ sensors, $\lambda_f = 100.0$, $\lambda_s = 0.5$, $f_0 \sim \mathcal{N}(0,1)$ and $f_1 \sim \mathcal{N}(1,1)$.

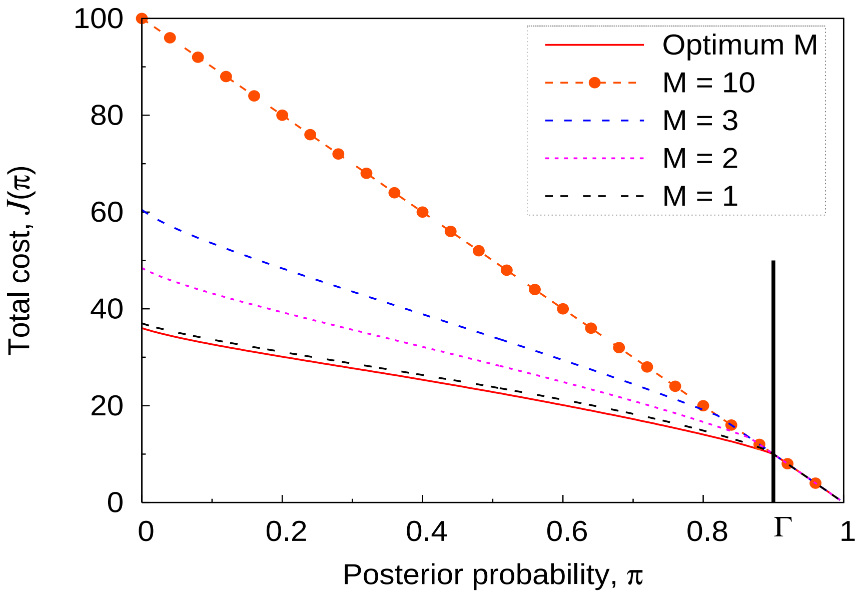


Figure 3.4: Total cost $J(\pi)$ for $n = 10$ sensors, $\lambda_f = 100.0$, $\lambda_s = 0.5$, $f_0 \sim \mathcal{N}(0,1)$ and $f_1 \sim \mathcal{N}(1,1)$. Note that the threshold corresponding to $M = 1$ is 0.895, for $M = 2$ is 0.870, for $M = 3$ is 0.825, and for M^* is $\Gamma = 0.9$.

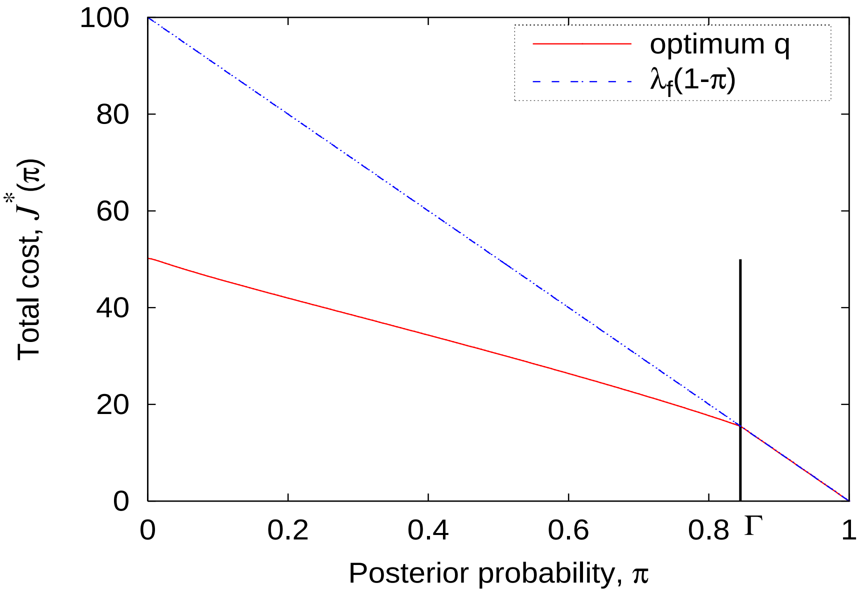


Figure 3.5: Total cost $J^*(\pi)$ for $n = 10$ sensors, $\lambda_f = 100.0$, $\lambda_s = 0.5$, $f_0 \sim \mathcal{N}(0, 1)$ and $f_1 \sim \mathcal{N}(1, 1)$. The dashed line $\lambda_f(1 - \pi)$ is the cost of false alarm.

threshold of 0.9, we see that the detection happens at $\tau_{M^*} = 161$. When $M_{k+1} = 10$ sensors (no sleep scheduling), we find the detection epoch to be $\tau_{10} = 153$. When $M_{k+1} = 3$ sensors (we chose 3 because $M^* \leqslant 3$), the stopping happens at $\tau_3 = 156$. From the above stopping times, it is clear that the detection delay does not vary significantly in the above three cases. By having an optimal sleep –wake scheduling, we observe that until the event occurs only one sensor is in awake state and as soon as the event occurs, the sleep –wake scheduler ramps up the number of sensors to 3, thereby making a quick decision. Thus, the optimal sleep –wake scheduling uses a minimal number of sensors before change and quickly ramps up the number of sensors after change for quick detection. Also, we see from Figure 3.3, that the π_k trajectory corresponding to $M_{k+1}(\pi) = 10$ (and $M_{k+1}(\pi) = 3$) gives more reliable information about the event than the π_k trajectory corresponding to $M_{k+1}(\pi) = M^*$.

We also plot the total cost function $J(\pi)$ for the above cases in Figure 3.4. Though the detection delays do not vary much, the total cost varies significantly. This is because the event happens at time slot $T = 152$. In the case of $M_{k+1} = M^*$, it is clear from Figures 3.2 and 3.3 that only one sensor is used for the first 158 time slots. This reduces the cost by 10 times compared to the case of $M_{k+1} = 10$ (in this sample path) and about 3 times compared to the case of $M_{k+1} = 3$ (in this sample path). We note from Figure 3.4, that it is better to keep 3 sensors active all the time than keeping 10 sensors active all the time. Also, in the case of $M_{k+1} = 1$, after the event occurs, the a posteriori probability takes more time to cross the threshold compared to the optimal sleep –wake (which quickly ramps up from 1 to 3 sensors) and hence, the total cost corresponding to $M_{k+1} = 1$ is slightly worse than that of $M_{k+1} = M^*$.

- **Optimal control of q_{k+1}:** In the case of control on q_k, we consider the same set of parameters as in the case of control on M_k. We computed the optimal policy from the DP defined in Eqn. 3.22 by value iteration method (with 1000 iterations). The optimal policy also gives the optimal probability of choosing a sensor in the awake state, q^*_{k+1}. We plot the total cost $J^*(\pi)$ in Figure 3.5. We also plot the optimum probability of a sensor in the awake state, $q^*(\pi)$ in Figure 3.6. We observe that for $\pi \leq 0.72$, $q^*(\pi)$ is an increasing function of π, and for $\pi > 0.72$, $q^*(\pi)$ decreases with π. This agrees well with the intuition for the optimal control on M_{k+1}.

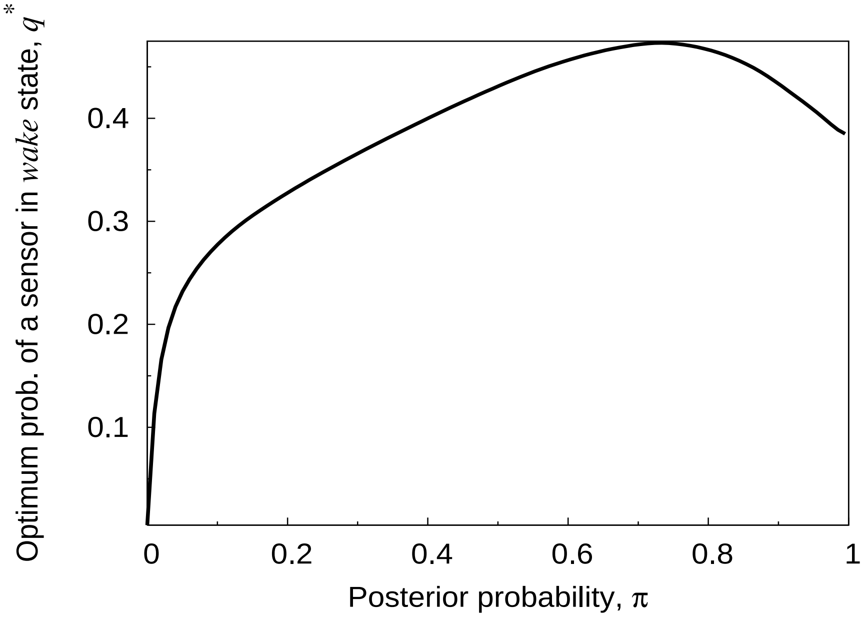


Figure 3.6: Optimum probability of a sensor in the awake state, $q_{k+1}^*(\pi)$ for $n = 10$ sensors, $\lambda_f = 100.0$, $\lambda_s = 0.5$, $f_0 \sim \mathcal{N}(0, 1)$ and $f_1 \sim \mathcal{N}(1, 1)$.

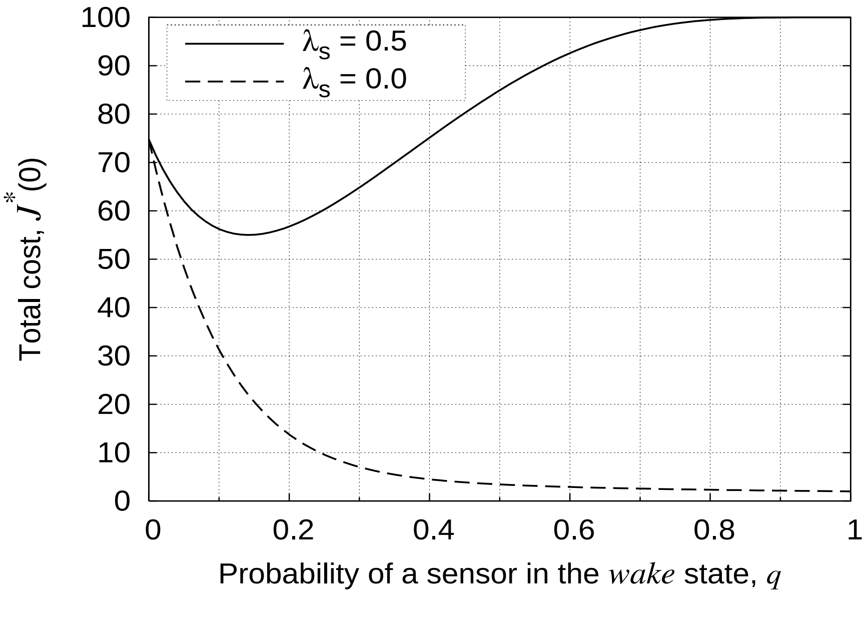


Figure 3.7: Total cost $J^*(0)$ for $n = 10$ sensors, $\lambda_f = 100.0$, $f_0 \sim \mathcal{N}(0, 1)$ and $f_1 \sim \mathcal{N}(1, 1)$.

- **Open loop control on q:**

  We consider the same set of parameters for the case of open loop control on q. We obtain $J^*(0)$ for various values of q and plotted in the Figure 3.7. We obtain the plot for $\lambda_s = 0.5$ and for $\lambda_s = 0.0$. In the special case of $q = 1$, i.e., having $M_{k+1} = 10$ sensors, and with $\lambda_s = 0.5$, we observe that the total cost is 100 which matches with the corresponding cost in Figure 3.4. Also, in the limiting case of $q \to 0$, all the sensor nodes are in the sleep state at all time slots, and the detection happens only based on Bayesian update (i.e., based on the prior distribution of T). Thus at $q = 0$, the total cost is the same (which is 73) for $\lambda_s = 0.5$ and $\lambda_s = 0.0$ which is also evident from Figure 3.7.

  Note that when $\lambda_s > 0$, for low values of q, the detection delay cost dominates over the observation costs in $J^*(0)$ and for high values of q, the observation costs dominate over the detection delay cost. Thus, there is a trade–off between the detection delay cost and the observation costs as q varies. This is captured in the Figure 3.7. Note that the Bayesian cost is optimal at $q = 0.15$.

  When $\lambda_s = 0$ (i.e., the cost of using a sensor per time slot is 0), it is always optimum to use as many sensors as possible at any time slot. Thus, in the open–loop control, between the probabilities of choosing a sensor in the awake state, q_1 and q_2 with $q_1 < q_2$, it is always optimum to use q_2 (since, by choosing q_2, more sensors will be in the awake state, and hence the detection could happen earlier). Since, detection delay is the only cost for $\lambda_s = 0$, (more sensors or) a higher probability of choosing a sensor in the awake state q results in (earlier detection or) lower cost.

From Figures 3.4, 3.5, and 3.7, we note that the total cost $J(\pi)$ is the least for optimal control on M_{k+1}. Also, we note that in the open loop control case, the least total cost $J^*(0) = 55$ is achieved when the attempt probability, q is 0.15 (see Figure 3.7; this corresponds to an average of 1.5 sensors being active). It is to be noted that this cost is larger than that achieved by the optimal closed loop policies ($J^*(0) = 50$ for the closed loop control on q_{k+1} and $J^*(0) = 38$ for the closed loop control on M_{k+1}). From

Figures 3.3 and 3.2, we see that when $M_{k+1}(\pi) = M^*(\pi)$, the switching of the sensors between sleep and awake states happen only in 2 slots out of 161 slots. Otherwise only 1 sensor is on.

3.6 Conclusion

In this chapter, we formulated the problem of jointly optimal sleep –wake scheduling and event detection in a sensor network that minimises the detection delay and the usage of sensing/communication resources. We have set out to solve the problem in Eqn. 3.8. We have derived the optimal control for three approaches using the theory of MDP. We showed the existence of the optimal policy and obtained some structural results.

We prescribe the sleep –wake scheduling policies as follows: When there is a feedback between the fusion centre and the sensors and if the feedback is unicast, it is optimal to use the control on M_{k+1} policy; when the feedback is only broadcast, then it is optimal to use the control on q_{k+1}. If there is no feedback between the fusion centre and the sensors, we prescribe the open loop control on q policy.

3.7 Appendix

Proof of Theorem 3.1

We use the following Lemma to prove Theorem 3.1.

Lemma 3.1 *If $f : [0,1] \to \mathbb{R}$ is concave, then for any $\mathbf{x} \in \mathbb{R}^m$ (for any $m \in \mathbb{Z}_+$), the function $h : [0,1] \to \mathbb{R}$ defined by*

$$h(y) = \mathsf{E}_{\phi_2(\mathbf{x};y)}\left[f\left(\frac{y\phi_1(\mathbf{X})}{y\phi_1(\mathbf{X}) + (1-y)\phi_0(\mathbf{X})}\right)\right]$$

is concave in y, where $\phi_1(\mathbf{x})$ and $\phi_0(\mathbf{x})$ are pdfs on $\mathbf{X}$, and $\phi_2(\mathbf{x};y) = y\phi_1(\mathbf{x}) + (1-y)\phi_0(\mathbf{x})$.

Proof For any given $\mathbf{x}$, define the function $h_1 : [0,1] \to \mathbb{R}$ as

$$h_1(y;\mathbf{x}) \quad := \quad f\left(\frac{y\phi_1(\mathbf{x})}{y\phi_1(\mathbf{x}) + (1-y)\phi_0(\mathbf{x})}\right)\Big[y\phi_1(\mathbf{x}) + (1-y)\phi_0(\mathbf{x})\Big].$$

As $\mathsf{T} := \int \cdots \, d\mathbf{x}$ is a linear operator and $h(y) = \mathsf{T}h_1(y;\mathbf{x})$, it is sufficient to show that $h_1(y;\mathbf{x})$ is concave in y. If $f(y)$ is concave then (see [Rockafellar, 1997])

$$f(y) = \inf_{(a_i,b_i)\in I}\{a_i y + b_i\}$$

where $I = \{(a,b) \in \mathbb{R}^2 : ay + b \geqslant f(y), y \in [0,1]\}$. Hence,

$$\begin{aligned}
&h_1(y;\mathbf{x}) \\
=&f\left(\frac{y\phi_1(\mathbf{x})}{y\phi_1(\mathbf{x}) + (1-y)\phi_0(\mathbf{x})}\right)\Big[y\phi_1(\mathbf{x}) + (1-y)\phi_0(\mathbf{x})\Big] \\
=&\inf_{(a_i,b_i)\in I}\left\{a_i\left(\frac{y\phi_1(\mathbf{x})}{y\phi_1(\mathbf{x}) + (1-y)\phi_0(\mathbf{x})}\right) + b_i\right\} \\
&\cdot\Big[y\phi_1(\mathbf{x}) + (1-y)\phi_0(\mathbf{x})\Big] \\
=&\inf_{(a_i,b_i)\in I}\Big\{a_i y\phi_1(\mathbf{x}) + b_i\Big[y\phi_1(\mathbf{x}) + (1-y)\phi_0(\mathbf{x})\Big]\Big\} \\
=&\inf_{(a_i,b_i)\in I}\Big\{\Big((a_i+b_i)\phi_1(\mathbf{x}) - b_i\phi_0(\mathbf{x})\Big)y + b_i\phi_0(\mathbf{x})\Big\}
\end{aligned}$$

which is an infimum of a collection of affine functions of y. This implies that $h_1(y;\mathbf{x})$ is concave in y (see [Rockafellar, 1997]). ■

The optimal total cost function $J^*(\pi)$ can be computed using a *value iteration* algortithm. Here, we first consider a finite K–horizon problem and then we let $k \to \infty$, to obtain the infinite horizon problem.

Note that the cost–to–go function, $J_K^K(\pi) = \lambda_f \cdot \big(1-\pi\big)$ is concave in π. Hence, by Lemma 3.1, we see that the cost–to–go functions $J_{K-1}^K(\pi)$, $J_{K-2}^K(\pi)$, $\cdots$, $J_0^K(\pi)$ are concave in π. Hence for $0 \leqslant \lambda \leqslant 1$,

$$\begin{aligned} J^*(\pi) &= \lim_{K\to\infty} J_0^K(\pi) \\ J^*(\lambda\pi_1 + (1-\lambda)\pi_2) &= \lim_{K\to\infty} J_0^K\Big(\lambda\pi_1 + (1-\lambda)\pi_2\Big) \\ &\geqslant \lim_{K\to\infty} \lambda J_0^K(\pi_1) + \lim_{K\to\infty} (1-\lambda) J_0^K(\pi_2) \\ &= \lambda J^*(\pi_1) + (1-\lambda) J^*(\pi_2) \end{aligned}$$

It follows that $J^*(\pi)$ is concave in π. ■

Proof of Theorem 3.2

Define the maps $C : [0,1] \to \mathbb{R}_+$ and $H : [0,1] \to \mathbb{R}_+$, as

$$\begin{aligned} C(\pi) &:= \lambda_f \cdot \big(1-\pi\big) \\ H(\pi) &:= \pi + A_{J^*}(\pi) \end{aligned}$$

Note that $C(1) = 0$, $H(1) = 1$, $C(0) = \lambda_f$ and $H(0) = A_{J^*}(0)$. Note that

$$\begin{aligned}
& A_{J^*}(0) \\
= \ & \min_{0 \leqslant m \leqslant n} \left\{ \lambda_s m + \mathsf{E}_{\phi_2(\mathbf{X}^{(m)};p)} \left[J^* \left(\frac{p \cdot \phi_1(\mathbf{X}^{(m)})}{\phi_2(\mathbf{X}^{(m)};p)} \right) \right] \right\} \\
\leqslant \ & \min_{0 \leqslant m \leqslant n} \left\{ \lambda_s m + J^* \left(\mathsf{E}_{\phi_2(\mathbf{X}^{(m)};p)} \left[\frac{p \cdot \phi_1(\mathbf{X}^{(m)})}{\phi_2(\mathbf{X}^{(m)};p)} \right] \right) \right\} \\
= \ & \min_{0 \leqslant m \leqslant n} \left\{ \lambda_s m + J^*(p) \right\} \\
= \ & J^*(p) \\
\leqslant \ & \lambda_f \cdot \big(1 - p\big) \quad \text{(from Eqn. 16)}
\end{aligned}$$

The inequality in the second step is justified using Jensen's inequality and the inequality in the last step follows from the definition of J^*.

Note that $H(1) - C(1) > 0$ and $H(0) - C(0) < 0$. As the function $H(\pi) - C(\pi)$ is concave, by the *intermediate value theorem*, there exists $\Gamma \in [0, 1]$ such that $H(\Gamma) = C(\Gamma)$. This Γ is unique as $H(\pi) = C(\pi)$ for at most two values of π. If in the interval $[0, 1]$, there are two distinct values of π for which $H(\pi) = C(\pi)$, then the signs of $H(0) - C(0)$ and $H(1) - C(1)$ should be the same. Hence, the optimal stopping rule is given by

$$\tau^* = \inf \left\{ k : \Pi_k \geqslant \Gamma \right\}$$

where the threshold Γ is given by $\Gamma + A_{J^*}(\Gamma) = \lambda_f \cdot \big(1 - \Gamma\big)$.

■

Proof of Theorem 3.3

Define

$$
\begin{aligned}
\phi_j(\mathbf{x}^{(m)}) &:= \prod_{i=1}^{m} f_j(x^{(i)}),\ j=0,1.\\
\mathbf{x}^{(l)} &:= (x^{(1)}, x^{(2)}, \cdots, x^{(m)}, x^{(m+1)}, \cdots, x^{(l)})\\
\mathbf{u} &:= (x^{(1)}, x^{(2)}, \cdots, x^{(m)})\\
\mathbf{v} &:= (x^{(m+1)}, x^{(m+2)}, \cdots, x^{(l)})\\
\hat{\pi} &:= \frac{\tilde{\pi}\phi_1(\mathbf{u})}{\tilde{\pi}\phi_1(\mathbf{u}) + (1-\tilde{\pi})\phi_0(\mathbf{u})}
\end{aligned}
$$

Note that

$$
\begin{aligned}
& B_{J^*}^{(l)}(\pi)\\
= & \int_{\mathbb{R}^l} J^* \left(\frac{\tilde{\pi}\cdot\phi_1(\mathbf{x}^{(l)})}{\phi_2(\mathbf{x}^{(l)};\tilde{\pi})}\right) \left[\phi_2(\mathbf{x}^{(l)};\tilde{\pi})\right]\, d\mathbf{x}^{(l)}\\
= & \int_{\mathbb{R}^m}\int_{\mathbb{R}^{l-m}} J^* \left(\frac{\hat{\pi}\phi_1(\mathbf{v})}{\phi_2(\mathbf{v};\hat{\pi})}\right) \phi_2(\mathbf{v};\hat{\pi})\, d\mathbf{v}\phi_2(\mathbf{u};\tilde{\pi})\, d\mathbf{u}\\
\leqslant & \int_{\mathbb{R}^m} J^* \left(\int_{\mathbb{R}^{l-m}} \frac{\hat{\pi}\phi_1(\mathbf{v})}{\phi_2(\mathbf{v};\hat{\pi})}\left[\phi_2(\mathbf{v};\hat{\pi})\right] d\mathbf{v}\right) \phi_2(\mathbf{u};\tilde{\pi})\, d\mathbf{u}\\
= & \int_{\mathbb{R}^m} J^*\left(\hat{\pi}\right)\phi_2(\mathbf{u};\tilde{\pi}) d\mathbf{u}\\
= & B_{J^*}^{(m)}(\pi)
\end{aligned}
$$

As J^* is concave, the inequality in the second line follows from Jensen's inequality. Hence proved. ■

Proof of Theorem 3.4

Eqn. 3.18 and the monotone property of $d(m;.)$ proves the theorem. ■

Proof of Theorem 3.5

Follows from the proof of Theorem 3.1. ■

Proof of Theorem 3.6

Follows from the proof of Theorem 3.2. ■

Proof of Theorem 3.7

Follows from the proof of Theorem 3.1. ■

Proof of Theorem 3.8

Follows from the proof of Theorem 3.2. ■

Chapter 4

Quickest Event Detection on Ad Hoc Wireless Networks

4.1 Introduction

In the previous chapter, we have studied event detection problems when the observations from the sensor nodes reach the fusion centre in the same slot in which each measurement is taken. Such communication is possible by means of parallel channels. Practical multiple access schemes are based on random access where nodes contend for the channel, and back off and reattempt in the case of a collision. Two popular standards are the IEEE 802.11 standard for wireless local area networks and the IEEE 802.15.4 standard for low rate personal area networks; both of these are based on variations of the CSMA/CA mechanism (see [Kumar et al., 2008]). In this chapter, we are concerned with the situation in which the sensors communicate their measurements to the fusion centre over a random access network. The main issue of concern is the following. If the sampling rate at the sensors is high then the fusion centre is supplied with more information, and this could reduce detection delay; however, the higher traffic on the multiple access network would cause increased queueing delays. It is this tradeoff that we study in this chapter.

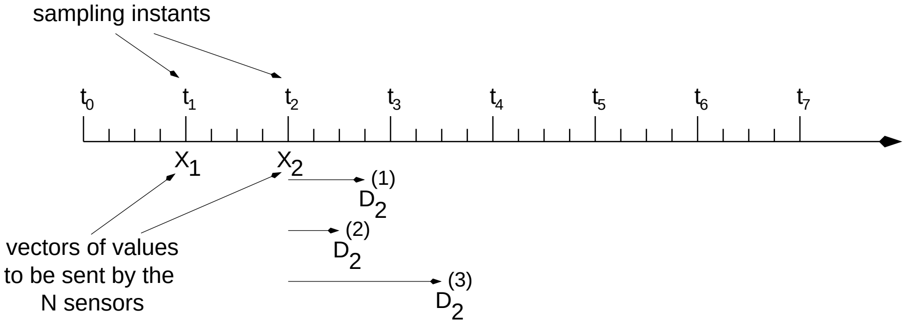


Figure 4.1: The sensors take samples periodically at instants $t_1, t_2, \cdots$, and prepare to send to the fusion centre a vector of values $\mathbf{X}_b = \left[X_b^{(1)}, X_b^{(2)}, \cdots, X_b^{(n)}\right]$ at t_b. Each sample is queued as a packet in the queue of the respective node. Due to multiple access delays, the packets arrive with random delays at the fusion centre; for example, for $\mathbf{X}_2$, the delays $D_2^{(1)}, D_2^{(2)}, D_2^{(3)}$, for the packets from sensors $1, 2$ and 3, are shown.

We consider *a small extent network* in which *the coverage* of a sensor includes the region of interest (ROI) $\mathcal{A}$, i.e., the statistics of the observations are the same at all the sensors, and the event changes the distribution of observations of all the sensors. n sensors are deployed in the ROI and they *synchronously* sample their environment at a particular sampling rate. Synchronized operation across sensors is practically possible in networks such as 802.11 WLANs and Zigbee networks since the access point and the PAN coordinator, respectively, transmit beacons that provide all nodes with a time reference. Based on the measurement samples, the nodes send certain values (e.g., quantized samples) to the fusion centre. Each value is carried by a packet, which is transmitted using a contention–based multiple access mechanism. We are interested in employing optimal decision making procedures at the fusion centre for event detection.

In this setting, due to the multiple access network delays between the sensor nodes and the fusion centre, several possibilities arise. In Figure 4.1 we show that although the sensors take samples synchronously, due to random access delays the various packets sent by the sensors arrive at the fusion centre asynchronously. As shown in the figure, the packets generated due to the samples taken at time t_2 arrive at the fusion centre with a delay of $D_2^{(1)}, D_2^{(2)}, D_2^{(3)}$, etc. It can even happen that a packet corresponding to the samples taken at time t_3 can arrive at the fusion centre before one of the packets generated due to the samples taken at time t_2.

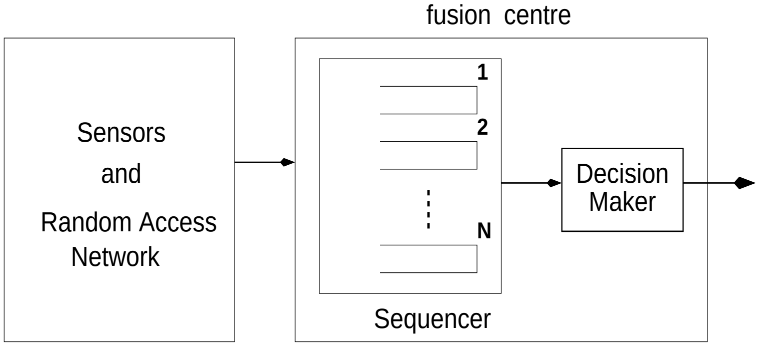


Figure 4.2: A conceptual block diagram of wireless sensor network. The fusion centre has a sequencer buffer which queues out–of–sequence samples and delivers the samples to the decision maker in time–order, as early as possible, batch–wise or sample–wise.

Figure 4.2 depicts a general queueing and decision making architecture in the fusion centre. All samples are queued in per–node queues in a sequencer. The way the sequencer releases the packets gives rise to the following three cases.

1. The sequencer queues the samples until all the samples of a "batch" (a batch is the set of samples generated at a sampling instant) are accumulated; it then releases the entire batch to the decision device. The batches arrive to the decision maker in a time sequence order. The decision maker processes the batches without knowledge of the state of the network (i.e., reception times at the fusion centre, and the numbers of packets in the various queues). We call this, *Network Oblivious Decision Making* (NODM). In factory and building automation scenarios, there is a major impetus to replace wireline networks between sensor nodes and controllers. In such applications, the first step could be to retain the fusion algorithm in the controller, while replacing the wireline network with a wireless ad hoc network. Indeed, we show that this approach is optimal for NODM, provided the sampling rate is appropriately optimized. This problem is studied in [Prasanthi and Kumar, 2006].

2. The sequencer releases samples only in time–sequence order but does not wait for an entire batch to accumulate. The decision maker processes samples as they arrive. We call this, *Network Aware Decision Making* (NADM). In NADM, whenever the decision maker receives a sample, it has to rollback its decision statistic to the

sampling instant, update the decision statistic with the received sample and then update the decision statistic to the current time slot. The decision maker makes a Bayesian update on the decision statistic even if it does not receive a sample in a slot. Thus, NADM requires a modification in the decision making algorithm in the fusion centre. We are interested in studying NADM in this chapter.

3. The sequencer does not queue any samples. The fusion centre acts on the values from the various sampling instants as they arrive, possibly out of order. The formulation of such a problem would be an interesting topic for future research.

4.1.1 Summary of Contributions

We summarise the contributions of this chapter below:

1. We formulate the problem of quickest event detection on ad hoc wireless network.
2. We propose a class of decision strategies called NADM, in which the decision maker makes a decision based on the samples as and when it comes, but in time–sequence order.
3. We obtain an optimal change detection procedure the mean detection delay of which is minimal in the class of NADM policies for which $\mathsf{P}_{\mathsf{FA}} \leqslant \alpha$.
4. We study the tradeoff between the sampling rate, r and the mean detection delay. We also study the detection delay performance as a function of the number of nodes n, for a given *number of observations per unit slot*, i.e., for a fixed nr.

4.1.2 Discussion of the Related Literature

In the existing literature on the topic of optimal sequential event detection in wireless sensor networks, to the best of our knowledge, the only work that considered the network delay is by Prasanthi and Kumar, [Prasanthi and Kumar, 2006], where the authors studied the optimal NODM procedure. There has been no other work that incorporates multiple access delay between the sensing nodes and the fusion centre.

Interestingly, in this chapter we introduce, what can be called a *cross layer* formulation involving *sequential decision theory* and *random access network delays.* In particular, we encounter the *fork–join queueing model* (see, for example, [Baccelli and Makowski, 1990]) that arises in distributed computing literature.

4.1.3 Outline of the Chapter

In Section 4.2, we discuss the event detection problem and setup the model. In Section 4.3, we review the change detection problem over a random access network in a way that naturally includes the network delay. In Section 4.4, we consider the special case of a network with a star topology, i.e., all nodes are one hop away from the fusion centre and provide a model for contention in the random access network. In Section 4.5, we formulate the NADM problem where we process the samples as they arrive at the fusion centre, but in a time causal manner. The out–of–sequence packets are queued in a sequencer buffer and are released to the decision maker as early as possible. We show in the NADM case that the change–detection problem can be modeled as a Partially Observable Markov Decision Process (POMDP). We show that a *sufficient statistic* for the observations include the *network state* (which include the queue lengths of the sequencer buffer, network–delays) and *the posterior probability of change having occurred* given the measurements received and the network states. As usual, the optimal policy can be characterised via a Bellman equation, which can then be used to derive insights into the structure of the policy. We show that the *optimal policy is a threshold on the posterior probability of change and that the threshold, in general, depends on the network state.* Finally, in Section 4.6 we compare, numerically, the mean detection delay performance of NODM and NADM processing. We show the tradeoff between the sampling rate r and the mean detection delay. Also, we show the tradeoff between the number of sensors and the mean detection delay.

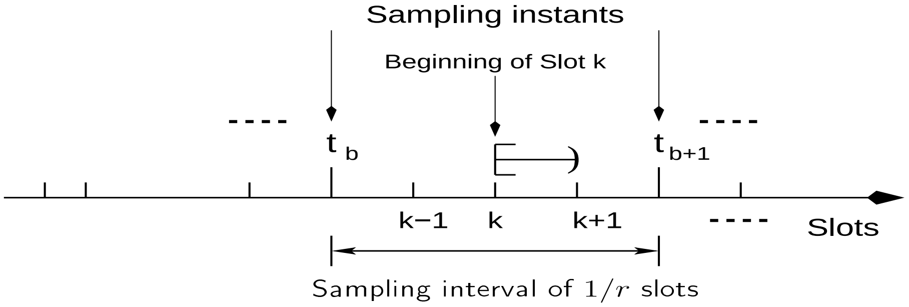


Figure 4.3: Time evolves over slots. The length of a slot is assumed to be unity. Thus, slot k represents the interval $[k, k+1)$ and the beginning of slot k represents the time instant k. Samples are taken periodically every $1/r$ slots, starting from $t_1 = 1/r$.

4.2 Event Detection Problem on Ad Hoc Networks

In this section, we introduce the model for the event detection problem on ad hoc wireless networks. The notation, we follow, is given here.

- Time is slotted and the slots are indexed by $k = 0, 1, 2 \ldots$. We assume that the length of a slot is unity and that slot k refers to the interval $[k, k+1)$. Thus, the beginning of slot k indicates the time instant k (see Figure 4.3), and the notation $k+$ means the time instant just after k and $k-$ means the time instant just before time instant k.

- The state of nature at slot k, $\Theta_k \in \{0, 1\}$. 0 represents the state "before change" and 1 represents the state "after change". It is assumed that the change time T (measured in slots), is geometrically distributed, i.e.,

$$\begin{aligned} \mathsf{P}\{T = 0\} &= \rho \\ \text{and, for } k \geqslant 1, \quad \mathsf{P}\{T = k \mid T > 0\} &= p(1-p)^{(k-1)}. \end{aligned} \tag{4.1}$$

  The value of 0 for T accounts for the possibility that the change took place at or before time instant $k = 0$, i.e., before the observations can be made.

- n sensors are *synchronously* sampling at the rate r samples/slot, i.e., the sensors make an observation every $1/r$ slots and send their observations to the fusion centre. Thus, for example, if $r = 0.1$, then a sample is taken by a sensor every 10^{th} slot. We assume that $1/r$ is an integer. The sampling instants are denoted $t_1, t_2, \ldots$ (see Figure 4.4). Define $t_0 = 0$; note that the first sample is taken at $t_1 = 1/r$. We sometimes refer to this as "coarse sampling" since the samples are taken at some multiples of the basic slot. As a consequence, if a change occurs in a slot, a sample that observes the change will be taken only at the next sampling instant, thus leading to what we call "coarse sampling delay."

- Let S_b, $b \geqslant 1$, be the state of nature at the bth sampling instant and S_0 the state at time 0. Then $S_b \in \{0, 1\}$, with

$$\mathsf{P}\{S_0 = 1\} = \rho = 1 - \mathsf{P}\{S_0 = 0\}$$

S_b evolves as follows. If $S_b = 0$ for $b \geqslant 0$, then

$$S_{b+1} = \begin{cases} 1 & \text{w.p.} & p_r \\ 0 & \text{w.p.} & (1 - p_r) \end{cases}$$

where $p_r = 1 - (1-p)^{1/r}$. Here, if $\Theta_{t_b} = 0$ and $\Theta_{t_{b+1}} = 1$, then the state of nature has changed from 0 to 1 at one of the slots $\frac{b}{r} + 1, \frac{b}{r} + 2, \cdots, \frac{b+1}{r}$. Denote by K the first sampling instant after the change occurs. Further, if $S_b = 1$, then $S_{b+1} = 1$. Thus, if $S_0 = 0$, then there is a change from 0 to 1 at the Kth sampling instant, where K is geometrically distributed. For $b \geqslant 1$,

$$\mathsf{P}\{K = b\} = p_r(1 - p_r)^{b-1}$$

- The vector of outputs from the sensor devices at the bth batch is denoted by

$$\mathbf{X}_b = \left[X_b^{(1)}, X_b^{(2)}, \cdots, X_b^{(n)}\right]$$

where $X_b^{(i)} \in \mathcal{X}$ is the bth output at the ith sensor. Given the state of nature,

$X_b^{(i)}$ are assumed to be (conditionally) independent across sensors and i.i.d. over sampling instants with probability distributions $F_0(x)$ and $F_1(x)$ before and after the change respectively. $\mathbf{X}_1$ corresponds to the first sample taken. In this work, we do not consider the problem of optimal processing of the sensor measurements to yield the sensor outputs, e.g., optimal quantizers (see [Veeravalli, 2001]).

- The vector of network delays of the batch b is denoted by

$$\mathbf{D}_b = \left[D_b^{(1)}, D_b^{(2)}, \cdots, D_b^{(n)}\right]$$

where $D_b^{(i)} \in \{1, 2, 3, \cdots\}$ is the network delay in slots, of the ith component of the bth batch (sampled at $t_b = b/r$). Also, note that $D_b^{(i)} \geqslant 1$, as it requires one time slot for the transmission of a packet to the fusion centre after a successful contention.

Each value to be sent to the fusion centre by a node is inserted into a packet which is queued for transmission. It is then transmitted to the fusion centre by using a contention based multiple access protocol. A node can directly transmit its observation to the fusion centre or route it through other nodes in the system. Each packet takes a time slot to transmit. The MAC protocol and the queues evolve over the same time slots. The fusion centre makes a decision about the change depending on whether *Network Oblivious* (NODM) processing or *Network Aware* (NADM) processing is employed at the fusion centre.

- In the case of NODM processing, the decision sequence (also called as *action sequence*), is $A_u,\ u \geqslant 0$, with $A_u \in \{\mathsf{stop\ and\ declare\ change}(1), \mathsf{take\ another\ sample}(0)\}$, where u is a time instant at which a complete batch of n samples corresponding to a sampling instant is received by the fusion centre.

- In the case of NADM processing, the decision sequence is $A_k,\ k \geqslant 0$, with $A_k \in \{\mathsf{stop\ and\ declare\ change}(1), \mathsf{take\ another\ sample}(0)\}$, i.e., a decision about the change is taken at the beginning of every slot.

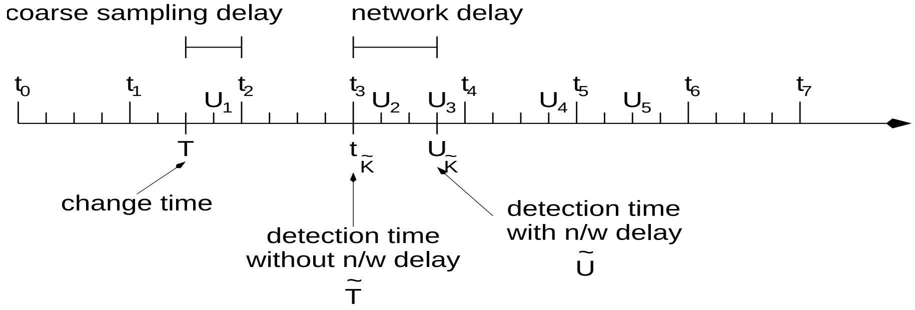


Figure 4.4: Change time and the detection instants with and without network delay are shown. The coarse sampling delay is given by $t_K - T$ where t_K is the first sampling instant after change, and the network delay is given by $U_{\tilde{K}} - t_{\tilde{K}}$.

4.3 Network Oblivious Decision Making (NODM)

In this section, we review the optimal NODM change detection problem (see [Premkumar et al.], [Prasanthi and Kumar, 2006]). From Figure 4.1, we note that although all the components of a batch b are generated at $t_b = b/r$, they reach the fusion center at times $t_b + D_b^{(i)}, i = 1, 2, \cdots, n$. In NODM processing, the samples, which are successfully transmitted, are queued in a sequencing buffer as they arrive (see Figure 4.5) and the sequencer releases a (complete) batch to the decision maker, as soon as all the components of a batch arrive. The decision maker makes a decision about the change at the time instants when a (complete) batch arrives at the fusion center. In the Network Oblivious (NODM) processing, the decision maker is oblivious to the network and processes the batch *as though it has just been generated* (i.e., as if there is no network, hence the name *Network Oblivious Decision Making*). We further define (see Figure 4.4)

- $U_b, (b \geqslant 1)$: the random instant at which the fusion center receives the complete batch $\mathbf{X}_b$
- $\widetilde{K} \in \{0, 1, \dots\}$: the batch index at which the decision takes place, if there was no network delay. $\widetilde{K} = 0$ means that the decision 1 (stop and declare change) is taken before any batch is generated
- $\widetilde{T} = t_{\widetilde{K}}$: the random time (a sampling instant) at which the fusion center stops and declares change, *if there was no network delay*

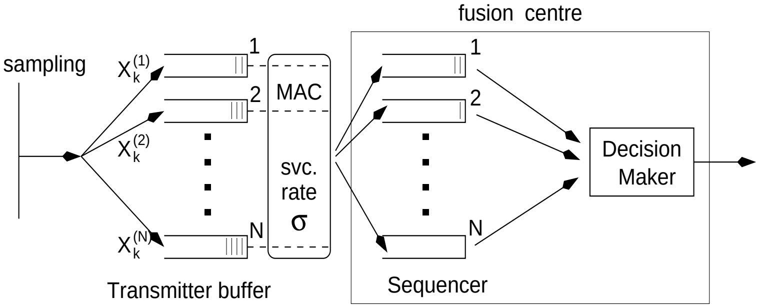


Figure 4.5: A sensor network model of Figure 4.2 with one hop communication between the sensor nodes and the fusion center. The random access network along with the sequencer is a fork–join queueing model.

- $\widetilde{U} = U_{\widetilde{K}}$: the random time slot at which the fusion center stops and declares change, *in the presence of network delay*

- $D_b = U_b - t_b$: Sojourn time of the bth batch, i.e., the time taken for all the samples of the bth batch to reach the fusion center. Note that D_b is given by $\max\{D_b^{(i)} : i = 1, 2, \cdots, n\}$. Thus, the delay of the batch $\widetilde{K}$ at which the detector declares a change is $U_{\widetilde{K}} - t_{\widetilde{K}} = \widetilde{U} - \widetilde{T}$

We now define the following detection metrics.

Definition 4.1 Mean Detection Delay is defined as the expected number of slots between the change point T and the stopping time instant τ, i.e., *Mean Detection Delay* $= \mathsf{E}\left[(\tau - T)^+\right]$.

Definition 4.2 Mean Decision Delay is defined as the expected number of slots between the change point T and the stopping time instant $\widetilde{T}$ in the (presence of *coarse sampling* delay and in the) absence of *network* delay, i.e., *Mean Decision Delay* $=$ $\mathsf{E}\left[\left(\widetilde{T} - T\right)\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\right]$.

With the above model and assumptions, the NODM problem is posed as follows. *Minimize the mean detection delay with a bound on the probability of false alarm*, the decision epochs being the time instants when a complete batch of n components corresponding to a sampling instant is received by the fusion center. In Section 4.5, we pose the problem of making a decision at every slot based on the samples as they arrive at the fusion center. The optimum NODM change detection problem is motivated by the approach in [Veeravalli, 2001]. For a given sampling rate r, the problem is defined as

$$\begin{aligned} \text{minimise} \quad & \mathsf{E}\left[(\widetilde{U} - T)\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\right] \\ \text{subject to} \quad & \mathsf{P}\left\{\widetilde{T} < T\right\} \leqslant \alpha \end{aligned} \tag{4.2}$$

where α is the constraint on the false alarm probability.

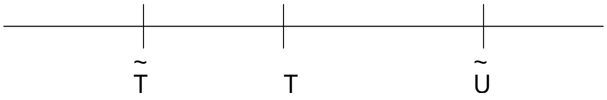


Figure 4.6: Illustration of an event of false alarm with $\widetilde{T} < T$, but $\widetilde{U} > T$

Remark 4.1 *Note that if $\alpha \geqslant 1 - \rho$, then the decision making procedure can be stopped and an alarm can be raised even before the first observation. Thus, one can assume that $\alpha < 1 - \rho$.*

Remark 4.2 *Note that here $\mathsf{P}\left\{\widetilde{T} < T\right\}$ is considered as the probability of false alarm and not $\mathsf{P}\left\{\widetilde{U} < T\right\}$, i.e., a case as shown in Figure 4.6 is considered a false alarm. This can be understood as follows: when the decision unit detects a change at slot $\widetilde{U}$, the measurements that triggered this inference would be carrying the "time stamp" $\widetilde{T}$, and hence, one can infer that the change actually occurred at or before $\widetilde{T}$. Thus if $\widetilde{T} < T$, it is an error.*

The problem defined in Eqn. 4.2 can be written as

$$\min_{\Pi_\alpha} \mathsf{E}\left[(\widetilde{U} - T)\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\right] \tag{4.3}$$

where Π_α is the set of detection policies for which $\mathsf{P}\left\{\widetilde{T} < T\right\} \leqslant \alpha$.

Theorem 4.1 *If the sampling is periodic at rate r and the batch sojourn time process D_b, $b \geqslant 1$, is stationary with mean $d(r)$, then*

$$\min_{\Pi_\alpha} \mathsf{E}\left[(\widetilde{U} - T)\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\right] \quad = \quad (d(r) + l(r))(1-\alpha) - \rho \cdot l(r) + \frac{1}{r} \min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K} - K\right]^+$$

where $l(r)$ is the delay due to (coarse) sampling.

(see [Premkumar et al.], [Prasanthi and Kumar, 2006] for the proof of Theorem 4.1)

Remark 4.3 *For example in Figure 4.4, the delay due to coarse sampling is $t_2 - T$, $\widetilde{K} - K = 3 - 2 = 1$, and the network delay is $U_3 - t_3$. The stationarity assumption on D_b, $b \geqslant 1$, is justifiable in a network in which measurements are continuously made, but the detection process is started only at certain times, as needed.*

Proof: The following is a sketch of the proof (the details are in [Premkumar et al.]):

$$\begin{aligned}\min_{\Pi_\alpha} \mathsf{E}\left[(\widetilde{U} - T)\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\right] \quad &= \quad \min_{\Pi_\alpha} \left\{\mathsf{E}\left[(\widetilde{U} - \widetilde{T})\mathbf{1}_{\{\widetilde{T} \geqslant T\}}\right] + \mathsf{E}\left[\widetilde{T} - T\right]^+\right\} \\ &= \quad \min_{\Pi_\alpha} \left\{\mathsf{E}[D]\left(1 - \mathsf{P}\left\{\widetilde{T} < T\right\}\right) + \ \mathsf{E}\left[\widetilde{T} - T\right]^+\right\}\end{aligned}$$

where the following fact is used; under periodic sampling, the queueing system evolution and the evolution of the statistical decision problem are independent, i.e., $\widetilde{K}$ is independent of $\{D_1, D_2, \ldots\}$ and $\mathsf{E}[D]$ is the mean stationary queueing delay (of a batch). By writing $\mathsf{E}[D] = d(r)$ and using the fact that the problem $\min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{T} - T\right]^+$ is solved by a policy $\pi_\alpha^* \in \Pi_\alpha$ with $\mathsf{P}\left\{\widetilde{T} < T\right\} = \alpha$, the problem becomes

$$d(r)(1-\alpha) + \min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{T} - T\right]^+ \quad = \quad (d(r) + l(r))(1-\alpha) - \rho \cdot l(r) + \frac{1}{r} \min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K} - K\right]^+$$

where $l(r)$ is the delay due to sampling. Notice that $\min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K} - K\right]^+$ is the basic change detection problem at the sampling instants. ■

Remark 4.4 *It is important to note that the independence between $\widetilde{K}$ and $\{D_1, D_2, \ldots\}$ arises from periodic sampling. Actually this is* conditional *independence given the rate*

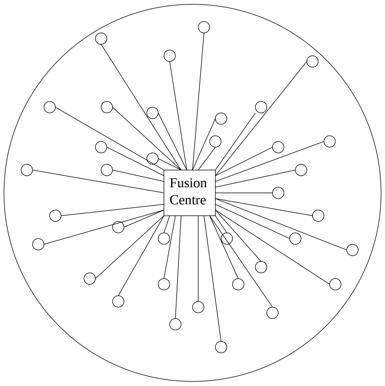


Figure 4.7: A sensor network with a star topology with the fusion center at the hub. The sensor nodes use a random access MAC to send their packets to the fusion centre.

of the periodic sampling process. If, in general, one considers a model in which the sampling is at random times (e.g., the sampling process randomly alternates between periodic sampling at two different rates or if adaptive sampling is used) then one can view it as a time varying sampling rate and the asserted independence will not hold.

Thus, the problem defined in Eqn. 4.2 effectively decouples into the sum of the optimal delay in the original optimal detection problem, i.e., $\frac{1}{r}\min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K}-K\right]^+$ as in [Veeravalli, 2001], a part that captures the network delay, i.e., $d(r)(1-\alpha)$, and a part that captures the sampling delay, i.e., $l(r)(1-\alpha)-\rho l(r)$.

4.4 Network Delay Model

From [Prasanthi and Kumar, 2006], we understand that in NODM processing, the optimal decision device and the queueing system are decoupled. Thus, one can employ an optimal sequential change detection procedure (see [Shiryaev, 1978]) for any random access network (in between the sensor nodes and the fusion centre). Also, NODM is oblivious to the random access network (in between the sensor nodes and the fusion centre) and processes a received batch as though it has just been generated. In the case of NADM (which we describe in Section 4.5), the decision maker processes samples, keeping network–delays into account, thus requiring the knowledge of the network dynamics. In this section, we provide a simple model for the random access network, that facilitates the analysis and optimization of NADM.

In order to understand issues, tradeoffs, and to compare control algorithms, we have considered the following network model. n sensors form a star topology[1] (see Figure 4.7) ad hoc wireless sensor network with the fusion centre as the hub. They synchronously sample their environment at the rate of r samples per slot periodically. At sampling instant $t_b = b/r$, sensor node i generates a packet containing the sample value $X_b^{(i)}$ (or some quantized version of it). This packet is then queued first-in-first-out in the buffer behind the radio link. It is as if each sample is a *fork* operation ([Baccelli and Makowski, 1990]) that puts a packet into each sensor queue (see Figure 4.5).

The sensor nodes contend for access to the radio channel, and transmit packets when they succeed. The service is modeled as follows. As long as there are packets in any of the queues, successes are assumed to occur at the constant rate of σ $(0 < \sigma < 1)$ per slot, with the intervals between the successes being i.i.d., geometrically distributed random variables, with mean $1/\sigma$. If, at the time a success occurs, there are n nodes contending (i.e., n queues are nonempty) then the success is ascribed to any one of the n nodes with equal probability.

The n packets corresponding to a sample arrive at random times at the fusion centre. If the fusion centre needs to accumulate all the n components of each sample then it must wait for that component of every sample that is the last to depart its mote. This is a *join* operation (see Figure 4.5).

It is easily recognized that our service model is the discrete time equivalent of generalized processor sharing (GPS – see, for example, [Kumar et al., 2004]). The queueing model, in the case of Network Oblivious Decision Making (NODM) is called the FJQ-GPS (fork–join queue, see [Baccelli and Makowski, 1990]).

In IEEE 802.11 networks and IEEE 802.15.4 networks, if appropriate parameters are used, then the adaptive backoff mechanism can achieve a throughput that is roughly constant over a wide range of n, the number of contending nodes. This is well known for the CSMA/CA implementation in IEEE 802.11 wireless LANs; see, for example,

[1] Note that *Theorem 1* is more general and does not assume a star topology.

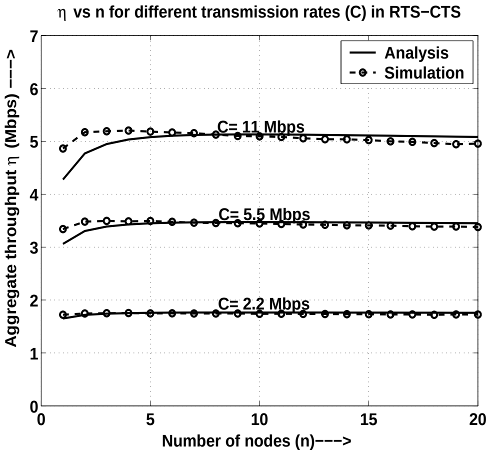


Figure 4.8: The aggregate saturation throughput η of an IEEE 802.11 network plotted against the number of nodes in the network, for various physical layer bit rates: 2.2 Mbps, 5.5 Mbps, and 11 Mbps. The two curves in each plot correspond to an analysis and an NS–2 simulation (reprinted from [Singh et al., 2008]).

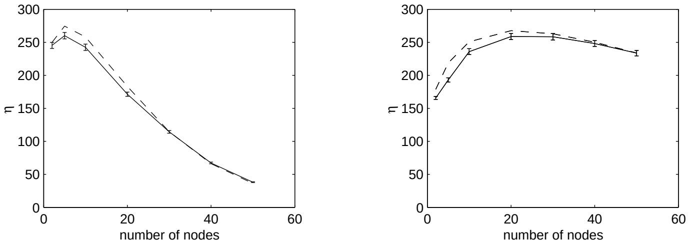


Figure 4.9: The aggregate saturation throughput η of an IEEE 802.15.4 star topology network plotted against the number of nodes in the network. Throughput obtained with default backoff parameters is shown on the left and that obtained with backoff multiplier = 3, is shown on the right. The two curves in each plot correspond to an analysis and an NS–2 simulation (reprinted from [Singh et al., 2008]).

Figure 4.8 [Kumar et al., 2008]. For each physical layer rate, the network service rate remains fairly constant with increasing number of nodes. From Figure 4.9 (taken from [Singh et al., 2008]) it can be seen that with the default backoff parameters, the saturation throughput of a star topology IEEE 802.15.4 network decreases with the number of nodes n, but with the backoff multiplier $= 3$, the throughput remains almost constant from $n = 10$ to $n = 50$ [Singh et al., 2008]; thus, in the latter case our GPS model can be applicable.

We provide the following result in [Prasanthi and Kumar, 2006], for the sake of completeness.

Theorem 4.2 *The stationary delay D of a FJQ–GPS queueing system is a proper random variable with finite mean if and only if $nr < \sigma$.*

Thus, for the FJQ–GPS queueing system to be stable, the sampling rate r is chosen such that $r < \frac{\sigma}{n}$.

4.5 Network Aware Decision Making (NADM)

In Section 4.3, we reviewed the problem of NODM quickest change detection over a random access network, and showed that (when the decision instants are U_k, as shown in Figure 4.4) the optimal decision maker is independent of the random access network, under periodic sampling. Hence, the Shiryaev procedure, which is shown to be delay optimal in the classical change–detection problem (see [Shiryaev, 1978]), can be employed in the decision device independently of the random access network. It is to be noted that the decision maker in the NODM case, waits for a complete batch of n samples to arrive, to make a decision about the change. Thus, the mean detection delay of the NODM has a network–delay component corresponding to a batch of n samples. In this section, we provide a mechanism of fusion at the decision device called the *Network Aware Decision Making* (NADM), in which we do not wait for an entire batch to arrive, but process the samples as soon as they arrive, but in a time–causal manner.

We now describe the processing in NADM. The principle that the fusion centre follows is to process all the samples of a batch as they arrive and finish processing all the samples of the batch before moving on to the next batch. Meanwhile, any samples belonging to later batches are held in the sequencer buffer. Thus, whenever a node (successfully) transmits a sample across the random access network, it is delivered to the decision maker if the decision maker has received all the samples from all the batches generated earlier. Otherwise, the sample is an out–of–sequence sample, and is queued in the sequencer buffer. It follows that, whenever the (successfully) transmitted sample is the last component of the batch that the decision maker is working on, then the *head of line* (HOL) samples, if any, in the queues of the sequencer buffer are also delivered to the decision maker. This is because, these HOL samples belong to the *next* batch that the decision maker should process. The decision maker makes a decision about the change at the beginning of every time slot, irrespective of whether it receives a sample or not. We will see that in NADM, whenever the decision maker receives a sample, it needs to take into account the network–delay of the sample when updating the decision statistic based on the newly arrived sample. The network–delay is a part of the state of the queueing system which is available to the decision maker. Thus, the state of the queueing system also plays a role in decision making.

In Section 4.5.1, we define the state of the queueing system. In Section 4.5.2, we define the dynamical system whose change of state (from 0 to 1) is the subject of interest to us. We define the state of the dynamical system, at time k, as a tuple that contains the *queueing state* and a vector of *states of nature* from time $k - \Delta_k$ to time k. We explain the evolution of the state of the dynamical system in Section 4.5.3. In Section 4.5.4, we explain the model of the sensor observations received by the decision maker. In Section 4.5.5, we formulate the NADM change detection problem and find a sufficient statistic for the observations in Section 4.5.6. In Section 4.5.7, we provide the optimal decision rule for the NADM change detection problem.

4.5.1 Notation and State of the Queueing System

Recall the notation introduced in Section 4.2. Time progresses in slots, indexed by $k = 0, 1, 2 \cdots$; the beginning of slot k is the time instant k. Recall that the nodes take samples at the instants $1/r$, $2/r$, $3/r$, $\cdots$. We define the state of the queueing system here. Note that the queueing system evolves over slots.

- $\lambda_k \in \{1, 2, \cdots, 1/r\}$ denotes the number of time slots to go for the *next* sampling

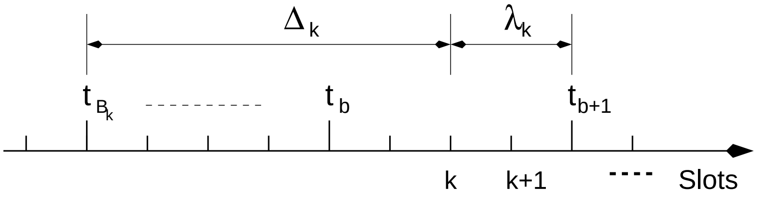
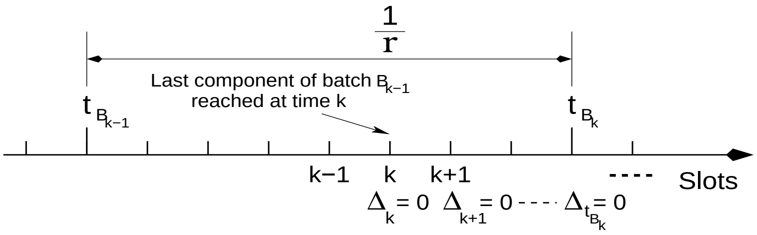


Figure 4.10: At time k, the decision maker is processing samples from (or is expecting the first sample from) batch B_k, which was sampled at t_{B_k}. Also, at time k, λ_k is the number of slots to go for the next sampling instant and Δ_k is the number of slots back at which batch B_k is generated.


$\frac{1}{r}$
Last component of batch B_{k-1}
reached at time k
$t_{B_{k-1}}$
t_{B_k}
k−1
k
k+1

Slots
$\Delta_k = 0$ $\Delta_{k+1} = 0$ ---- $\Delta_{t_{B_k}} = 0$

Figure 4.11: Illustration of a scenario in which $\Delta_k = 0$. If the last component from batch B_{k-1} is received at k, and if there is no sampling instant between $t_{B_{k-1}}$ and k, then $\Delta_k = 0$. Also, note in this case that $\Delta_k = \Delta_{k+1} = \cdots = \Delta_{t_{B_k}} = 0$. In this scenario, at time instants $k, k+1, \cdots, t_{B_k}$, *all the queues at the sensor nodes and at the sequencer are empty*, and at time instant $t_{B_k}+$, all sensor node queues have one packet which is generated at t_{B_k}.

instant, at the beginning of time slot k (see Figure 4.10). Thus,

$$\lambda_k \quad := \quad \frac{1}{r} - \left(k \mod \frac{1}{r}\right). \tag{4.4}$$

Note that at the sampling instants t_b, $\lambda_{t_b} = \frac{1}{r}$. Also, $\lambda_0 = \frac{1}{r}, \lambda_1 = \frac{1}{r} - 1, \cdots, \lambda_{1/r-1} = 1, \lambda_{1/r} = 1/r$, etc.

- $B_k \in \{1, 2, 3, \cdots\}$ denotes the index of the batch that is being processed by the decision maker at the beginning of time slot k, or if the decision maker is in between processing two batches then B_k is the index of the batch that it is expecting to receive a sample from. Note $B_0 = B_1 = \cdots = B_{1/r} = 1$. Also, note that the batch B_k is generated at time instant B_k/r and the earliest any sample from it can be received at the fusion centre is $B_k/r + 1$.

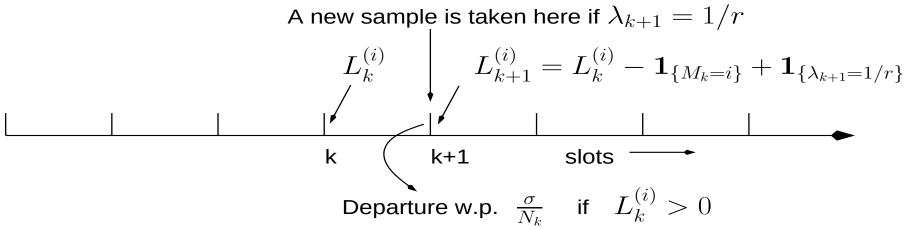

A new sample is taken here if $\lambda_{k+1} = 1/r$
$L_k^{(i)}$
$L_{k+1}^{(i)} = L_k^{(i)} - \mathbf{1}_{\{M_k=i\}} + \mathbf{1}_{\{\lambda_{k+1}=1/r\}}$
k
k+1
slots
Departure w.p. $\frac{\sigma}{N_k}$ if $L_k^{(i)} > 0$

Figure 4.12: The evolution of $L_k^{(i)}$ from time slot k to time slot $k+1$. If during time slot k, node i transmits (successfully) a packet to the fusion centre (i.e., $M_k = i$), then that packet is flushed out of its queue at the end of time slot k. Also, a new sample is generated (every $1/r$ slots) exactly at the beginning of a time slot. Thus, $L_{k+1}^{(i)}$ just after the beginning of time slot $k+1$ (i.e., at $(k+1)+$) is given by $L_{k+1}^{(i)} = L_k^{(i)} - \mathbf{1}_{\{M_k=i\}} + \mathbf{1}_{\{\lambda_{k+1}=1/r\}}$.

- $\Delta_k \in \{0, 1, 2, \cdots\}$ denotes the delay in the number of time slots between the time instants k and B_k/r (see Figure 4.10).

$$\Delta_k \;:=\; \max\left\{k - \frac{B_k}{r}, 0\right\}. \tag{4.5}$$

  Note that the batches of samples taken after B_k/r and up to (including) k are queued either in the sensor node queues or in the sequencer buffer in the fusion centre. If at time k, the fusion centre receives a sample which is the last sample from batch B_{k-1}, then $B_k = B_{k-1} + 1$. If the sampling instant of the B_kth batch is later than k (i.e., $B_k/r > k$), then $\Delta_k = 0$ (and will remain 0 up to time B_k/r at which instant, a new batch is generated, see Figure 4.11). This corresponds to the case, when all the samples generated up to time slot k, have already been processed by the decision maker. In particular, $\Delta_0 = \Delta_1 = \cdots = \Delta_{\frac{1}{r}-1} = 0$.

- $L_k^{(i)} \in \{0, 1, 2, \cdots\}$ denotes the queue length of the ith sensor node just after the beginning of time slot k (i.e., at time instant $k+$). The vector of queue lengths is $\mathbf{L}_k = [L_k^{(1)}, L_k^{(2)}, \cdots, L_k^{(n)}]$. Let $N_k := \sum_{i=1}^n \mathbf{1}_{\{L_k^{(i)}>0\}}$ be the number of non–empty queues at the sensor nodes, just after the beginning of time slot k. i.e., the number of sensor nodes that contend for slot k is N_k. Hence, using the network model we have provided in Section 4.4, the evolution of $L_k^{(i)}$ (see Figure 4.12) is given by the following:

$$\begin{aligned}
L_0^{(i)} &= 0 \\
L_{k+1}^{(i)} &= \begin{cases} L_k^{(i)} + \mathbf{1}_{\{\lambda_{k+1}=1/r\}} & \text{w.p. } 1 \quad \text{if } N_k = 0, \\ L_k^{(i)} + \mathbf{1}_{\{\lambda_{k+1}=1/r\}} & \text{w.p. } (1-\sigma) \quad \text{if } N_k > 0, \\ \max\{L_k^{(i)} - 1, 0\} + \mathbf{1}_{\{\lambda_{k+1}=1/r\}} & \text{w.p. } \frac{\sigma}{N_k} \quad \text{if } N_k > 0. \end{cases}
\end{aligned}$$

Note that when all the samples generated up to time slot k have already been processed by the decision maker and k is not a sampling instant, i.e., $\Delta_k = 0$ and $\lambda_k \neq 1/r$, then $\mathbf{L}_k = \mathbf{0}$ (as there are no outstanding samples in the system). For e.g., $\mathbf{L}_1 = \mathbf{L}_2 = \cdots = \mathbf{L}_{1/r-1} = \mathbf{0}$. Also, note that just after sampling instant t_b, $L_{t_b}^{(i)} \geqslant 1$.

- $M_k \in \{0, 1, 2, \cdots, n\}$ denotes the index of the node that successfully transmits in slot k. $M_k = 0$ means that there is no successful transmission in slot k. Thus, from the network model we have provided in Section 4.4, we note that

$$M_k = \begin{cases} 0 \quad \text{w.p. } 1 & \text{if } N_k = 0 \\ 0 \quad \text{w.p. } (1-\sigma) & \text{if } N_k > 0 \\ j \quad \text{w.p. } \frac{\sigma}{N_k} & \text{if } L_k^{(j)} > 0, \; j = 1, 2, \cdots, n \end{cases}$$

- $W_k^{(i)} \in \{0, 1, 2, \cdots\}$ denotes the queue length of the ith sequencer buffer at time k. The vector of queue lengths is given by $\mathbf{W}_k = [W_k^{(1)}, W_k^{(2)}, \cdots, W_k^{(n)}]$. Note that $\mathbf{W}_k = \mathbf{0}$ if $\Delta_k = 0$, i.e., the sequencer buffer is empty if there are no outstanding samples in the system. In particular, $\mathbf{W}_0 = \mathbf{W}_1 = \cdots = \mathbf{W}_{\frac{1}{r}} = \mathbf{0}$. The evolution of $W_k^{(i)}$ (see Figure 4.13) is given in the Appendix.

- $R_k^{(i)} \in \{0, 1\}$ denotes whether the sample $X_{B_k}^{(i)}$ (i.e., the sample from sensor node i in the batch currently being processed) has been received and processed by the decision maker at time k. $R_k^{(i)} = 0$ means that the sample $X_{B_k}^{(i)}$ has not yet been received by the decision maker and $R_k^{(i)} = 1$ means that the sample $X_{B_k}^{(i)}$ has been

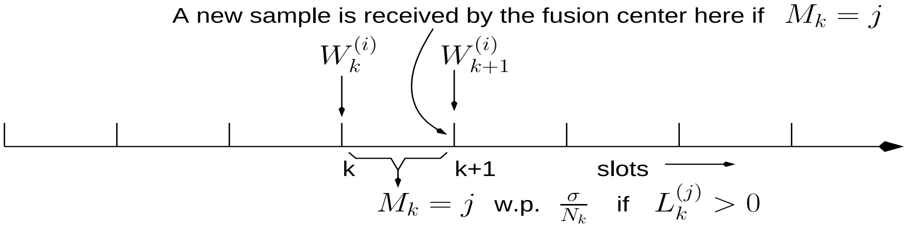

A new sample is received by the fusion center here if $M_k = j$
$W_k^{(i)}$
$W_{k+1}^{(i)}$
k
k+1
slots
$M_k = j$ w.p. $\frac{\sigma}{N_k}$ if $L_k^{(j)} > 0$

Figure 4.13: The evolution of $W_k^{(i)}$ from time slot k to time slot $k+1$. If a sample from node i is transmitted (successfully) during time slot k (i.e., $M_k = i$), then it is received by the fusion centre at the end of time slot k (i.e. at $(k+1)-$). If this sample is from batch B_k, it is passed on to the decision maker. Otherwise, it is queued in the sequencer buffer, in which case $W_{k+1}^{(i)} = W_k^{(i)} + 1$. On the other hand, if a sample from some other node j is transmitted (successfully) during time slot k (i.e., $M_k = j \neq i$), and if this sample is the last component to be received from batch B_k, then the HOL packet, if any, is also delivered to the decision maker. Thus, in this case, $W_{k+1}^{(i)} = \max\{W_k^{(i)} - 1, 0\}$.

received and processed by the decision maker. The vector of $R_k^{(i)}$s is given by $\mathbf{R}_k = [R_k^{(1)}, R_k^{(2)}, \cdots, R_k^{(n)}]$. Note that $W_k^{(i)} = 0$ if $R_k^{(i)} = 0$, i.e., the ith sequencer buffer is empty if the sample expected by the decision maker has not yet been transmitted. Also note that when $\Delta_k = 0$, $\mathbf{R}_k = \mathbf{0}$, as the samples from the current batch B_k have yet to be generated or have just been generated.

We now relate the queue lengths $L_k^{(i)}$ and $W_k^{(i)}$. Note that at the beginning of time slot k, $\left\lfloor \frac{k}{1/r} \right\rfloor$ samples have been generated so far, at sensor node i. Of these, $B_k - 1 + R_k^{(i)}$ samples have been processed by the decision maker and the remaining samples are in the sensor and sequencer queues. Thus, we have

$$\begin{aligned} L_k^{(i)} + W_k^{(i)} &= \left\lfloor \frac{k}{1/r} \right\rfloor - (B_k - 1) - R_k^{(i)} \qquad (4.6) \\ &= \left\lfloor \frac{k - B_k/r + 1/r}{1/r} \right\rfloor - R_k^{(i)} \\ &= \begin{cases} \left\lfloor \frac{\Delta_k}{1/r} \right\rfloor + 1 - R_k^{(i)} & \text{if } k > B_k/r \\ 1 - R_k^{(i)} & \text{if } k = B_k/r \\ -R_k^{(i)} & \text{if } k < B_k/r. \end{cases} \end{aligned}$$

Note that if $k \leqslant B_k/r$, $\Delta_k = 0$. Thus, we write the above Eqn. as,

$$L_k^{(i)} + W_k^{(i)} = \begin{cases} \left\lfloor \frac{\Delta_k}{1/r} \right\rfloor + 1 - R_k^{(i)} & \text{if } \Delta_k > 0 \\ 1 & \text{if } \Delta_k = 0, \lambda_k = 1/r \\ 0 & \text{if } \Delta_k = 0, \lambda_k \neq 1/r. \end{cases} \tag{4.7}$$

Note that in the above Eqn. $\Delta_k = 0, \lambda_k = 1/r$ (or equivalently $k = B_k/r$), corresponds to the case when a sample from batch B_k is just taken and all the samples from all previous batches have been processed. Thus, in this case $L_k^{(i)} = 1$ (as $W_k^{(i)} = 0$). In the case of $\Delta_k = 0, \lambda_k \neq 1/r$ (or equivalently $k < B_k/r$), all the samples from all previous batches have been processed and a new sample from batch B_k is not taken yet. Thus, in this case $L_k^{(i)} = 0$ (and $W_k^{(i)} = 0$). Hence, given $\mathbf{Q}_k = [\lambda_k, B_k, \Delta_k, \mathbf{W}_k, \mathbf{R}_k]$, the queue lengths $L_k^{(i)}$s can be computed. Define the functions $\phi_{L^{(i)}}(\mathbf{Q}_k)$ and $\phi_N(\mathbf{Q}_k)$ as

$$\phi_{L^{(i)}}(\mathbf{Q}_k) := \begin{cases} \left\lfloor \frac{\Delta_k}{1/r} \right\rfloor + 1 - R_k^{(i)} - W_k^{(i)} & \text{if } \Delta_k > 0 \\ 1 & \text{if } \Delta_k = 0, \lambda_k = 1/r \\ 0 & \text{if } \Delta_k = 0, \lambda_k \neq 1/r \end{cases} \tag{4.8}$$

$$\phi_N(\mathbf{Q}_k) := \sum_{i=1}^{n} \mathbf{1}_{\{\phi_{L^{(i)}}(\mathbf{Q}_k) > 0\}}. \tag{4.9}$$

$\phi_{L^{(i)}}(\mathbf{Q}_k)$ computes the queue length $L_k^{(i)}$ and $\phi_N(\mathbf{Q}_k)$ computes N_k. Thus, the state of the queueing system at time k, can be expressed as $\mathbf{Q}_k = [\lambda_k, B_k, \Delta_k, \mathbf{W}_k, \mathbf{R}_k]$. Note that the *decision maker can observe the state* $\mathbf{Q}_k$ *perfectly.* The evolution of the queueing system is explained in the next subsection.

4.5.2 Evolution of the Queueing System

The evolution of the queueing system from time k to time $k + 1$ depends only on the random variable M_k, the success/no–success of contention on the random access channel.

Thus, the state of the queueing system at time $k+1$ is given by

$$\begin{aligned}\mathbf{Q}_{k+1} &= \phi_{\mathbf{Q}}(\mathbf{Q}_k, M_k) && (4.10)\\ &:= [\phi_\lambda(\mathbf{Q}_k, M_k), \phi_B(\mathbf{Q}_k, M_k), \phi_\Delta(\mathbf{Q}_k, M_k), \phi_{\mathbf{W}}(\mathbf{Q}_k, M_k), \phi_{\mathbf{R}}(\mathbf{Q}_k, M_k)] && (4.11)\end{aligned}$$

where we see that M_k appears as "state noise." The functions $\phi_\lambda(\mathbf{Q}_k, M_k)$, $\phi_B(\mathbf{Q}_k, M_k)$, $\phi_\Delta(\mathbf{Q}_k, M_k)$, $\phi_{\mathbf{W}}(\mathbf{Q}_k, M_k)$, and $\phi_{\mathbf{R}}(\mathbf{Q}_k, M_k)$ are described in detail in the Appendix.

In the next subsection, we provide a model of the dynamical system the state of which also has the state of nature that changes from 0 to 1 at a random time T.

4.5.3 System State Evolution Model

Let $\Theta_k \in \{0,1\}, k \geqslant 0$, be the state of nature at the beginning of time slot k. Recall that T is the change point, i.e., for $k < T$, $\Theta_k = 0$ and for $k \geqslant T$, $\Theta_k = 1$, and that the distribution of T is given in Eqn. 4.1. The state Θ_k is observed only through the sensor measurements, but these are delayed. We will formulate the optimal NADM change detection problem as a partially observable Markov decision process (POMDP) with the delayed observations. The approach and the terminology used here is in accordance with the stochastic control framework in [Bertsekas, 2000a]. At time k, a sample, if any, that the decision maker receives is generated at time $B_k/r < k$ (i.e., samples arrive at the decision maker with a network–delay of $\Delta_k = k - \frac{B_k}{r}$ slots). To make an inference about Θ_k from the sensor measurements, we must consider the vector of states of nature that corresponds to the time instants $k - \Delta_k, k - \Delta_k + 1, \cdots, k$. We define the vector of states at time k, $\mathbf{\Theta}_k := [\Theta_{k-\Delta_k}, \Theta_{k-\Delta_k+1}, \cdots, \Theta_k]$. Note that the length of the vector depends on the network–delay Δ_k. When $\Delta_k > 0$, $\mathbf{\Theta}_k = [\Theta_{\frac{B_k}{r}}, \Theta_{\frac{B_k}{r}+1}, \cdots, \Theta_k]$, and when $\Delta_k = 0$, $\mathbf{\Theta}_k$ is just $[\Theta_k]$.

Consider the discrete–time system, which at the beginning of time slot k is described by the state

$$\Gamma_k \;=\; [\mathbf{Q}_k, \mathbf{\Theta}_k],$$

where we recall that

$$\begin{aligned}\mathbf{Q}_k &= \Big[\lambda_k, B_k, \Delta_k, \mathbf{W}_k, \mathbf{R}_k\Big],\\ \mathbf{\Theta}_k &= [\Theta_{k-\Delta_k}, \Theta_{k-\Delta_k+1}, \cdots, \Theta_k].\end{aligned}$$

Note that $\Gamma_0 = \left[\left[\frac{1}{r}, 1, 0, \mathbf{0}\right], \Theta_0\right]$. At each time slot k, we have the following set of controls $\{0, 1\}$ where 0 represents "take another sample", and 1 represents "stop and declare change". Thus, at time slot k, when the control chosen is 1, the state Γ_{k+1} is given by a terminal absorbing state t; when the control chosen is 0, the state evolution is given by $\Gamma_{k+1} = [\mathbf{Q}_{k+1}, \mathbf{\Theta}_{k+1}]$, where

$$\begin{aligned}\mathbf{Q}_{k+1} &= \phi_{\mathbf{Q}}(\mathbf{Q}_k, M_k),\\ \mathbf{\Theta}_{k+1} &= \begin{cases}\left[\Theta_k + \mathbf{1}_{\{T=k+1\}}\right], & \text{if } \Delta_{k+1} = 0\\ \left[\Theta_{k-\Delta_k}, \Theta_{k-\Delta_k+1}, \cdots, \Theta_k, \Theta_k + \mathbf{1}_{\{T=k+1\}}\right], & \text{if } \Delta_{k+1} = \Delta_k + 1\\ \left[\Theta_{k-\Delta_k+\frac{1}{r}}, \Theta_{k-\Delta_k+\frac{1}{r}+1}, \cdots, \Theta_k, \Theta_k + \mathbf{1}_{\{T=k+1\}}\right], & \text{if } \Delta_{k+1} = \Delta_k + 1 - \frac{1}{r}.\end{cases}\\ &=: \phi_{\mathbf{\Theta}}\left(\mathbf{\Theta}_k, \mathbf{Q}_k, M_k, \mathbf{1}_{\{T=k+1\}}\right) \qquad (4.12)\end{aligned}$$

where it is easy to observe that $\Theta_k + \mathbf{1}_{\{T=k+1\}} = \Theta_{k+1}$. When $\Delta_{k+1} = \Delta_k + 1$, the batch B_k is still in service, and hence, in addition to the current state $\Theta_{k+1} = \Theta_k + \mathbf{1}_{\{T=k+1\}}$, we need to keep the states $\Theta_{k-\Delta_k}, \Theta_{k-\Delta_k+1}, \cdots, \Theta_k$. Also, when $\Delta_{k+1} = \Delta_k + 1 - \frac{1}{r}$, then the batch index is incremented, and hence, the vector of states that determines the distribution of the observations sampled at or after B_{k+1}/r and before $k+1$ is given by $\left[\Theta_{k-\Delta_k+\frac{1}{r}}, \Theta_{k-\Delta_k+\frac{1}{r}+1}, \cdots, \Theta_k,\ \Theta_k + \mathbf{1}_{\{T=k+1\}}\right]$.

Define $O_k := \mathbf{1}_{\{T=k+1\}}$, and define $\mathbf{N}_k := [M_k, O_k]$ be the state–noise during time slot k. The distribution of state–noise $\mathbf{N}_k$ given the state of the discrete–time system

Γ_k is given by $\mathsf{P}\left\{M_k = m, O_k = o \,\middle|\, \Gamma_k = [\mathbf{q}, \boldsymbol{\theta}]\right\}$ and is the product of the distribution functions, $\mathsf{P}\left\{M_k = m \,\middle|\, \Gamma_k = [\mathbf{q}, \boldsymbol{\theta}]\right\}$ and $\mathsf{P}\left\{O_k = o \,\middle|\, \Gamma_k = [\mathbf{q}, \boldsymbol{\theta}]\right\}$. These distribution functions are provided in the Appendix.

The problem is to detect the change in the state Θ_k as early as possible by sequentially observing the samples at the decision maker. At each time k, in the dynamical system, the queueing state $\mathbf{Q}_k$ can be observed, but the vector of the states of nature $\mathbf{\Theta}_k$ can not be observed directly and can be observed only through the sensor measurements $\mathbf{Y}_k$ that the decision maker receives.

4.5.4 Model of Sensor Observation received by Decision Maker

Let $\mathbf{Y}_{k+1} \in \{\emptyset\} \cup \mathbb{R}^1 \cup \mathbb{R}^2 \cup \cdots \cup \mathbb{R}^n$ denote the vector of samples received, if any, by the decision maker at the beginning of slot $k+1$ (i.e., the decision maker can receive nothing or a vector of $N+1$ samples where N ranges from 0 to $n-1$). At the beginning of time slot $k+1$, the following possibilities arise:

- **No successful transmission:** This corresponds to the case i) when all the queues are empty at the sensor nodes ($N_k = 0$), or ii) when some queues are non–empty at the sensor nodes ($N_k > 0$), but the contention fails. In either case, $M_k = 0$ and the decision maker does not receive any sample, i.e., $\mathbf{Y}_{k+1} = \emptyset$.

- **Successful transmission of node j's sample from a later batch:** This corresponds to the case, when the decision maker has already received the jth component of the current batch B_k (i.e., $R_k^{(j)} = 1$) and that it has not received some sample, say $i \neq j$, from the batch B_k (i.e., $R_k^{(i)} = 0$, for some i). The received sample (is an out–of–sequence sample and) is queued in the sequencer buffer ($W_{k+1}^{(j)} = W_k^{(j)} + 1$). Thus, in this case, $M_k = j$ and the decision maker does not receive any sample, i.e., $\mathbf{Y}_{k+1} = \emptyset$.

- **Successful transmission of node j's current sample which is not the last sample from the batch B_k to be received by the decision maker:** This corresponds to the case when the decision maker has not received the jth component of the batch B_k before time slot k ($R_k^{(j)} = 0$), and the fusion centre is yet to receive some other samples of batch B_k (i.e., $\sum_{i=1}^n R_k^{(i)} < n-1$). Thus, in this case, $M_k = j$ and the decision maker receives the sample $\mathbf{Y}_{k+1} = X_{B_k}^{(j)}$.

- **Successful transmission of node j's current sample which is the last sample from the batch B_k to be received by the decision maker:** This corresponds to the case when the decision maker has not received the jth component of the batch B_k before time slot k ($R_k^{(j)} = 0$). Also, this sample is the last component of batch B_k, that is received by the fusion centre (i.e., $\sum_{i=1}^n R_k^{(i)} = n-1$). In this case (along with the received sample), the queues of the sequencer buffer deliver the *head of line* (HOL) samples (which correspond to the batch index $B_k + 1$), if any, to the decision maker, and the queues are decremented by one ($W_{k+1}^{(i)} = \max\{W_k^{(i)} - 1, 0\}$). Thus, in this case, $M_k = j$ and the decision maker receives the vector of samples $\mathbf{Y}_{k+1} = \left[X_{B_k}^{(j)}, X_{B_k+1}^{(i'_1)}, X_{B_k+1}^{(i'_2)}, \cdots, X_{B_k+1}^{(i'_N)}\right]$ where $W_k^{(i)} > 0$ for $i \in \{i'_1, i'_2, \cdots i'_N\}$, and $W_k^{(i)} = 0$ for $i \notin \{i'_1, i'_2, \cdots i'_N\}$.

4.5.5 The NADM Change Detection Problem

We now formulate the NADM change–detection problem in which the observations from the sensor nodes are sent over a random access network to the fusion center and the fusion center processes the samples in the NADM mode.

In Section 4.5.3, we defined the state $\Gamma_k = [\mathbf{Q}_k, \mathbf{\Theta}_k]$ on which we formulate the NADM change detection problem as a POMDP. Recall that at the beginning of slot k, the decision maker receives a vector of sensor measurements $\mathbf{Y}_k$ and observes the state $\mathbf{Q}_k$ of the queueing system. Thus, at time k, $\mathbf{Z}_k = [\mathbf{Q}_k, \mathbf{Y}_k]$ is the observation of the decision maker about the state of the dynamical system Γ_k.

Let $A_k \in \{0, 1\}$ be the control (or action) chosen by the decision maker after having observed $\mathbf{Z}_k$ at k. Recall that 0 represents "take another sample" and 1 represents the action "stop and declare change". Let $\mathbf{I}_k = \left[\mathbf{Z}_{[0:k]}, A_{[0:k-1]}\right]$ be the *information* vector that is available to the decision maker, at the beginning of time slot k. Let τ be a stopping time with respect to the sequence of random variables $\mathbf{I}_1, \mathbf{I}_2, \cdots$. Note that $A_k = 0$ for $k < \tau$ and $A_k = 1$ for $k \geqslant \tau$. We are interested in obtaining a stopping time τ that minimises the mean detection delay subject to a constraint on the probability of false alarm.

$$\begin{aligned} \text{minimise} \quad & \mathsf{E}\left[(\tau - T)^+\right] \\ \text{subject to} \quad & \mathsf{P}\left\{\tau < T\right\} \leqslant \alpha \end{aligned} \tag{4.13}$$

Note that in the case of NADM, at any time k, a decision about the change is made based on the information $\mathbf{I}_k$ (which includes the batch index we are processing and the delays). Thus, in the case of NADM, false alarm is defined as the event $\{\tau < T\}$ and, hence, $\tau \geqslant T$ is not classified as a false alarm even if it is due to pre–change measurements only. However, in the case of NODM, this is classified as a false alarm as the decision about the change is based on the batches received until time k.

Let c be the cost per unit delay in detection. We are interested in obtaining a stopping time τ^* that minimises the expected cost (Bayesian risk) given by

$$\begin{aligned} C(c, \tau^*) &= \min_{\tau} \mathsf{E}\left[\mathbf{1}_{\{\Theta_\tau = 0\}} + c \cdot (\tau - T)^+\right] \\ &= \min_{\tau} \mathsf{E}\left[\mathbf{1}_{\{\Theta_\tau = 0\}} + c \cdot \sum_{k=0}^{\tau-1} \mathbf{1}_{\{\Theta_k = 1\}}\right] \\ &= \min_{\tau} \mathsf{E}\left[g_\tau(\Gamma_\tau, A_\tau) + \sum_{k=0}^{\tau-1} g_k(\Gamma_k, A_k)\right] \\ &= \min_{\mu} \sum_{k=0}^{\infty} \mathsf{E}[g_k(\Gamma_k, A_k)] \end{aligned} \tag{4.14}$$

where, as defined earlier, $\Gamma_k = [\mathbf{Q}_k, \mathbf{\Theta}_k]$. Let $\boldsymbol{\theta} = [\theta_\delta, \theta_{\delta-1}, \cdots, \theta_1, \theta_0]$. We define the cost function $g_k(\cdot,\cdot)$ for $k \leqslant \tau$ as

$$g_k([\mathbf{q},\boldsymbol{\theta}],a) \;=\; \begin{cases} 0, & \text{if } \theta_0 = 0, a = 0 \\ 1, & \text{if } \theta_0 = 0, a = 1 \\ c, & \text{if } \theta_0 = 1, a = 0 \\ 0, & \text{if } \theta_0 = 1, a = 1 \end{cases} \tag{4.15}$$

and for $k > \tau$, $g_k(\cdot,\cdot) := 0$. Recall that $A_k = 0$ for $k < \tau$ and $A_k = 1$ for $k \geqslant \tau$. Note that A_k, the control at time slot k, depends only on $\mathbf{I}_k$. Thus, every stopping time τ, corresponds to a policy $\mu = (\mu_0, \mu_1, \cdots)$ such that $A_k = \mu_k(\mathbf{I}_k)$, with $A_k = 0$ for $k < \tau$ and $A_k = 1$ for $k \geqslant \tau$. Since $\boldsymbol{\Theta}_k$ is observed only through $\mathbf{I}_k$, we look at a sufficient statistic for $\mathbf{I}_k$ in the next subsection.

4.5.6 Sufficient Statistic

In Section 4.5.2, we have illustrated the evolution of the queueing system $\mathbf{Q}_k$ and we have shown in different scenarios, the vector $\mathbf{Y}_k$ received by the decision maker. Recall from Section 4.5.2 that

$$\mathbf{Y}_{k+1} = \begin{cases} \emptyset, & \text{if } M_k = 0, \\ \emptyset, & \text{if } M_k = j > 0, R_k^{(j)} = 1, \\ Y_{k+1,0}, & \text{if } M_k = j > 0, R_k^{(j)} = 0, \sum_{i=1}^{n} R_k^{(i)} < n-1 \\ \left[Y_{k+1,0}, Y_{k+1,1}, \cdots, Y_{k+1,N}\right], & \text{if } M_k = j > 0, R_k^{(j)} = 0, \sum_{i=1}^{n} R_k^{(i)} = n-1, \\ & \sum_{i=1}^{n} \mathbf{1}_{\{W_k^{(i)} > 0\}} = N. \end{cases}$$

Note that $Y_{k+1,0}$ corresponds to $X_{B_k}^{(M_k)}$. The last part of the above equation corresponds to the last pending sample of the batch B_k arriving at the decision maker at time $k+1$, with some samples from batch $B_k + 1$ $(= B_{k+1})$ also being released by the sequencer. In this case, the state of nature at the sampling instant of the batch $B_{k+1} = B_k + 1$ is $\Theta_{k-\Delta_k+1/r}$. Note that $\Theta_{k-\Delta_k+1/r}$ is a component of the vector $\boldsymbol{\Theta}_k$ as $k - \Delta_k + 1/r = (B_k + 1)/r < k$.

Thus, the distribution of $Y_{k+1,0}, Y_{k+1,1}, \cdots, Y_{k+1,N}$ is given by

$$f_{Y_{k+1,0}}(\cdot) = \begin{cases} f_0(\cdot), & \text{if } \Theta_{k-\Delta_k} = 0 \\ f_1(\cdot), & \text{if } \Theta_{k-\Delta_k} = 1 \text{ and} \end{cases}$$

$$f_{Y_{k+1,i}}(\cdot) = \begin{cases} f_0(\cdot), & \text{if } \Theta_{k-\Delta_k+1/r} = 0 \\ f_1(\cdot), & \text{if } \Theta_{k-\Delta_k+1/r} = 1 \end{cases}, \ i = 1, 2, \cdots, N.$$

Thus, at time $k+1$, the current observation $\mathbf{Y}_{k+1}$ depends only on the previous state Γ_k, previous action A_k, and the previous noise of the system $\mathbf{N}_k$. Thus, a sufficient statistic is $\left[\mathsf{P}\left\{\Gamma_k = [\mathbf{q}, \boldsymbol{\theta}] \,\middle|\, \mathbf{I}_k\right\}\right]_{[\mathbf{q},\boldsymbol{\theta}]\in\mathcal{S}}$ (see page 244, [Bertsekas, 2000a]) where $\mathcal{S}$ is the set of all states of the dynamical system defined in Sec. 4.5.3. Let $\mathbf{q} = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}]$. Note that

$$\begin{aligned}
& \mathsf{P}\left\{\Gamma_k = [\mathbf{q}, \boldsymbol{\theta}] \,\middle|\, \mathbf{I}_k\right\} \\
= \ & \mathsf{P}\left\{\Gamma_k = [\mathbf{q}, \boldsymbol{\theta}] \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k, \mathbf{Y}_k\right\} \\
= \ & \mathbf{1}_{\{\mathbf{Q}_k=\mathbf{q}\}} \cdot \mathsf{P}\left\{\boldsymbol{\Theta}_k = \boldsymbol{\theta} \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\} \\
= \ & \mathbf{1}_{\{\mathbf{Q}_k=\mathbf{q}\}} \\
& \cdot \mathsf{P}\left\{[\Theta_{k-\delta}, \Theta_{k-\delta+1}, \cdots, \Theta_{k-1}, \Theta_k] = [\theta_\delta, \theta_{\delta-1}, \cdots, \theta_1, \theta_0] \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\} \\
= \ & \mathbf{1}_{\{\mathbf{Q}_k=\mathbf{q}\}} \cdot \mathsf{P}\left\{\Theta_{k-\delta} = \theta_\delta \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\} \\
& \cdot \prod_{j=1}^{\delta} \mathsf{P}\left\{\Theta_{k-\delta+j} = \theta_{\delta-j} \,\middle|\, \Theta_{k-\delta+j'} = \theta_{\delta-j'}, j' = 0, 1, \cdots, j-1, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\}
\end{aligned} \tag{4.16}$$

Observe that

$$\begin{aligned}
& \mathsf{P}\left\{\Theta_{k-\delta+j} = \theta_{\delta-j} \,\middle|\, \Theta_{[k-\delta:k-\delta+j-2]}, \Theta_{k-\delta+j-1} = 0, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\} \\
= & \begin{cases} 1-p, & \text{if } \theta_{\delta-j} = 0 \\ p, & \text{if } \theta_{\delta-j} = 1 \end{cases}
\end{aligned}$$

and

$$\begin{aligned}
&\mathsf{P}\left\{\Theta_{k-\delta+j} = \theta_{\delta-j} \,\middle|\, \Theta_{[k-\delta:k-\delta+j-2]}, \Theta_{k-\delta+j-1} = 1, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\}\\
= &\begin{cases} 0, & \text{if } \theta_{\delta-j} = 0\\ 1, & \text{if } \theta_{\delta-j} = 1. \end{cases}
\end{aligned}$$

This is because given $\Theta_{k-\delta}$, the events $\{\Theta_{k-\delta+j} = \theta_{\delta-j}\}$, $\{\mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\}$ are conditionally independent. Thus, Eqn. 4.16 can be written as

$$\begin{aligned}
&\mathsf{P}\left\{\Gamma_k = [\mathbf{q}, \boldsymbol{\theta}] \,\middle|\, \mathbf{I}_k\right\}\\
= &\begin{cases}
\mathbf{1}_{\{\mathbf{Q}_k = \mathbf{q}\}} \cdot \mathsf{P}\left\{\Theta_{k-\delta} = 1 \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\}, & \text{if } \boldsymbol{\theta} = \mathbf{1}\\
\mathbf{1}_{\{\mathbf{Q}_k = \mathbf{q}\}} \cdot \mathsf{P}\left\{\Theta_{k-\delta} = 0 \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\} \cdot (1-p)^{\delta-j-1}p, & \text{if } \boldsymbol{\theta} = [0, \cdots, 0, \underbrace{1}_{\theta_j}, \cdots, 1]\\
\mathbf{1}_{\{\mathbf{Q}_k = \mathbf{q}\}} \cdot \mathsf{P}\left\{\Theta_{k-\delta} = 0 \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\} \cdot (1-p)^{\delta}, & \text{if } \boldsymbol{\theta} = \mathbf{0}
\end{cases}
\end{aligned} \tag{4.17}$$

Define $\widetilde{\Theta}_k := \Theta_{k-\Delta_k}$, and define

$$\begin{aligned}
\Psi_k &:= \mathsf{P}\left\{\widetilde{\Theta}_k = 1 \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}], \mathbf{Y}_k\right\}\\
&= \mathsf{P}\left\{\Theta_{k-\delta} = 1 \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}], \mathbf{Y}_k\right\}\\
\Pi_k &:= \mathsf{P}\left\{\Theta_k = 1 \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}], \mathbf{Y}_k\right\}\\
&= \mathsf{P}\left\{T \leqslant k \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}], \mathbf{Y}_k\right\}.
\end{aligned} \tag{4.18}$$

Thus, Eqn. 4.17 can be written as

$$\begin{aligned}
&\mathsf{P}\left\{\Gamma_k = [[\lambda, b, \delta, \mathbf{w}, \mathbf{r}], \boldsymbol{\theta}] \,\middle|\, \mathbf{I}_k\right\}\\
= &\begin{cases}
\mathbf{1}_{\{\mathbf{Q}_k = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}]\}} \cdot \Psi_k, & \text{if } \boldsymbol{\theta} = \mathbf{1}\\
\mathbf{1}_{\{\mathbf{Q}_k = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}]\}} \cdot (1 - \Psi_k) \cdot (1-p)^{\delta-j-1}p, & \text{if } \boldsymbol{\theta} = [0, \cdots, 0, \underbrace{1}_{\theta_j}, \cdots, 1]\\
\mathbf{1}_{\{\mathbf{Q}_k = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}]\}} \cdot (1 - \Psi_k) \cdot (1-p)^{\delta}, & \text{if } \boldsymbol{\theta} = \mathbf{0}
\end{cases}
\end{aligned} \tag{4.19}$$

We now find a relation between Π_k and Ψ_k in the following Lemma.

Lemma 4.1 *The relation between the conditional probability distributions* Π_k *and* Ψ_k *is given by*

$$\Pi_k \;=\; \Psi_k + (1-\Psi_k)\left(1-(1-p)^{\delta}\right) \tag{4.20}$$

From Eqn. 4.19 and Lemma 4.1, it is clear that a sufficient statistic for $\mathbf{I}_k$ is $\nu_k = [\mathbf{Q}_k, \Pi_k]$. Also, we show in the Appendix that ν_k can be computed recursively, i.e., when $A_k = 0$, $\nu_{k+1} = [\mathbf{Q}_{k+1}, \Pi_{k+1}] = [\mathbf{Q}_{k+1}, \phi_{\Pi}(\nu_k, \mathbf{Z}_{k+1})]$, and when $A_k = 1$, $\nu_{k+1} = \mathsf{t}$, a terminal state. Thus, ν_k can be thought of as entering into a terminating (absorbing) state t at τ (i.e., $\nu_k = [\mathbf{Q}_k, \Pi_k]$ for $k < \tau$ and $\nu_k = \mathsf{t}$ for $k \geqslant \tau$). Since ν_k is sufficient, for every policy μ_k there corresponds a policy $\widetilde{\mu}_k$ such that $\mu_k(\mathbf{I}_k) = \widetilde{\mu}_k(\nu_k)$ (see page 244, [Bertsekas, 2000a]).

4.5.7 Optimal Stopping Time τ

Let $\mathcal{Q}$ be the set of all possible states of the queueing system, $\mathbf{Q}_k$. Thus the state space of the sufficient statistic is $\mathcal{C} = (\mathcal{Q} \times [0,\ 1]) \cup \{\mathsf{t}\}$. Recall that the action space is $\mathcal{A} = \{0, 1\}$. Define the one–stage cost function $\widetilde{g} : \mathcal{C} \times \mathcal{A} \to \mathbb{R}_+$ as follows. Let $\nu \in \mathcal{C}$ be a state of the system and let $a \in \mathcal{A}$ be a control. Then,

$$\widetilde{g}(\nu, a) \;=\; \begin{cases} \lambda_f(1-\pi), & \text{if } \nu = [\mathbf{q}, \pi], a = 1 \\ \pi, & \text{if } \nu = [\mathbf{q}, \pi], a = 0 \\ 0, & \text{if } \nu = \mathsf{t}. \end{cases}$$

Note from Eqn. 4.15 that

$$\begin{aligned} \mathsf{E}[g(\Theta_k, A_k)] &= \mathsf{E}[g_k(\Theta_k, \mu_k(\mathbf{I}_k))] \\ &= \mathsf{E}\left[\mathsf{E}\left[g_k(\Theta_k, \mu_k(\mathbf{I}_k)) \,\middle|\, \mathbf{I}_k\right]\right] \\ &= \mathsf{E}[\widetilde{g}(\nu_k, \widetilde{\mu}_k(\nu_k))]\,. \end{aligned}$$

Since, $\{\nu_k\}$ is a controlled Markov process, and the one–stage cost function $\widetilde{g}(\cdot,\cdot)$, the transition probability kernel for $A_k = 1$ and for $A_k = 0$ (i.e., $\mathsf{P}\left\{\mathbf{Z}_{k+1} \,\middle|\, \nu_k\right\}$), do not depend on time k, and the optimization problem defined in Eqn. 4.14 is over infinite horizon, it is sufficient to look for an optimal policy in the space of stationary Markov policies (see page 83, [Bertsekas, 2000b]). Thus, the optimization problem defined in Eqn. 4.14 can be written as

$$\begin{aligned} C(c,\tau^*) &= \min_{\widetilde{\mu}} \sum_{k=0}^{\infty} \mathsf{E}\big[\widetilde{g}\big(\nu_k, \widetilde{\mu}_k(\nu_k)\big)\big] \\ &= \sum_{k=0}^{\infty} \mathsf{E}\big[\widetilde{g}\big(\nu_k, \widetilde{\mu}^*(\nu_k)\big)\big] \,. \end{aligned} \tag{4.21}$$

Thus, the optimal total cost is given by

$$J^*([\mathbf{q}_0, \pi_0]) \;=\; \sum_{k=0}^{\infty} \mathsf{E}\left[\widetilde{g}\big(\nu_k, \widetilde{\mu}^*(\nu_k)\big) \,\middle|\, \nu_0 = [\mathbf{q}_0, \pi_0]\right]. \tag{4.22}$$

The optimal cost function satisfies the following Bellman's equation

$$J^*([\mathbf{q},\pi]) \;:=\; \min\left\{1-\pi, c\pi + \mathsf{E}\left[J^*\left(\mathbf{Q}_{k+1}, \phi_\Pi(\nu_k, \mathbf{Z}_{k+1})\right) \,\middle|\, \nu_k = [\mathbf{q},\pi]\right]\right\}. \tag{4.23}$$

where the function $\phi_\Pi(\nu_k, \mathbf{Z}_{k+1})$ is provided in the Appendix.

Remark 4.1 The optimal stationary Markov policy (i.e., the optimum stopping rule τ) in general depends on $\mathbf{Q}$. Hence, the decision delay and the queueing delay are coupled, unlike in the NODM case.

We characterize the optimal policy in the following theorem.

Theorem 4.3 *The optimal stopping rule τ^* is a network–state dependent threshold rule on the a posteriori probability Π_k, i.e., there exists thresholds $\gamma(\mathbf{q})$ such that*

$$\tau \;=\; \inf\{k \geqslant 0 : \Pi_k \geqslant \gamma(\mathbf{Q}_k)\} \tag{4.24}$$

In general, the thresholds $\gamma(\mathbf{Q}_k)$s (i.e., optimum policy) can be numerically obtained by solving Eqn. 4.23 using value iteration method (see pp. 88–90, [Bertsekas, 2000b]). However, computing the optimal policy for the NADM procedure is hard as the state space is huge even for moderate values of n. Hence, we resort to a suboptimal policy based on the following threshold rule, which is motivated by the structure of the optimal policy.

$$\tau \quad = \quad \inf\{k \geqslant 0 : \Pi_k \geqslant \gamma\} \tag{4.25}$$

where γ is chosen such that $\mathsf{P}\left\{\tau < T\right\} = \alpha$ is met.

Thus, we have formulated a sequential change detection problem when the sensor observations are sent to the decision maker over a random access network, and the fusion centre processes the samples in the NADM mode. The information for decision making now needs to include the network state $\mathbf{Q}_k$ (in addition to the samples received by the decision maker); we have shown that $[\mathbf{Q}_k, \Pi_k]$ is sufficient for the *information* history $\mathbf{I}_k$. Also, we have provided the structure for the optimal policy. Since, obtaining the optimal policy is computationally hard, we gave a simple threshold based policy, which is motivated by the structure of the optimal policy.

Remark: The communication channel between the sensor nodes and the fusion centre can be noisy, and there can be noise at the fusion centre as well. The effect of the channel impairments and the noise at the fusion centre can be seen in the form of dropping of packets that carry the samples from the sensors. If at time k, the fusion centre has not yet received a sample from a sensor i of bth batch, but receives a sample (from the sensor i) of a later batch b' (where $b' > b$), then it means that the packets of sensor i that carried the samples from batches $b, b+1, \cdots, b'-1$ are dropped. We will still get a POMDP with a large complexity. However, if the probability of packet drop can be neglected (since the number of retransmissions is quite large, the probability of packet drop is very small even for fairly large packet error rates (even 10% to 20%)), the analysis in this Chapter directly applies.

4.6 Numerical Results

Minimising the mean detection delay not only requires an optimal decision rule at the fusion centre but also involves choosing the optimal values of the sampling rate r, and the number of sensors n. To explore this, we obtain the minimum decision delay for each value of the sampling rate r numerically, and the network delay via simulation. We use Eqn. 4.25 to obtain the mean detection delay for NADM, and Theorem 4.1 to obtain the mean detection delay for NODM procedure.

4.6.1 Optimal Sampling Rate

Consider a sensor network with n nodes. For a given probability of false alarm, the decision delay (detection delay without the network–delay component) decreases with increase in sampling rate. This is due to the increase in the number of samples that the fusion centre receives within a given time. But, as the sampling rate increases, the network delay increases due to the increased packet communication load in the network. Therefore it is natural to expect the existence of a sampling rate r^*, with $r^* < \sigma/n$, (the sampling rate should be less than σ/n, for the queues to be stable; see Theorem 4.2) that optimizes the tradeoff between these two components of detection delay. Such an r^*, in the case of NODM can be obtained by minimising the following expression over r (refer Theorem 1 of [Premkumar et al.]).

$$(d(r) + l(r))\,(1-\alpha) - \rho \cdot l(r) + \frac{1}{r}\min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K} - K\right]^+$$

Note that in the above expression, the delay term $\min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K} - K\right]^+$ also depends on the sampling rate r via the probability of change $p_r = 1-(1-p)^{(1/r)}$. The delay due to coarse sampling $l(r)(1-\alpha) - \rho \cdot l(r)$ can be found analytically (see Appendix). We can approximate the delay $\min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K} - K\right]^+$ by the asymptotic (as $\alpha \to 0$) delay as $\frac{|\ln(\alpha)|}{n\mathrm{KL}(f_1,f_0)+|\ln(1-p_r)|}$ where $\mathrm{KL}(f_1, f_0)$ is the Kullback–Leibler divergence between the pdfs f_1 and f_0 (see [Tartakovsky and Veeravalli, 2005]). But, obtaining the network–delay (i.e., $d(r)(1-\alpha)$) analytically is hard, and hence an analytical characterisation of r^* appears intractable. Hence, we have resorted to evaluation via simulation and numerical computation.

The distribution of sensor observations are taken to be $\mathcal{N}(0,1)$ and $\mathcal{N}(1,1)$, before and after the change respectively for all the 10 nodes. We choose $\rho = 0$ and the probability of occurrence of change in a slot to be $p = 0.0005$, i.e., the mean time until change is 2000 slots. $\min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K} - K\right]^+$ and $d(r)$ are obtained from simulation for $\alpha = 0.01$ and $\sigma = 0.3636$, for 1000 simulation runs, and the expression for mean detection delay (displayed above) is plotted against r in Figure 4.14. In Figure 4.14, we also plot the approximate mean detection delay which is obtained through the expression for $l(r)$ and the approximation, $\min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K} - K\right]^+ \approx \frac{|\ln(\alpha)|}{nI(f_1,f_0)+|\ln(1-p_r)|}$. We study this approximation as this provides an (approximate) explicit expression for the mean decision delay. The delay in the FJQ–GPS does not have a closed form expression. Hence, we still need simulation for the delay due to queueing network. Note that at time $k = 0$, the length of the queues are set of zero. The mean detection delay due to NADM procedure (see Eqn. 4.25) is also plotted in Figure 4.14.

As would have been expected, we see from Figure 4.14 that the NADM procedure has a better mean detection delay performance than the NODM procedure. Note that $\sigma/n = 0.03636$ and hence for the queues to be stable (see Theorem 4.2), the sampling rate has to be less that $\sigma/n = 0.03636$ ($1/28 < 0.03636 < 1/27$). As the sampling rate r increases to 1/28 (the maximum allowed sampling rate), the queueing delay increases rapidly. This is evident from Figure 4.14. Also, we see from Figure 4.14 that operating at a sampling rate around $1/34 (\approx 0.0294)$ samples/slot would be optimum. The optimal sampling rate is found to be approximately the same for NODM and NADM. At the optimal sampling rate the mean detection delay of NODM is 90 slots and that of NADM is 73 slots.

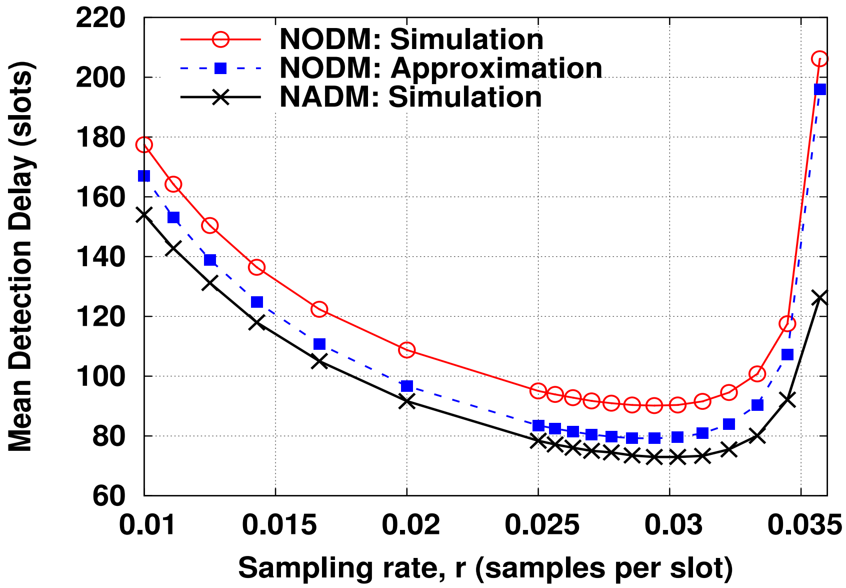


Figure 4.14: Mean detection delay for $n = 10$ nodes is plotted against the sampling rate r for both NODM and NADM. For NODM, an approximate analysis is also plotted. This was obtained with the prior probability $\rho = 0$, $p = 0.0005$, probability of false alarm target $\alpha = 0.01$, $\sigma = 0.3636$ and with the sensor observations being $\mathcal{N}(0,1)$ and $\mathcal{N}(1,1)$, before and after the change respectively.

4.6.2 Optimal Number of Sensor Nodes (Fixed Observation Rate)

Now let us consider fixing $n \times r$. This is the number of observations the fusion centre receives per slot in a network with n nodes sampling at a rate r (samples per slot). It is also a measure of the energy spent by the network per slot. Since it has been assumed that the observations are conditionally independent and identically distributed across the sensors and over time, it is natural to ask how beneficial it is to have more nodes sampling at a lower rate, when compared to fewer nodes sampling at a higher rate with the number of observations per slot being the same. With $\rho = 0$, $p = 0.0005$, $\alpha = 0.01$, and $\sigma = 0.3636$, and $f_0 \sim \mathcal{N}(0,1)$ and $f_1 \sim \mathcal{N}(1,1)$, and from 1000 simulation runs, we present simulation results for two examples, the first one being $nr = 1/3$ (the case of a heavily loaded network) and the second one being $nr = 1/100$ (the case of a lightly loaded network, $nr \ll \sigma$).

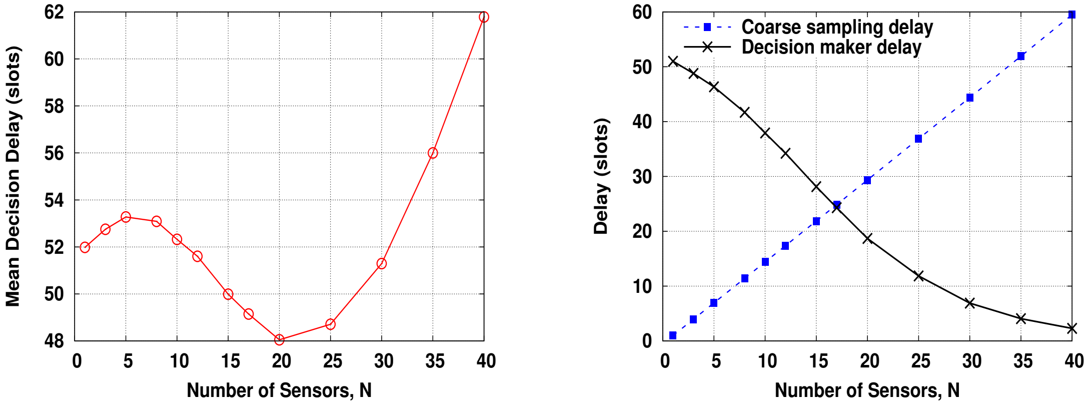


Figure 4.15: Mean *decision* delay of NODM procedure for $n \times r = 1/3$ is plotted against the the number of nodes n. The plot is obtained with $\rho = 0$, $p = 0.0005$, $\alpha = 0.01$ and with the sensor observations being $\mathcal{N}(0,1)$ and $\mathcal{N}(1,1)$, before and after the change respectively. The components of the mean decision delay, i.e., the coarse sampling delay $(1-\alpha)l(r) - \rho l(r)$, and the decision maker delay, $\frac{1}{r}\min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K} - K\right]^+$ are shown on the right.

Figure 4.15 shows the plot of mean decision delay, $l(r)(1-\alpha-\rho)+\frac{1}{r}\min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K} - K\right]^+$ versus the number of sensors when $nr = 1/3$. As n increases, the sampling rate $r = 1/(3n)$ decreases and hence the coarse sampling delay $l(r)(1-\alpha)$ increases; this can be seem to be approximately linear by analysis of the expression for $l(r)$ given in the Appendix. Also, as n increases, the decision maker gets more samples at the decision instants and hence the delay due to the decision maker $\frac{1}{r}\min_{\Pi_\alpha} \mathsf{E}\left[\widetilde{K} - K\right]^+$ decreases (this is evident from the right side of Figure 4.15). Figure 4.15 shows that in the region where n is large (i.e., $n \geqslant 20$) or n is very small (i.e., $n < 5$), as n increases, the mean decision delay increases. This is because in this region as n increases, the decrease in the delay due to decision maker is smaller compared to the increase in the delay due to coarse sampling. However, in the region where n is moderate (i.e., for $5 \leqslant n < 20$), as n increases, the decrease in the delay due to decision maker is large compared to the increase in the delay due to coarse sampling. Hence in this region, the mean decision delay decreases with n. Therefore, we infer that when $n \times r = \frac{1}{3}$, deploying 20 nodes sampling at $1/60$ samples per slot is optimal, when there is no network delay.

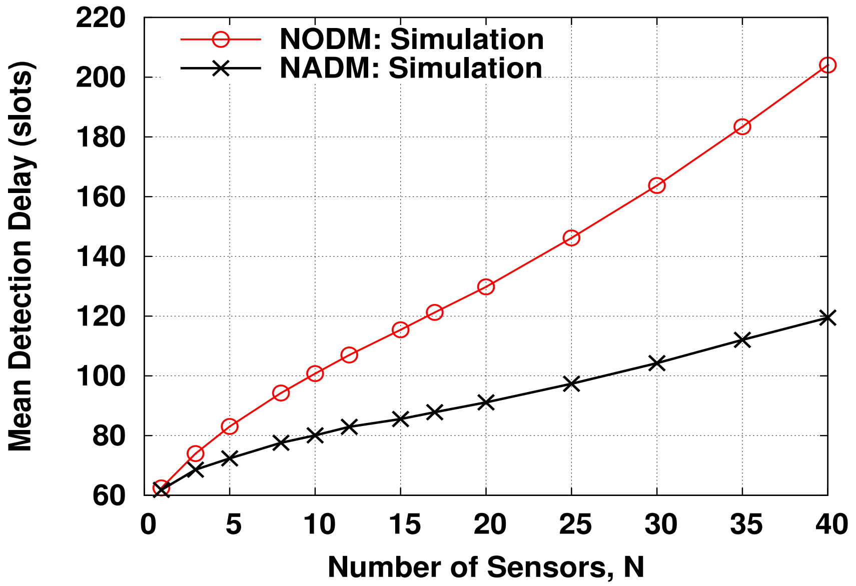


Figure 4.16: Mean *detection* delay for $n \times r = 1/3$ is plotted against the number of nodes n. This was obtained with $\rho = 0$, $p = 0.0005$, $\alpha = 0.01$ $\sigma = 0.3636$ and with the sensor observations being $\mathcal{N}(0,1)$ and $\mathcal{N}(1,1)$, before and after the change respectively.

Figure 4.16 shows the mean detection delay (i.e., the network delay plus the decision delay shown in Figure 4.15) versus the number of nodes n for a fixed $n \times r = 1/3$. As the the number of nodes n increases, the sampling rate $r = 1/(3n)$ decreases. For large n (and equivalently small r), in the case of NODM with the Shiryaev procedure, the network delay, $d(r) \approx \frac{n}{\sigma}$ as it requires n (independent) successes, each with probability σ, in the random access network to transport a batch of n samples (also, since the sampling rate r is small, one would expect that a batch is delivered before a new batch is generated) and the decision maker requires just one batch of n samples to stop (after the change occurs). Hence, for large n, the detection delay is approximately $l(r)(1-\alpha)+d(r)(1-\alpha) \approx l(r)(1-\alpha)+\frac{n}{\sigma}(1-\alpha)$. It is to be noted that for large n, to achieve a false alarm probability of α, the decision maker requires $N_\alpha < n$ samples (the mean of the log–likelihood ratio, LLR of received samples, after change, is the Kullback–Leibler divergence between pdfs f_1 and f_0, given by $I(f_1, f_0) > 0$. Hence, the posterior probability, which is a function of LLR, increases with the the number of received samples. Thus, to cross a threshold of $\gamma(\alpha)$, we need N_α samples. Thus, for large n, in the NADM procedure, the detection

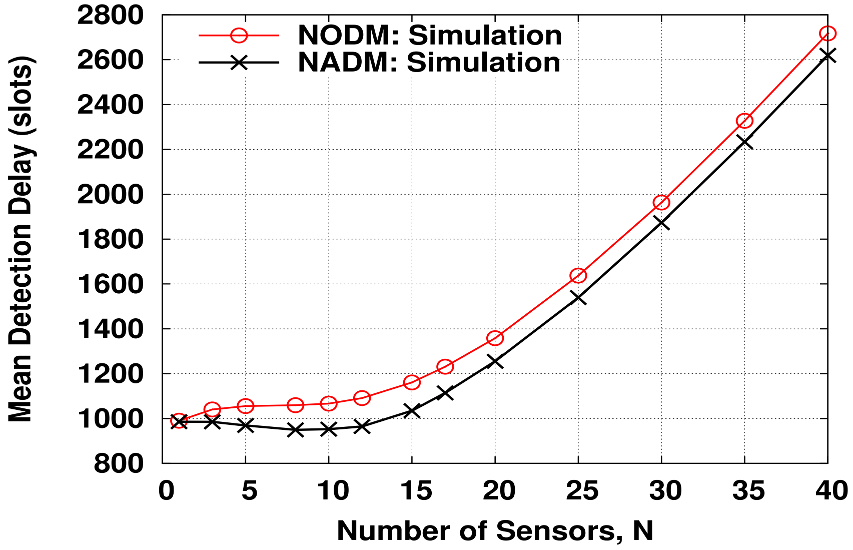


Figure 4.17: Mean *detection* delay for $n \times r = 0.01$ is plotted against the the number of nodes n. This was obtained with $\rho = 0$, $p = 0.0005$, $\alpha = 0.01$ and with the sensor observations being $\mathcal{N}(0,1)$ and $\mathcal{N}(1,1)$, before and after the change respectively.

delay is approximately $l(r)(1-\alpha) + \frac{N_\alpha}{\sigma}(1-\alpha)$, where N_α/σ is the mean network–delay to transport N_α samples. Thus, for large n, the difference in the mean detection delay between NODM and NADM procedures is approximately $\frac{1-\alpha}{\sigma}(n - N_\alpha)$. Note that N_α depends only on α and hence the quantity $\frac{1-\alpha}{\sigma}(n - N_\alpha)$ increases with n. This behaviour is in agreement with Figure 4.16. Also, as $n \times r = 1/3$, we expect the network delay to be very large (as $1/3$ is close to $\sigma = 0.3636$) and hence having a single node is optimal which is also evident from Figure 4.16.

It is also possible to find an example where the optimal number of nodes is greater than 1. For example this occurs in the above setting for $n \times r = 0.01$ (see Figure 4.17). Note that having $n = 10$ sensors is optimal for the NADM procedure. The NODM procedure makes the decision only when it receives a batch of n samples corresponding to a sampling instant, whereas NADM procedure makes the decision at every time slot irrespective of whether it receives a sample in that time slot or not. Thus, the Bayesian update that NADM does at every time slot makes it stop earlier than NODM.

4.7 Conclusion

In this chapter, we have considered the problem of minimising the mean detection delay in an event detection on ad hoc wireless sensor network. We provide two ways of processing samples in the fusion centre: i) *Network Oblivious* (NODM) processing, and ii) *Network Aware* (NADM) processing. We show that in NODM processing, under periodic sampling, the detection delay decouples into decision and network delays. An important implication of this is that an optimal sequential change detection algorithm can be used in the decision device independently of the random access network. We also formulate and solve the change detection problem in the NADM setting in which case the optimal decision maker needs to use the network state in its optimal stopping rule. Also, we study the network delay involved in this problem and show that it is important to operate at a particular sampling rate to achieve the minimum detection delay.

4.8 Appendix

Evolution of $\mathbf{Q}_k$

Let $\mathbf{Q}_{k+1} = [\lambda_{k+1}, B_{k+1}, \Delta_{k+1}, \mathbf{W}_{k+1}, \mathbf{R}_{k+1}] = \phi_{\mathbf{Q}}(\mathbf{Q}_k, M_k)$, where we define the evolution of each of the components of the state below.

- λ_{k+1}: If $\lambda_k = 1$, then there is only one slot–to–go for the next sampling instant and hence $\lambda_{k+1} = 1/r$ (as there are $1/r$ slots–to–go for the next sampling instant from slot $k+1$). If $\lambda_k > 1$, then the number of slots–to–go for the next sampling instant is $\lambda_k - 1$, i.e.,

$$\lambda_{k+1} \;=\; \phi_\lambda(\mathbf{Q}_k, M_k) \;:=\; \frac{1}{r}\mathbf{1}_{\{\lambda_k=1\}} + (\lambda_k - 1)\mathbf{1}_{\{\lambda_k>1\}} \tag{4.26}$$

- B_{k+1}: B_{k+1} can be either B_k or B_k+1. The only event that makes $B_{k+1} = B_k+1$ is as follows: At the beginning of slot k, the decision maker has already received and processed all components, other than the jth component (for some j), of batch B_k (i.e., $R_k^{(j)} = 0$ and $\sum_{i=1}^n R_k^{(i)} = n-1$) and during time slot k, there is a successful transmission from sensor node j (i.e., $M_k = j$). Thus,

$$\begin{aligned} B_{k+1} &= \phi_B(\mathbf{Q}_k, M_k) \\ &:= \begin{cases} B_k + 1 & \text{if } M_k = j > 0, R_k^{(j)} = 0, \sum\limits_{i=1}^n R_k^{(i)} = n-1 \\ B_k & \text{otherwise.} \end{cases} \end{aligned} \tag{4.27}$$

- Δ_{k+1}: At time $k+1$, Δ_{k+1} denotes the number of slots back at which the batch that is currently being processed is generated. Δ_{k+1} can be 0 or Δ_k+1 or Δ_k+1-1/r. $\Delta_{k+1} = 0$ if the batch that is currently expected at time $k+1$ by the decision maker is not yet generated or has been generated at time $k+1$. $\Delta_{k+1} = \Delta_k + 1$ if the batch that is currently being processed at time $k+1$ is the same as the batch that was expected or being processed at time k. $\Delta_{k+1} = \Delta_k + 1 - 1/r$ if the batch that is currently expected or being processed at time $k+1$ is B_k+1, where B_k is the batch that was being processed at time k. Δ_{k+1} is given by

$$\begin{aligned}
&\Delta_{k+1} \\
:= \quad & \max\left\{k+1-\frac{B_{k+1}}{r},0\right\} \\
= \quad & \max\left\{k-\frac{B_k}{r}+1-\frac{B_{k+1}-B_k}{r},0\right\} \\
= \quad & \max\left\{\left(k-\frac{B_k}{r}\right)\mathbf{1}_{\{(k-B_k/r)>0\}}+\left(k-\frac{B_k}{r}\right)\mathbf{1}_{\{(k-B_k/r)\leqslant 0\}}+1-\frac{B_{k+1}-B_k}{r},0\right\} \\
= \quad & \max\left\{\left(k-\frac{B_k}{r}\right)\mathbf{1}_{\{\Delta_k>0\}}+\left(k-\frac{B_k}{r}\right)\mathbf{1}_{\{\Delta_k=0\}}+1-\frac{B_{k+1}-B_k}{r},0\right\} \\
= \quad & \max\left\{\Delta_k\mathbf{1}_{\{\Delta_k>0\}}+\left(k-\frac{B_k}{r}\right)\mathbf{1}_{\{\Delta_k=0\}}+1-\frac{B_{k+1}-B_k}{r},0\right\} \\
= \quad & \max\left\{\Delta_k\mathbf{1}_{\{\Delta_k>0\}}+\left(k-\frac{B_k}{r}\right)\mathbf{1}_{\{\Delta_k=0\}}\left(\mathbf{1}_{\{k-B_k/r=0\}}+\mathbf{1}_{\{k-B_k/r<0\}}\right)+1-\frac{B_{k+1}-B_k}{r},0\right\} \\
= \quad & \max\left\{\Delta_k\mathbf{1}_{\{\Delta_k>0\}}+\left(k-\frac{B_k}{r}\right)\mathbf{1}_{\{\Delta_k=0\}}\left(\mathbf{1}_{\{k=B_k/r\}}+\mathbf{1}_{\{(B_k-1)/r<k<B_k/r\}}\right)+1-\frac{B_{k+1}-B_k}{r},0\right\} \\
= \quad & \max\left\{\Delta_k\mathbf{1}_{\{\Delta_k>0\}}+\mathbf{1}_{\{\Delta_k=0\}}\left(0\cdot\mathbf{1}_{\{\lambda_k=1/r\}}+\left(k-\frac{B_k}{r}\right)\cdot\mathbf{1}_{\{\lambda_k\neq 1/r\}}\right)+1-\frac{B_{k+1}-B_k}{r},0\right\} \qquad (4.28)
\end{aligned}$$

We interpret the above equation as follows.

- ○ If at the beginning of time slot k, there are outstanding samples of batch B_k (i.e., $\Delta_k > 0$), then at the beginning of the time slot $k+1$, there are two possibilities:
  - ∗ $B_{k+1} = B_k$, which means that there are still outstanding samples of batch B_k in the queues of the sensor node. Thus $\Delta_{k+1} = \Delta_k + 1$.
  - ∗ $B_{k+1} = B_k + 1$, which means that all samples of batch B_k are received by the fusion centre and hence, the delay of the next sample, if already generated, is $\Delta_{k+1} = (\Delta_k + 1 - 1/r)^+$.
- ○ If at the beginning of time slot k, there are no outstanding samples of batch B_k (i.e., $\Delta_k = 0$), then it means that the samples of batch B_k are not yet generated if $\lambda_k \neq 1/r$ or just been generated at time k if $\lambda_k = 1/r$. Thus, in this case also, at the beginning of the time slot $k+1$, there are two possibilities:
  - ∗ $\lambda_k = 1/r$, which means that the samples of batch B_k are generated at

k and if $B_{k+1} = B_k$, this means that some of these samples are still in sensor node queues. Hence $\Delta_{k+1} = 1$.

* $\lambda_k \neq 1/r$, which means that the samples of batch B_k are not yet generated and hence $\Delta_{k+1} = 0$.

- The event that $\Delta_k = 0$, $B_{k+1} = B_k + 1$ can happen only if $n = 1$, $1/r = 1$ and there is a successful transmission during slot k, in which case, $\Delta_{k+1} = 0$.

This is summarized as follows.

$$\begin{aligned} \Delta_{k+1} &= \phi_\Delta(\mathbf{Q}_k, M_k) \\ &:= \begin{cases} 0 & \text{if } \Delta_k = 0, B_{k+1} = B_k, \lambda_k \neq 1/r \\ 1 & \text{if } \Delta_k = 0, B_{k+1} = B_k, \lambda_k = 1/r \\ 0 & \text{if } \Delta_k = 0, B_{k+1} = B_k + 1 \\ \Delta_k + 1 & \text{if } \Delta_k > 0, B_{k+1} = B_k \\ (\Delta_k + 1 - \frac{1}{r})^+ & \text{if } \Delta_k > 0, B_{k+1} = B_k + 1. \end{cases} \end{aligned} \tag{4.29}$$

Note that B_{k+1} is obtained using $\phi_B(\mathbf{Q}_k, M_k)$ (see Eqn. 4.27).

- $W_{k+1}^{(i)}$
  - **No successful transmission** ($M_k = 0$)**:** In this case, the sequencer queue is not affected at the beginning of time slot $k+1$, i.e., $W_{k+1}^{(i)} = W_k^{(i)}$.
  - **Successful transmission of node** i**'s current sample** ($M_k = i, R_k^{(i)} = 0$)**:** In this case, the sample is delivered to the decision maker as soon as it is received and hence, the sequencer queue is not affected, at the beginning of time slot $k+1$. Note that $W_{k+1}^{(i)} = W_k^{(i)} = 0$.
  - **Successful transmission of node** i**'s later sample** ($M_k = i, R_k^{(i)} = 1$)**:** In this case, the received sample is an out–of–sequence sample and hence it is queued in the sequencer queue, i.e., $W_{k+1}^{(i)} = W_k^{(i)} + 1$.

- ○ **Successful transmission of node j's sample** $(M_k = j \neq i)$**:** In this case, if the received sample corresponds to the last sample of batch B_k (i.e., $R_k^{(j)} = 0$ and $\sum_{i=1}^n R_k^{(i)} = n-1$), then the HOL packet, if any, in the sequencer queues are delivered to the decision maker. i.e., $W_{k+1}^{(i)} = (W_k^{(i)} - 1)^+$.

  On the other hand, if in this case, the received sample is not the last sample of batch B_k (i.e., $\sum_{i=1}^n R_k^{(i)} < n-1$), then the sequencer queue is not affected at the beginning of time slot $k+1$. i.e., $W_{k+1}^{(i)} = W_k^{(i)}$.

The above points are summarized as follows:

$$\begin{aligned} W_{k+1}^{(i)} &= \phi_{W^{(i)}}(\mathbf{Q}_k, M_k) \\ &:= \begin{cases} W_k^{(i)} & \text{if } M_k = 0, \\ 0 = W_k^{(i)} & \text{if } M_k = i, R_k^{(i)} = 0 \\ W_k^{(i)} + 1 & \text{if } M_k = i, R_k^{(i)} = 1 \\ W_k^{(i)} & \text{if } M_k = j \notin \{0, i\} \text{ and } \sum_{s=1}^n R_k^{(s)} < n-1, \\ (W_k^{(i)} - 1)^+ & \text{if } M_k = j \notin \{0, i\}\ R_k^{(j)} = 0, \text{and } \sum_{s=1}^n R_k^{(s)} = n-1 \end{cases} \end{aligned} \tag{4.30}$$

- $R_{k+1}^{(i)}$

  - ○ **No successful transmission** $(M_k = 0)$**:** In this case, $R^{(i)}$ is not affected at the beginning of time slot $k+1$, i.e., $R_{k+1}^{(i)} = R_k^{(i)}$.
  - ○ **Successful transmission of node i's sample** $(M_k = i)$**:** In this case, the sample is delivered to the decision maker if it is not an out–of–sequence sample; otherwise, it is stored in the sequencer queue. In either case, the decision maker must have received the sample $X_{B_k}^{(i)}$ at the beginning of time slot $k+1$. Note that $R_{k+1}^{(i)} = R_k^{(i)} = 1$.
  - ○ **Successful transmission of node j's sample** $(M_k = j \neq i)$**:** There are two cases here.
    - ∗ **Case 1:** The received sample is not the last sample of batch B_k (i.e., $\sum_{i=1}^n R_k^{(i)} < n-1$ or $R_k^{(j)} = 1$ and $\sum_{i=1}^n R_k^{(i)} = n-1$). In this case, the

state of $R^{(i)}$ remains the same at the beginning of time slot $k+1$, i.e., $R^{(i)}_{k+1} = R^{(i)}_k$.

- $*$ **Case 2:** The received sample is the last sample of batch B_k (i.e., $R^{(j)}_k = 0$ and $\sum_{i=1}^{n} R^{(i)}_k = n-1$). In this case, the state of $R^{(i)}_{k+1}$ remains 1 at the beginning of time slot $k+1$ if $W^{(i)}_k > 0$; otherwise, $R^{(i)}_{k+1} = 0$, i.e., $R^{(i)}_{k+1} = \mathbf{1}_{\{W^{(i)}_k > 0\}}$.

The above points are summarized as follows:

$$\begin{aligned} R^{(i)}_{k+1} &= \phi_{R^{(i)}}(\mathbf{Q}_k, M_k) \\ &:= \begin{cases} R^{(i)}_k & \text{if } M_k = 0, \\ 1 & \text{if } M_k = i, \\ R^{(i)}_k & \text{if } M_k = j \notin \{0,i\}, \sum_{s=1}^{n} R^{(s)}_k < n-1, \\ R^{(i)}_k & \text{if } M_k = j \notin \{0,i\}, R^{(j)}_k = 1, \sum_{s=1}^{n} R^{(s)}_k = n-1, W^{(i)}_k = 0 \\ 0 & \text{if } M_k = j \notin \{0,i\}, R^{(j)}_k = 0, \sum_{s=1}^{n} R^{(s)}_k = n-1, W^{(i)}_k = 0 \\ 1 & \text{if } M_k = j \notin \{0,i\}, R^{(j)}_k = 0, \sum_{s=1}^{n} R^{(s)}_k = n-1, W^{(i)}_k > 0 \end{cases} \end{aligned}$$

■

Distribution of state noise N

Let $\mathbf{q} = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}]$. Note that $\mathsf{P}\left\{M_k = m \,\middle|\, \mathbf{Q}_k = \mathbf{q}, \mathbf{\Theta}_k = \boldsymbol{\theta}\right\} = \mathsf{P}\left\{M_k = m \,\middle|\, \mathbf{Q}_k = \mathbf{q}\right\}$ and is given by

$$\mathsf{P}\left\{M_k = 0 \,\middle|\, \mathbf{Q}_k = \mathbf{q}\right\} = \begin{cases} 1 & \text{if } \phi_N(\mathbf{q}) = 0 \\ 1 - \sigma & \text{if } \phi_N(\mathbf{q}) > 0 \end{cases}$$

$$\mathsf{P}\left\{M_k = m \,\middle|\, \mathbf{Q}_k = \mathbf{q}\right\} = \begin{cases} 0 & \text{if } \phi_N(\mathbf{q}) = 0 \\ \frac{\sigma}{\phi_N(\mathbf{q})} & \text{if } \phi_{L^{(m)}}(\mathbf{q}) > 0, \quad m = 1, 2, 3, \cdots, n. \end{cases}$$

where $\phi_N(\mathbf{q})$ and $\phi_{L^{(m)}}(\mathbf{q})$ are obtained from Eqns. 4.9 and 4.8. The distribution function, $\mathsf{P}\left\{O_k = o \,\middle|\, \mathbf{Q}_k = \mathbf{q}, \mathbf{\Theta}_k = \boldsymbol{\theta}\right\} = \mathsf{P}\left\{O_k = o \,\middle|\, \mathbf{Q}_k = \mathbf{q}, \Theta_k = \theta\right\}$ is given by

$$\mathsf{P}\left\{O_k = o \,\middle|\, \mathbf{Q}_k = \mathbf{q}, \Theta_k = 0\right\} = \begin{cases} 1 - p & \text{if } o = 0 \\ p & \text{if } o = 1, \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathsf{P}\left\{O_k = o \,\middle|\, \mathbf{Q}_k = \mathbf{q}, \Theta_k = 1\right\} = \begin{cases} 1 & \text{if } o = 0 \\ 0 & \text{otherwise.} \end{cases}$$

■

Proof of Lemma 4.1

Let $\mathbf{q} = [\lambda, b, \delta, \mathbf{w}, \mathbf{r}]$. From Eqn. 4.18,

$$\begin{aligned}
\Pi_k &:= \mathsf{P}\left\{T \leqslant k \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\} \\
&= \mathsf{P}\left\{T \leqslant k-\delta \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\} + \ \mathsf{P}\left\{k-\delta < T \leqslant k \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\} \\
&= \mathsf{P}\left\{T \leqslant k-\delta \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\} \\
&\quad + \mathsf{P}\left\{T > k-\delta \,\middle|\, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\} \cdot \mathsf{P}\left\{T \leqslant k \,\middle|\, T > k-\delta, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\}, \\
&= \Psi_k + (1-\Psi_k) \cdot \mathsf{P}\left\{T \leqslant k \,\middle|\, T > k-\delta, \mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k\right\}, \\
&= \Psi_k + (1-\Psi_k) \cdot \frac{\mathsf{P}\{k-\delta < T \leqslant k\}\, \mathsf{P}\left\{\mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k \,\middle|\, k-\delta < T \leqslant k\right\}}{\mathsf{P}\{T > k-\delta\}\, \mathsf{P}\left\{\mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k \,\middle|\, T > k-\delta\right\}} \\
&= \Psi_k + (1-\Psi_k) \cdot \frac{\mathsf{P}\{k-\delta < T \leqslant k\}}{\mathsf{P}\{T > k-\delta\}} \qquad (4.31) \\
&= \Psi_k + (1-\Psi_k)\left(1-(1-p)^{\delta}\right) \qquad (4.32)
\end{aligned}$$

Eqn. 4.31 is justified as follows. Note that

$$\begin{aligned}
&\mathsf{P}\left\{\mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k \,\middle|\, k-\delta < T \leqslant k\right\} \\
= \ &\mathsf{P}\left\{\mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}, \mathbf{X}_{[1:B_k-1]}, \{X_{B_k}^{(i)} : R_k^{(i)} = 1\}, u_{[0:k-1]} \,\middle|\, k-\delta < T \leqslant k\right\} \\
= \ &\mathsf{P}\left\{\mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q} \,\middle|\, k-\delta < T \leqslant k\right\} \\
&\cdot \mathsf{P}\left\{\mathbf{X}_{[1:B_k-1]}, \{X_{B_k}^{(i)} : R_k^{(i)} = 1\} \,\middle|\, k-\delta < T \leqslant k, \mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}\right\} \\
&\cdot \mathsf{P}\left\{u_{[0:k-1]} \,\middle|\, k-\delta < T \leqslant k, \mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}, \mathbf{X}_{[1:B_k-1]}, \{X_{B_k}^{(i)} : R_k^{(i)} = 1\}\right\} \\
= \ &\mathsf{P}\left\{\mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}\right\} \cdot \mathsf{P}\left\{\mathbf{X}_{[1:B_k-1]}, \{X_{B_k}^{(i)} : R_k^{(i)} = 1\} \,\middle|\, k-\delta < T, \mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}\right\} \\
&\cdot \mathsf{P}\left\{u_{[0:k-1]} \,\middle|\, \mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}, \mathbf{X}_{[1:B_k-1]}, \{X_{B_k}^{(i)} : R_k^{(i)} = 1\}\right\} \\
= \ &\mathsf{P}\left\{\mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q} \,\middle|\, T > k-\delta\right\} \cdot \mathsf{P}\left\{\mathbf{X}_{[1:B_k-1]}, \{X_{B_k}^{(i)} : R_k^{(i)} = 1\} \,\middle|\, T > k-\delta, \mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}\right\} \\
&\cdot \mathsf{P}\left\{u_{[0:k-1]} \,\middle|\, \middle|\, T > k-\delta, \mathbf{Q}_{[0:k-1]}, \mathbf{Q}_k = \mathbf{q}, \mathbf{X}_{[1:B_k-1]}, \{X_{B_k}^{(i)} : R_k^{(i)} = 1\}\right\} \\
= \ &\mathsf{P}\left\{\mathbf{I}_{k-1}, \mathbf{Q}_k = \mathbf{q}, \mathbf{Y}_k \,\middle|\, T > k-\delta\right\}.
\end{aligned}$$

We use the following facts in the above justification: i) the evolution of the queueing system $\mathbf{Q}_k$ is independent of the change point T, ii) whenever $T > k-\delta$, the distribution of any sample $X_h^{(i)}$, $h \leqslant B_k$ is f_0, and iii) the control $u_k = \tilde{\mu}(\mathbf{I}_k)$. ■

Recursive computation of Π_k

At time k, based on the index of the node that successfully transmits a packet M_k, the set of all sample paths Ω can be partitioned based on the following events,

$$\begin{aligned}
\mathcal{E}_{1,k} &:= \left\{\omega : M_k(\omega) = 0 \text{ or } M_k(\omega) = j > 0, R_k^{(j)}(\omega) = 1\right\}\\
\mathcal{E}_{2,k} &:= \left\{\omega : M_k(\omega) = j > 0, R_k^{(j)}(\omega) = 0, \sum_{i=1}^{n} R_k^{(i)}(\omega) < n-1\right\}\\
\mathcal{E}_{3,k} &:= \left\{\omega : M_k(\omega) = j > 0, R_k^{(j)}(\omega) = 0, \sum_{i=1}^{n} R_k^{(i)}(\omega) = n-1\right\},
\end{aligned}$$

i.e., $\Omega = \mathcal{E}_{1,k} \cup \mathcal{E}_{2,k} \cup \mathcal{E}_{3,k}$. We note that the above events can also be described by using $\mathbf{Q}_k$ and $\mathbf{Q}_{k+1}$ in the following manner

$$\begin{aligned}
\mathcal{E}_{1,k} &= \{\omega : \mathbf{W}_{k+1}(\omega) = \mathbf{W}_k(\omega), \mathbf{R}_{k+1}(\omega) = \mathbf{R}_k(\omega)\}\\
&\quad \bigcup \{\omega : \mathbf{W}_{k+1}(\omega) = \mathbf{W}_k(\omega) + \mathbf{e}_j, \mathbf{R}_{k+1}(\omega) = \mathbf{R}_k(\omega)\}\\
\mathcal{E}_{2,k} &= \{\omega : \mathbf{W}_{k+1}(\omega) = \mathbf{W}_k(\omega), \mathbf{R}_{k+1}(\omega) = \mathbf{R}_k(\omega) + \mathbf{e}_j\}\\
\mathcal{E}_{3,k} &= \left\{\omega : \sum_{i=1}^{n} R_k^{(i)}(\omega) = n-1, \forall i, W_{k+1}^{(i)}(\omega) = (W_k^{(i)}(\omega) - 1)^+, R_{k+1}^{(i)}(\omega) = \mathbf{1}_{\{W_k^{(i)} > 0\}}\right\}.
\end{aligned}$$

Here, the events $\mathcal{E}_{1,k}$ and $\mathcal{E}_{2,k}$ represent the case $B_{k+1} = B_k$, and the event $\mathcal{E}_{3,k}$ represents the case $B_{k+1} = B_k + 1$ (i.e., only if the event $\mathcal{E}_{3,k}$ occurs then the batch index is incremented). We are interested in obtaining Π_{k+1} from $[\mathbf{Q}_k, \Pi_k]$ and $\mathbf{Z}_{k+1}$. We show that at time $k+1$, the statistic Ψ_{k+1} (after having observed $\mathbf{Z}_{k+1}$) can be computed in a recursive manner using Ψ_k and $\mathbf{Q}_k$. Using Lemma 4.1 (using Eqn. 4.20) we compute

Π_{k+1} from Ψ_{k+1}.

$$\begin{aligned}
\Psi_{k+1} &= \mathsf{P}\left\{\widetilde{\Theta}_{k+1} = 1 \mid \mathbf{I}_{k+1}\right\} \\
&= \sum_{c=1}^{3} \mathsf{P}\left\{\widetilde{\Theta}_{k+1} = 1, \mathcal{E}_{c,k} \mid \mathbf{I}_{k+1}\right\} \\
&= \sum_{c=1}^{3} \mathsf{P}\left\{\widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{c,k}, \mathbf{I}_{k+1}\right\} \mathbf{1}_{\mathcal{E}_{c,k}} \qquad (\because \mathcal{E}_{c,k} \text{ is } \mathbf{I}_{k+1} \text{ measurable})
\end{aligned}$$

- **Case** $M_k = 0$ **or** $M_k = j > 0$**,** $R_k^{(j)} = 1$:

$$\begin{aligned}
&\Pi_{k+1} \\
= \; & \mathsf{P}\left\{\Theta_{k+1} = 1 \mid \mathcal{E}_{1,k}, \mathbf{I}_{k+1}\right\} \\
= \; & \mathsf{P}\left\{\Theta_{k+1} = 1 \mid \mathcal{E}_{1,k}, \mathbf{I}_k, \mathbf{Q}_{k+1} = \mathbf{q}'\right\} \\
= \; & \frac{\mathsf{P}\left\{\Theta_{k+1} = 1 \mid \mathcal{E}_{1,k}, \mathbf{I}_k\right\} \cdot f_{\mathbf{Q}_{k+1}|\Theta_{k+1}, \mathcal{E}_{1,k}, \mathbf{I}_k}(\mathbf{q}'|1, \mathcal{E}_{1,k}, \mathbf{I}_k)}{f_{\mathbf{Q}_{k+1} \,\big|\, \mathcal{E}_{1,k}, \mathbf{I}_k}(\mathbf{q}'|\mathcal{E}_{1,k}, \mathbf{I}_k)} \qquad \text{(by Bayes rule)} \\
= \; & \mathsf{P}\left\{\Theta_{k+1} = 1 \mid \mathcal{E}_{1,k}, \mathbf{I}_k\right\} \qquad (\mathbf{Q}_{k+1} \text{ is independent of } \Theta_{k+1}) \\
= \; & \mathsf{P}\left\{\Theta_k = 0, \Theta_{k+1} = 1 \mid \mathbf{I}_k\right\} + \mathsf{P}\left\{\Theta_k = 1, \Theta_{k+1} = 1 \mid \mathbf{I}_k\right\} \\
= \; & (1 - \Pi_k)p + \Pi_k
\end{aligned}$$

- **Case** $M_k = j > 0$**,** $R_k^{(j)} = 0$**,** $\sum_{i=1}^{n} R_k^{(i)} < n - 1$: In this case, the sample $X_{B_k}^{(j)}$ is successfully transmitted and is passed on to the decision maker. The decision maker receives just this sample, and computes Π_{k+1}. We compute Ψ_{k+1} from Ψ_k and then we use Lemma 4.1 (using Eqn. 4.20) to compute Π_{k+1} from Ψ_{k+1}.

$$\begin{aligned}
&\Psi_{k+1} \\
= \; & \mathsf{P}\left\{\widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_{k+1}\right\} \\
= \; & \mathsf{P}\left\{\widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', y]\right\} \\
= \; & \mathsf{P}\left\{\widetilde{\Theta}_k = 0, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', y]\right\} \\
& + \mathsf{P}\left\{\widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', y]\right\}
\end{aligned}$$

Since, we consider the case when the fusion center received a sample at time $k+1$ and $B_{k+1} = B_k$, $\Delta_{k+1} = \Delta_k + 1$ and hence, the state $\widetilde{\Theta}_{k+1} = \Theta_{k+1-\Delta_{k+1}} = \Theta_{k-\Delta_k} = \widetilde{\Theta}_k$. Thus, in this case, Ψ_{k+1} can be written as

$$\begin{aligned}
&\Psi_{k+1}\\
=\ & \mathsf{P}\left\{\widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', y]\right\}\\
\overset{(a)}{=}\ & \frac{\mathsf{P}\left\{\widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k\right\} \cdot \mathsf{P}\left\{\mathbf{Q}_{k+1} = \mathbf{q}' \mid \widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1, \mathcal{E}_{2,k}, \mathbf{I}_k\right\}}{\mathsf{P}(\mathbf{Q}_{k+1} = \mathbf{q}' | \mathcal{E}_{2,k}, \mathbf{I}_k) \cdot f_{\mathbf{Y}_{k+1}|\mathcal{E}_{2,k},\mathbf{I}_k,\mathbf{Q}_{k+1}}(y|\mathcal{E}_{2,k}, \mathbf{I}_k, \mathbf{q}')}\\
& \cdot f_{\mathbf{Y}_{k+1}|\widetilde{\Theta}_k,\widetilde{\Theta}_{k+1},\mathcal{E}_{2,k},\mathbf{I}_k,\mathbf{Q}_{k+1}}(y \mid 1, 1, \mathcal{E}_{2,k}, \mathbf{q}', \mathbf{I}_k)\\
\overset{(b)}{=}\ & \frac{\mathsf{P}\left\{\widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k\right\} \cdot \mathsf{P}(\mathbf{Q}_{k+1} = \mathbf{q}' | \mathcal{E}_{2,k}, \mathbf{I}_k) \cdot f_{\mathbf{Y}_{k+1}|\widetilde{\Theta}_k}(y \mid 1)}{\mathsf{P}(\mathbf{Q}_{k+1} = \mathbf{q}' | \mathcal{E}_{2,k}, \mathbf{I}_k) \cdot f_{\mathbf{Y}_{k+1}|\mathcal{E}_{2,k},\mathbf{I}_k,\mathbf{Q}_{k+1}}(y|\mathcal{E}_{2,k}, \mathbf{I}_k, \mathbf{q}')}\\
\overset{(c)}{=}\ & \frac{\mathsf{P}\left\{\widetilde{\Theta}_k = 1, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k\right\} \cdot f_1(y)}{\mathsf{P}\left\{\widetilde{\Theta}_k = 0 \mid \mathcal{E}_{2,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}\right\} \cdot f_{\mathbf{Y}_{k+1}|\widetilde{\Theta}_k}(y|0) + \mathsf{P}\left\{\widetilde{\Theta}_k = 1 \mid \mathcal{E}_{2,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}\right\} \cdot f_{\mathbf{Y}_{k+1}|\widetilde{\Theta}_k}(y|1)}\\
\overset{(d)}{=}\ & \frac{\Psi_k f_1(y)}{(1-\Psi_k) f_0(y) + \Psi_k f_1(y)}
\end{aligned}$$

We explain the steps $(a), (b), (c), (d)$ below.

(a) By Bayes rule, for events A, B, C, D, E, F, we have

$$\mathsf{P}\{AB \mid CDEF\} = \frac{\mathsf{P}\{AB \mid CD\} \mathsf{P}\{E \mid ABCD\} \mathsf{P}\{F \mid ABCDE\}}{\mathsf{P}\{E \mid CD\} \mathsf{P}\{F \mid CDE\}}$$

(b) $\mathbf{Q}_{k+1}$ is independent of $\widetilde{\Theta}_k$, $\widetilde{\Theta}_{k+1}$. Also, given $\widetilde{\Theta}_k$, $\mathbf{Y}_{k+1}$ is independent of $\widetilde{\Theta}_{k+1}, \mathcal{E}_{2,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}$

(c) For any events A, B, and a continuous random variable Y, the conditional density function $f_{Y|A}(y|A) = \mathsf{P}\{B \mid A\} f_{Y|AB}(y|AB) + \mathsf{P}\{B^c \mid A\} f_{Y|AB^c}(y|AB^c)$. Also, given $\widetilde{\Theta}_k$, $\mathbf{Y}_{k+1}$ is independent of $\mathcal{E}_{2,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}$

(d) $\mathcal{E}_{2,k}$ is $[\mathbf{I}_k, \mathbf{Q}_{k+1}]$ measurable, and hence, given $[\mathbf{I}_k, \mathbf{Q}_{k+1}]$, $\widetilde{\Theta}_k$ is independent of $\mathcal{E}_{2,k}$.

- **Case** $M_k = j > 0$, $R_k^{(j)} = 0$, $\sum_{i=1}^n R_k^{(i)} = n-1$: In this case, at time $k+1$, the decision maker receives the last sample of batch B_k, $X_{B_k}^{(j)}$ (that is successfully transmitted during slot k) and the samples of batch B_k+1, if any, that are queued in the sequencer buffer. We compute Ψ_{k+1} from Ψ_k and then we use Lemma 4.1 (using Eqn. 4.20) to compute Π_{k+1} from Ψ_{k+1}. In this case, the decision maker receives $N := \sum_{i=1}^{n} \mathbf{1}_{\{W_k^{(i)}>0\}}$ samples of batch B_k+1. Also, note that N is $\mathbf{I}_k$ measurable.

$$\begin{aligned}
\Psi_{k+1} &= \mathsf{P}\left\{\widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_{k+1}\right\} \\
&= \mathsf{P}\left\{\widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_{k}, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', \mathbf{y}]\right\} \\
&= \mathsf{P}\left\{\widetilde{\Theta}_{k} = 0, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_{k}, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', \mathbf{y}]\right\} \\
&\quad + \mathsf{P}\left\{\widetilde{\Theta}_{k} = 1, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_{k}, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', \mathbf{y}]\right\}
\end{aligned}$$

Since, we consider the case $B_{k+1} = B_k + 1$, $\Delta_{k+1} = \Delta_k + 1 - 1/r$ and hence, the state $\widetilde{\Theta}_{k+1} = \Theta_{k+1-\Delta_{k+1}} = \Theta_{k-\Delta_k+1/r}$.
Let $\mathbf{y} = [y_0, y_1, \cdots, y_n]$. Consider

$$\begin{aligned}
&\mathsf{P}\left\{\widetilde{\Theta}_k = \widetilde{\theta}, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_k, [\mathbf{Q}_{k+1}, \mathbf{Y}_{k+1}] = [\mathbf{q}', \mathbf{y}]\right\} \\
\overset{(a)}{=}\ & \frac{\mathsf{P}\left\{\widetilde{\Theta}_k = \widetilde{\theta}, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_k\right\} \cdot \mathsf{P}\left\{\mathbf{Q}_{k+1} = \mathbf{q}' \mid \widetilde{\Theta}_k = \widetilde{\theta}, \widetilde{\Theta}_{k+1} = 1, \mathcal{E}_{3,k}, \mathbf{I}_k\right\}}{\mathsf{P}(\mathbf{Q}_{k+1} = \mathbf{q}' | \mathcal{E}_{3,k}, \mathbf{I}_k) \cdot f_{\mathbf{Y}_{k+1}|\mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}}(\mathbf{y} | \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{q}')} \\
&\cdot f_{\mathbf{Y}_{k+1}|\widetilde{\Theta}_k, \widetilde{\Theta}_{k+1}, \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}}(\mathbf{y} \mid \widetilde{\theta}, 1, \mathcal{E}_{3,k}, \mathbf{q}', \mathbf{I}_k) \\
\overset{(b)}{=}\ & \frac{\mathsf{P}\left\{\widetilde{\Theta}_k = \widetilde{\theta}, \widetilde{\Theta}_{k+1} = 1 \mid \mathcal{E}_{3,k}, \mathbf{I}_k\right\} \cdot \mathsf{P}(\mathbf{Q}_{k+1} = \mathbf{q}' | \mathcal{E}_{3,k}, \mathbf{I}_k) \cdot f_{\widetilde{\theta}}(y_0) \prod_{i=1}^{n} f_1(y_i)}{\mathsf{P}(\mathbf{Q}_{k+1} = \mathbf{q}' | \mathcal{E}_{3,k}, \mathbf{I}_k) \cdot f_{\mathbf{Y}_{k+1}|\mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}}(\mathbf{y} | \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{q}')} \\
\overset{(c)}{=}\ & \frac{\mathsf{P}\left\{\widetilde{\Theta}_k = \widetilde{\theta} \mid \mathcal{E}_{3,k}, \mathbf{I}_k\right\} \cdot \mathsf{P}\left\{\widetilde{\Theta}_{k+1} = 1 \mid \widetilde{\Theta}_k = \widetilde{\theta}, \mathcal{E}_{3,k}, \mathbf{I}_k\right\} \cdot f_{\widetilde{\theta}}(y_0) \prod_{i=1}^{n} f_1(y_i)}{\sum_{\widetilde{\theta}'=0}^{1} \sum_{\widetilde{\theta}''=0}^{1} \mathsf{P}\left\{\widetilde{\Theta}_k = \widetilde{\theta}', \widetilde{\Theta}_{k+1} = \widetilde{\theta}'', \mid \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}\right\} \cdot f_{\mathbf{Y}_{k+1}|\widetilde{\Theta}_k, \widetilde{\Theta}_{k+1} \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}}(y | \widetilde{\theta}', \widetilde{\theta}'', \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{q}')}.
\end{aligned}$$

We explain the steps $(a), (b), (c)$ below.

(a) By Bayes rule, for events A, B, C, D, E, F, we have

$$\mathsf{P}\{AB \mid CDEF\} = \frac{\mathsf{P}\{AB \mid CD\} \mathsf{P}\{E \mid ABCD\} \mathsf{P}\{F \mid ABCDE\}}{\mathsf{P}\{E \mid CD\} \mathsf{P}\{F \mid CDE\}}$$

(b) $\mathbf{Q}_{k+1}$ is independent of $\widetilde{\Theta}_k$, $\widetilde{\Theta}_{k+1}$. Also, given $\widetilde{\Theta}_k$, $\mathbf{Y}_{k+1,0}$ is independent of

$\widetilde{\Theta}_{k+1}, \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}$, and given $\widetilde{\Theta}_{k+1}$, $\mathbf{Y}_{k+1,i}$ is independent of $\widetilde{\Theta}_k, \mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}$. It is to be noted that given the state of nature, the sensor measurements $Y_{k+1,0}, Y_{k+1,1}, \cdots, Y_{k+1,n}$ are conditionally independent.

(c) For any events A, B, and a continuous random variable Y, the conditional density function $f_{Y|A}(y|A) = \mathsf{P}\{B \mid A\} f_{Y|AB}(y|AB) + \mathsf{P}\{B^c \mid A\} f_{Y|AB^c}(y|AB^c)$. Also, given $\widetilde{\Theta}_k$, $\mathbf{Y}_{k+1}$ is independent of $\mathcal{E}_{3,k}, \mathbf{I}_k, \mathbf{Q}_{k+1}$

It is to be noted that the event $\mathcal{E}_{3,k}$ is $[\mathbf{I}_k, \mathbf{Q}_{k+1}]$ measurable, and hence, given $[\mathbf{I}_k, \mathbf{Q}_{k+1}]$, $\widetilde{\Theta}_k$ is independent of $\mathcal{E}_{3,k}$. Thus, in this case,

$$\Psi_{k+1} \quad = \quad \frac{(1-\Psi_k)p_r f_0(y_0)\prod_{i=1}^n f_1(y_i) + \Psi_k f_1(y_0)\prod_{i=1}^n f_1(y_i)}{(1-\Psi_k)(1-p_r)f_0(y_0)\prod_{i=1}^n f_0(y_i) + (1-\Psi_k)p_r f_0(y_0)\prod_{i=1}^n f_1(y_i) + \Psi_k f_1(y_0)\prod_{i=1}^n f_1(y_i)}.$$

Thus, using Lemma 4.1 (using Eqn. 4.20), we have

$$\begin{aligned}
\Pi_{k+1} &= \Psi_{k+1} + (1-\Psi_{k+1})(1-(1-p)^{\Delta_{k+1}}) \\
&=: \phi_\Psi(\Psi_k, \mathbf{Z}_{k+1}) + (1-\phi_\Psi(\Psi_k, \mathbf{Z}_{k+1}))\,(1-(1-p)^{\Delta_{k+1}}) \\
&= \phi_\Psi\left(\frac{\Pi_k - (1-(1-p)^{\Delta_k})}{(1-p)^{\Delta_k}}, \mathbf{Z}_{k+1}\right) \\
&\quad + \left(1 - \phi_\Psi\left(\frac{\Pi_k - (1-(1-p)^{\Delta_k})}{(1-p)^{\Delta_k}}, \mathbf{Z}_{k+1}\right)\right)(1-(1-p)^{\Delta_{k+1}}) \\
&=: \phi_\Pi\left([\mathbf{Q}_k, \Pi_k], \mathbf{Z}_{k+1}\right).
\end{aligned}$$

■

Proof of Theorem 4.3

We use the following Lemma to prove Theorem 4.3.

Lemma 4.2 *If $f : [0,1] \to \mathbb{R}$ is concave, then the function $h : [0,1] \to \mathbb{R}$ defined by*

$$h(y) = \mathsf{E}_{\phi(\mathbf{x})}\left[f\left(\frac{y \cdot \phi_2(\mathbf{x}) + (1-y)p_r \cdot \phi_1(\mathbf{x})}{y \cdot \phi_2(\mathbf{x}) + (1-y)p_r \cdot \phi_1(\mathbf{x}) + (1-y)(1-p_r) \cdot \phi_0(\mathbf{x})}\right)\right]$$

is concave, where $\phi(\mathbf{x}) = y \cdot \phi_2(\mathbf{x}) + (1-y)p_r \cdot \phi_1(\mathbf{x}) + (1-y)(1-p_r) \cdot \phi_0(\mathbf{x})$, $0 < p_r < 1$, and $\phi_0(\mathbf{x})$, $\phi_1(\mathbf{x})$, and $\phi_2(\mathbf{x})$ are pdfs on $\mathbf{X}$.

Proof Define the function $h_1 : [0,1] \to \mathbb{R}$ as

$$h_1(y;\mathbf{x}) \quad := \quad f\left(\frac{y\phi_2(\mathbf{x}) + (1-y)p_r\phi_1(\mathbf{x})}{y\phi_2(\mathbf{x}) + (1-y)p_r\phi_1(\mathbf{x}) + (1-y)(1-p_r)\phi_0(\mathbf{x})}\right)\phi(\mathbf{x}).$$

Define the operator T as follows. $\mathsf{T}(\cdot) := \int(\cdot)\, d\mathbf{x}$. Note that T is a linear operator and that $h(y) = \mathsf{T}(h_1(y;\mathbf{x}))$. Hence, it is sufficient to show that $h_1(y;\mathbf{x})$ is concave in y. Note that $f(y)$ is concave iff

$$f(y) = \inf_{(a_i,b_i)\in I}\{a_i y + b_i\}$$

where $I = \{(a,b) \in \mathbb{R}^2 : ay + b \geqslant f(y), y \in [0,1]\}$. Hence, for each $\mathbf{x}$

$$\begin{aligned}
h_1(y;\mathbf{x}) =& f\left(\frac{y\phi_2(\mathbf{x}) + (1-y)p_r\phi_1(\mathbf{x})}{y\phi_2(\mathbf{x}) + (1-y)p_r\phi_1(\mathbf{x}) + (1-y)(1-p_r)\phi_0(\mathbf{x})}\right)\phi(\mathbf{x}) \\
=& \inf_{(a_i,b_i)\in I}\left\{a_i\left(\frac{y\phi_2(\mathbf{x}) + (1-y)p_r\phi_1(\mathbf{x})}{y\phi_2(\mathbf{x}) + (1-y)p_r\phi_1(\mathbf{x}) + (1-y)(1-p_r)\phi_0(\mathbf{x})}\right) + b_i\right\}\phi(\mathbf{x}) \\
=& \inf_{(a_i,b_i)\in I}\left\{a_i\Big(y\phi_2(\mathbf{x}) + (1-y)p_r\phi_1(\mathbf{x})\Big) + b_i\phi(\mathbf{x})\right\} \\
=& \inf_{(a_i,b_i)\in I}\{\text{an affine function of } y\}
\end{aligned}$$

where we used the definition of $\phi(\mathbf{x})$ in the third equality. From the last step in the above Eqn. we infer that $h_1(y;\mathbf{x})$ is concave in y for each $\mathbf{x}$. ■

Note that in the finite H–horizon (truncated version of Eqn. 4.22), we note from *value iteration* that the cost–to–go function, for a given $\mathbf{q}$, $J_H^H([\mathbf{q},\pi]) = 1-\pi$ is concave in π. Hence, by Lemma 4.2, we see that for any given $\mathbf{q}$, the cost–to–go functions $J_{H-1}^H([\mathbf{q},\pi])$, $J_{H-2}^H([\mathbf{q},\pi])$, $\cdots$, $J_0^H([\mathbf{q},\pi])$ are concave in π.

Hence for $0 \leqslant \lambda \leqslant 1$,

$$\begin{aligned}
J^*([\mathbf{q},\pi]) &= \lim_{H\to\infty} J_0^H([\mathbf{q},\pi]) \\
J^*([\mathbf{q},\lambda\pi_1+(1-\lambda)\pi_2]) &= \lim_{H\to\infty} J_0^H\Big([\mathbf{q},\lambda\pi_1+(1-\lambda)\pi_2]\Big) \\
&\geqslant \lim_{H\to\infty} \lambda J_0^H([\mathbf{q},\pi_1]) + \lim_{H\to\infty}(1-\lambda)J_0^H([\mathbf{q},\pi_2]) \\
&= \lambda J^*([\mathbf{q},\pi_1]) + (1-\lambda)J^*([\mathbf{q},\pi_2])
\end{aligned}$$

It follows that for a given $\mathbf{q}$, $J^*([\mathbf{q},\pi])$ is concave in π. ■

Define the maps $\xi : \mathcal{Q}\times[0,1]\to\mathbb{R}_+$ and $\kappa : \mathcal{Q}\times[0,1]\to\mathbb{R}_+$, as

$$\begin{aligned}
\xi([\mathbf{q},\pi]) &:= 1-\pi \\
\kappa([\mathbf{q},\pi]) &:= c\cdot\pi + A_{J^*}([\mathbf{q},\pi])
\end{aligned}$$

where $A_{J^*}([\mathbf{q},\pi]) := \mathsf{E}\Big[J^*\left(\left[\mathbf{Q}_{k+1},\phi_\Pi(\nu_k,\mathbf{Z}_{k+1})\right]\right) \mid \nu_k = [\mathbf{q},\pi]\Big]$. Note that $\xi([\mathbf{q},1]) = 0$,

$\kappa([\mathbf{q},1]) = c$, $\xi([\mathbf{q},0]) = 1$ and

$$\begin{aligned}
\kappa([\mathbf{q},0]) &= A_{J^*}([\mathbf{q},0]) \\
&= \mathsf{E}\left[J^*\left([\mathbf{Q}_{k+1}, \phi_\Pi(\nu_k, \mathbf{Z}_{k+1})]\right) \middle| \nu_k = [\mathbf{q},0]\right] \\
&\overset{(3)}{=} \mathsf{E}\left[J^*\left([\phi_{\mathbf{Q}}(\mathbf{Q}_k, M_k), \phi_\Pi(\nu_k, \mathbf{Z}_{k+1})]\right) \middle| \nu_k = [\mathbf{q},0]\right] \\
&= \sum_{m=0}^{n} \mathsf{E}\left[J^*\left([\phi_{\mathbf{Q}}(\mathbf{Q}_k, M_k), \phi_\Pi(\nu_k, \mathbf{Z}_{k+1})]\right) \middle| M_k = m, \nu_k = [\mathbf{q},0]\right] \mathsf{P}\left\{M_k = m \middle| \nu_k = [\mathbf{q},0]\right\} \\
&= \sum_{m=0}^{n} \mathsf{E}\left[J^*\left([\phi_{\mathbf{Q}}(\mathbf{q}, m), \phi_\Pi(\nu_k, \mathbf{Z}_{k+1})]\right) \middle| M_k = m, \nu_k = [\mathbf{q},0]\right] \mathsf{P}\left\{M_k = m \middle| \nu_k = [\mathbf{q},0]\right\} \\
&\overset{(6)}{\leqslant} \sum_{m=0}^{n} J^*\left(\left[\phi_{\mathbf{Q}}(\mathbf{q}, m), \mathsf{E}\left[\phi_\Pi(\nu_k, \mathbf{Z}_{k+1}) \middle| M_k = m, \nu_k = [\mathbf{q},0]\right]\right]\right) \mathsf{P}\left\{M_k = m \middle| \nu_k = [\mathbf{q},0]\right\} \\
&= \sum_{m=0}^{n} J^*\left([\phi_{\mathbf{Q}}(\mathbf{q}, m), p\right) \mathsf{P}\left\{M_k = m \middle| \nu_k = [\mathbf{q},0]\right\} \\
&\overset{(8)}{\leqslant} \sum_{m=0}^{n} (1-p) \cdot \mathsf{P}\left\{M_k = m \middle| \nu_k = [\mathbf{q},0]\right\} \\
&= 1 - p \\
&< 1
\end{aligned}$$

where in the above derivation, we use the evolution of $\mathbf{Q}_k$ in step 3 (see Appendix), the Jensen's inequality (as for any given $\mathbf{q}$, $J^*(\mathbf{q},\pi)$ is concave in π) in step 6, and the inequality $J^*(\mathbf{q},\pi) \leqslant 1-\pi$ in step 8.

Note that $\kappa([\mathbf{q},1]) - \xi([\mathbf{q},1]) > 0$ and $\kappa([\mathbf{q},0]) - \xi([\mathbf{q},0]) < 0$. Also, for a fixed $\mathbf{q}$, the function $\kappa([\mathbf{q},\pi]) - \xi([\mathbf{q},\pi])$ is concave in π. Hence, by the *intermediate value theorem*, for a fixed $\mathbf{q}$, there exists $\gamma(\mathbf{q}) \in [0,1]$ such that $\kappa([\mathbf{q},\gamma]) = \xi([\mathbf{q},\gamma])$. This γ is unique as $\kappa([\mathbf{q},\pi]) = \xi([\mathbf{q},\pi])$ for at most two values of π. If in the interval $[0,1]$, there are two distinct values of π for which $\kappa([\mathbf{q},\pi]) = \xi([\mathbf{q},\pi])$, then the signs of $\kappa([\mathbf{q},0]) - \xi([\mathbf{q},0])$ and $\kappa([\mathbf{q},1]) - \xi([\mathbf{q},1])$ should be the same. Hence, the optimal stopping rule is given by

$$\tau^* = \inf\left\{k : \Pi_k \geqslant \gamma(\mathbf{Q}_k)\right\}$$

where the threshold $\gamma(\mathbf{q})$ is given by $c \cdot \gamma(\mathbf{q}) + A_{J^*}([\mathbf{q}, \gamma(\mathbf{q})]) = 1 - \gamma(\mathbf{q})$. ■

Chapter 5

Optimal Transient–Change Detection

5.1 Introduction

In the previous chapters, we are concerned with the detection of a persistent change (i.e., once the change occurs, the system stays in the in–change state for ever). However, in some applications, the event that causes the change disappears after a finite time, and the system goes to an out–of–change state which, based on the observations, is hard to distinguish from the pre–change state. This change model, which we call *transient change*, is applicable in intrusion detection applications where an intruder appears at a random time, stays for a random length of time in the system, and then leaves the region of interest (ROI). The goal is to detect whether a change has occurred, as early as possible, even after the change has disappeared at the time of detection. We study this problem of transient change detection in this chapter.

In the transient change model, the distribution of the observations after the change disappears (i.e., in the out–of–change state) is the same as that when the change has not happened yet (i.e., the pre–change state). Thus making a decision about the state of the system as being pre–change or post–change (which includes the in–change and the out–of–change state in the transient change model), based on the observations, appears to be more challenging than in the case of persistent change.

5.1.1 Summary of Contributions

We summarise the contributions of this chapter below:

1. We provide a model for the transient change and formulate the optimal transient change detection problem.

2. We obtain the following procedures for detecting a transient change:

   (i) MinD (**Min**imum Detection **D**elay) which minimises the mean detection delay when the probability of false alarm is limited to α

   (ii) A–MinD (**A**symptotic – **Min**imum Detection **D**elay) which is obtained as a limit of of the MinD procedure when the mean time until the occurrence of change goes to ∞ (i.e., for a rare event)

   (iii) MaxP (**Max**imum **P**robability of change) which maximises the probability of stopping when the change is present (which we call the probability of detection) when the probability of false alarm is limited to α.

5.1.2 Discussion of the Related Literature

Our work differs from all the previous work discussed in Section 1.1 in the following ways.

1. In all the previous work, the change is persistent, i.e., once the system goes from the pre–change state to the in–change state, it remains in the in–change state for ever. But, we consider a sequential change detection problem where the change is transient, i.e., the system stays in the in–change state only for a finite amount of time.

2. We also pose the problem of maximising the probability of detection (defined in Section 5.2.3) subject to a false alarm constraint. In the classical change detection problem, since the probability of detection is $1 - \mathsf{P}_{\mathsf{FA}}$ this problem does not arise.

In a recent technical report, Polunchenko and Tartakovsky studied the non–Bayesian transient change detection problem, and studied the supremum detection delay performance of CUSUM under a false alarm constraint, [Polunchenko and Tartakovsky, 2009].

5.1.3 Outline of the Chapter

The rest of this chapter is organised as follows. In Section 5.2, we formulate the *transient change detection* problem. In Section 5.3, we obtain a sequential change detection procedure MinD that achieves the minimum mean detection delay ADD subject to the false alarm constraint, $\mathsf{P}_{\mathsf{FA}} \leqslant \alpha$. In Section 5.4, we discuss the asymptotic behaviour of MinD, which we call $\mathsf{A} - \mathsf{MinD}$, as the probability of occurrence of the change goes to zero (i.e., as the average time for the change to occur goes to ∞). In Section 5.5, we obtain a sequential change detection procedure MaxP which achieves the maximum *probability of detection* P_{D} under the constraint, $\mathsf{P}_{\mathsf{FA}} \leqslant \alpha$. We provide numerical results in Section 5.6. Finally, we summarise in Section 5.7.

5.2 Problem Formulation

We consider a discrete time system in which time is measured in slots and the slots are indexed by non–negative integers. We assume that all nodes are time synchronised. Also, we assume that the length of a slot is unity, and slot k refers to the time interval $[k, k+1)$.

5.2.1 Change Model

A change occurs at a random time $T \in \mathbb{Z}_+$ and disappears at a random time $E \in \mathbb{Z}_+$, such that $E > T$. This change–model is motivated by the behaviour of physical intrusion (say, by a human) in a region under surveillance. Let Θ_k represent the state of the system at time k. We say that Θ_k is 0 before the change occurs (pre–change), 1 when the change is present in the system (in–change), and 2 after the change disappears (out–of–change) (see Figure 5.1). Thus, with the state space of the system being defined as $\boldsymbol{\Theta} = \{0, 1, 2\}$,

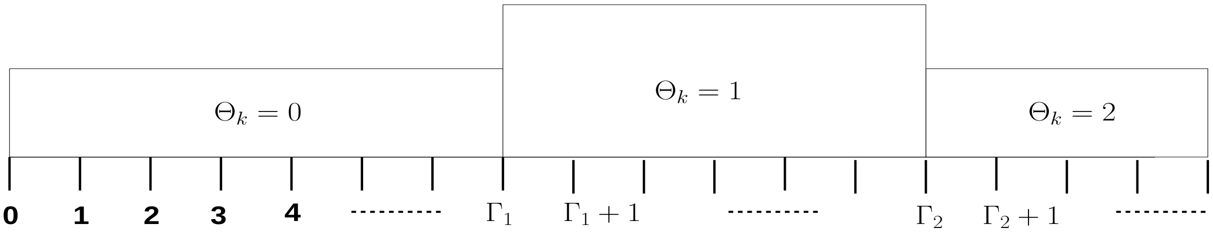


Figure 5.1: **State evolution**. At any time k, state $\Theta_k \in \{0, 1, 2\}$, where 0 represents pre–change, 1 represents in–change and 2 represents out–of–change. Note that at time T the state changes from 0 to 1, and at time E the state changes from 1 to 2.

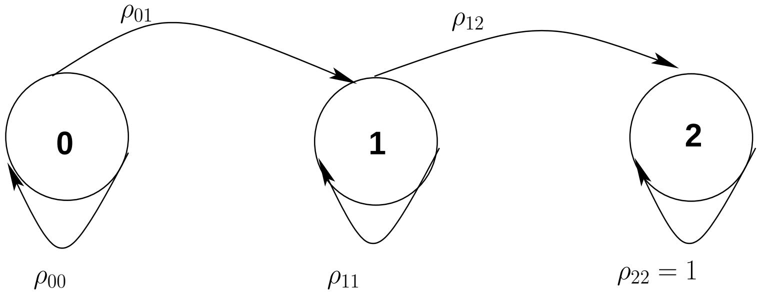


Figure 5.2: **State transition diagram**. Recall that State 0 represents pre–change state, 1 represents the in–change and 2 represents the out–of–change state. Note that state $\theta - 1$ can not be visited from state θ and that the state '2' is an absorbing state.

$$\Theta_k = \begin{cases} 0, & \text{for } k < T \\ 1, & \text{for } T \leqslant k < E \\ 2, & \text{for } k \geqslant E. \end{cases}$$

We assume that the evolution of the state process $\{\Theta_k\}$ is Markovian (see Figure 5.2) and that the transition probabilities are given by

$$\mathsf{P}\{\Theta_k = j \mid \Theta_{k-1} = i\} = \rho_{ij}, \quad i, j \in \{0, 1, 2\}, \tag{5.1}$$

with the following conditions: $\rho_{02} = \rho_{10} = \rho_{20} = \rho_{21} = 0$, and the other ρ_{ij}s are strictly positive. With these conditions, it is easy to see that $\rho_{00} + \rho_{01} = 1$, $\rho_{11} + \rho_{12} = 1$, and $\rho_{22} = 1$, i.e., the system can not go to the out–of–change state from the pre–change state directly (as $\rho_{02} = 0$), and it can not go to the pre–change state from the in–change (as $\rho_{10} = 0$) or the out–of–change state (as $\rho_{20} = 0$). Also, the out–of–change state is an absorbing state (as $\rho_{22} = 1$). Thus, given $\Theta_0 = 0$, the distributions of T and $E - T$ are geometric, and are given by

$$\begin{aligned}
\mathsf{P}\{T = k \mid \Theta_0 = 0\} &= \rho_{00}^{k-1}\rho_{01} = (1-\rho_{01})^{k-1}\rho_{01} \\
\mathsf{P}\{E - T = k \mid \Theta_0 = 0\} &= \rho_{11}^{k-1}\rho_{12} = (1-\rho_{12})^{k-1}\rho_{12}.
\end{aligned}$$

Let the distribution of Θ_0 be given by

$$\mathsf{P}\{\Theta_0 = \theta\} = \begin{cases} 1-\rho, & \text{if } \theta = 0 \\ \rho, & \text{if } \theta = 1 \\ 0, & \text{if } \theta = 2, \end{cases} \tag{5.2}$$

for some $0 \leqslant \rho \leqslant 1$. We note that the transient change model reduces to the classical change model when $\rho_{12} = 0$.

5.2.2 Observation Model

Observations are obtained sequentially starting from time $k = 1$ onwards. Let the random variable $\mathbf{X}_k$ denote the observation at time k. The distribution of $\mathbf{X}_k$ in the pre–change and the out–of–change state (i.e., when $1 \leqslant k < T$ or $k \geqslant E$) is given by $F_0(\cdot)$, and that in the in–change state (i.e., when $T \leqslant k < E$) is given by $F_1(\cdot)$, where $F_1(\cdot) \neq F_0(\cdot)$. We assume that the corresponding pdfs f_0 and $f_1 \neq f_0$ exist (and also that the measure described by f_0 is absolutely continuous with respect to that described by f_1). Conditioned on the state of the nature (i.e., on T and E), the observations are independent across time.

The model, we consider here, is an extension of of what is considered in the classical work, [Shiryaev, 1978], [Page, 1954] and [Veeravalli, 2001]), and that the special case of $E = \infty$ corresponds to the classical change detection problem. Note that at any time k, *the state Θ_k is not observed directly and that it is observed partially only through the observation* $\mathbf{X}_k$. Also, note that the states "0" and "2" are indistinguishable from the observations.

At every integer–valued time k, we observe $\mathbf{X}_k$ and hence, we have a collection of observations $\mathbf{X}_{[1:k]}$. Based on the observations $\mathbf{X}_{[1:k]}$, ρ, ρ_{ij}s, and the pre–change and the in–change pdfs $f_0(\cdot)$ and $f_1(\cdot)$, the decision maker has to make a decision on whether the change has occurred (denoted by action "1") or to continue observing (denoted by action "0"). Let A_k be the decision made by the fusion centre at time k and let $\mathbf{I}_k := [\mathbf{X}_{[1:k]}, A_{[0:k-1]}]$ be the information available to the decision maker (fusion centre) at time k. Let τ be a stopping time with respect to the information sequence $\mathbf{I}_1, \mathbf{I}_2, \cdots$.

We now define the various performance measures of sequential change detection procedures in Section 5.2.3 and formulate the transient change detection problems.

5.2.3 Definitions

In the definitions below, we use the terms change detection procedure and stopping time interchangeably, since the stopping time defines a sequential change detection procedure.

Definition 5.1 Probability of False Alarm (P_{FA}) for a change detection procedure τ is defined as the probability that the change detection procedure τ raises an alarm in state 0, i.e.,

$$\mathsf{P}_{\mathsf{FA}}(\tau) \ := \ \mathsf{P}\left\{\tau < T\right\} \ = \ \mathsf{P}\left\{\Theta_\tau = 0\right\}$$

Definition 5.2 Probability of Detection (P_{D}) for a change detection procedure τ is defined as the probability that the change detection procedure τ raises an alarm in state 1, i.e.,

$$\mathsf{P}_{\mathsf{D}}(\tau) \ := \ \mathsf{P}\left\{T \leqslant \tau < E\right\} \ = \ \mathsf{P}\left\{\Theta_\tau = 1\right\}$$

Definition 5.3 **Probability of Miss ($\mathsf{P_M}$)** for a change detection procedure τ is defined as the probability that the change detection procedure τ raises an alarm in state 2, i.e.,

$$\mathsf{P_M}(\tau) \;:=\; \mathsf{P}\left\{\tau \geqslant E\right\} \;=\; \mathsf{P}\left\{\Theta_\tau = 2\right\}$$

Note that for any τ, $\mathsf{P_{FA}}(\tau) + \mathsf{P_D}(\tau) + \mathsf{P_M}(\tau) = 1$. The definition of the mean detection delay (ADD) is the same as what we have defined in Chapter 2, i.e.,

$$\mathsf{ADD}(\tau) := \mathsf{E}\left[(\tau - T)^+\right]$$

A smaller ADD will result in a larger $\mathsf{P_{FA}}$. But, we will constrain the $\mathsf{P_{FA}}$.

We now formulate the following transient change detection problems

P5.1: We are interested in detecting the change as soon as it occurs, and thus a natural choice is the change detection procedure that raises an alarm with the least mean detection delay ADD subject to a false alarm constraint. We thus formulate the problem as

$$\begin{array}{ll} \text{minimise} & \mathsf{ADD}(\tau) \\ \text{subject to} & \mathsf{P_{FA}}(\tau) \leqslant \alpha. \end{array}$$

P5.2: Since, the change disappears at time E, we are interested in obtaining a change detection procedure that stops and declares an alarm before E. Thus, we formulate the change detection problem as one that maximises the probability of detection $\mathsf{P_D}$ subject to a false alarm constraint. We thus have

$$\begin{array}{ll} \text{maximise} & \mathsf{P_D}(\tau) \\ \text{subject to} & \mathsf{P_{FA}}(\tau) \leqslant \alpha. \end{array}$$

We solve the problem **P5.1** in Section 5.3 and the problem **P5.2** in Section 5.5.

5.3 Minimum Detection Delay Policy (MinD)

In this section, we consider the problem **P5.1** defined in Eqn. 5.3. Recall that at time k, $\Theta_k \in \{0,1\}$ is the state of nature, and $A_k \in \mathcal{A} = \{0,1\}$ is the decision (or control or action) chosen by the decision maker after having observed $\mathbf{X}_k$. We note that 0 represents "take another sample" and 1 represents the action "stop and declare change." We recall that $\mathbf{I}_k = \left[\mathbf{X}_{[1:k]}, A_{[0:k-1]}\right]$ is the *information* vector that is available to the decision maker, at the beginning of time slot k and that τ is a stopping time with respect to the sequence $\mathbf{I}_1, \mathbf{I}_2, \cdots$. At any time k, the cost function $c_k : \{0,1\} \times \mathcal{A} \to \mathbb{R}_+$ for a state–action pair (θ, a) is defined as follows. For any time before stopping, i.e., for $k \leqslant \tau$,

$$c_k(\theta, a) \quad := \quad \begin{cases} 0, & \text{if } \theta = 0, a = 0 \\ 1, & \text{if } \theta = 1, a = 0 \\ 1, & \text{if } \theta = 2, a = 0 \\ \lambda_f, & \text{if } \theta = 0, a = 1 \\ 0, & \text{if } \theta = 1, a = 1 \\ 0, & \text{if } \theta = 2, a = 1 \end{cases} \tag{5.3}$$

and for $k > \tau$, $c_k(\cdot,\cdot) := 0$. Thus, for $k \leqslant \tau$, we have

$$c_k(\Theta_k, A_k) \quad := \quad \lambda_f \mathbf{1}_{\{\Theta_k=0\}} \mathbf{1}_{\{A_k=1\}} + \mathbf{1}_{\{\Theta_k \neq 0\}} \mathbf{1}_{\{A_k=0\}} \tag{5.4}$$

We are interested in obtaining an optimum stopping time τ that minimises the mean detection delay subject to a constraint on the probability of false alarm.

$$\begin{aligned} \min_{\tau} \quad & \mathsf{E}\left[(\tau - T)^+\right] \\ \text{subject to} \quad & \mathsf{P}\left\{\tau < T\right\} \leqslant \alpha. \end{aligned} \tag{5.5}$$

Let λ_f be the cost of false alarm. We are interested in obtaining a stopping time τ^{MinD}

that minimises the expected cost (Bayesian risk), i.e.,

$$\begin{aligned}
R(\tau^{\mathsf{MinD}}) &:= \min_{\tau} \ \mathsf{E}\big[\lambda_f \cdot \mathbf{1}_{\{\tau<T\}} + (\tau - T)^+\big] \\
&= \min_{\tau} \ \mathsf{E}\big[\lambda_f \cdot \mathbf{1}_{\{\Theta_\tau=0\}} + (\tau - T)^+\big] \\
&= \min_{\tau} \ \mathsf{E}\left[\lambda_f \cdot \mathbf{1}_{\{\Theta_\tau=0\}} + \sum_{k=0}^{\tau-1} \mathbf{1}_{\{\Theta_k\neq 0\}}\right] \\
&= \min_{\tau} \ \mathsf{E}\left[c_\tau(\Theta_\tau, 1) + \sum_{k=0}^{\tau-1} c_k(\Theta_k, 0)\right] \\
&= \min_{\tau} \ \mathsf{E}\left[\sum_{k=0}^{\tau} c_k(\Theta_k, A_k)\right] \\
&= \min_{\tau} \ \mathsf{E}\left[\sum_{k=0}^{\infty} c_k(\Theta_k, A_k)\right] \\
&= \min_{\tau} \ \sum_{k=0}^{\infty} \mathsf{E}[c_k(\Theta_k, A_k)] \ \text{(by monotone convergence theorem)} \qquad (5.6)
\end{aligned}$$

Note that λ_f is a Lagrange multiplier and is chosen such that the false alarm constraint is satisfied with equality, i.e., $\mathsf{P}_{\mathsf{FA}}(\tau^{\mathsf{MinD}}) = \alpha$ (see [Shiryaev, 1978]). We note that for every stopping time τ, there exists a policy $\mu = (\mu_1, \mu_2, \cdots)$ such that for any k, when $\tau = k$, $A_{k'} = \mu_{k'}(\mathbf{I}_{k'}) = 0$ for all $k' < k$ and $A_{k'} = \mu_{k'}(\mathbf{I}_{k'}) = 1$ for all $k' \geqslant k$. Hence, the optimal Bayesian cost given by Eqn. 5.6 becomes

$$\begin{aligned}
R(\tau^{\mathsf{MinD}}) = \min_{\tau} \sum_{k=0}^{\infty} \mathsf{E}[c_k(\Theta_k, A_k)] &= \min_{\mu} \sum_{k=0}^{\infty} \mathsf{E}[c_k(\Theta_k, A_k)] \\
&= \min_{\mu} \sum_{k=0}^{\infty} \mathsf{E}[\mathsf{E}[c_k(\Theta_k, A_k) \mid \mathbf{I}_k]] \\
&= \min_{\mu} \sum_{k=0}^{\infty} \mathsf{E}[\mathsf{E}[c_k(\Theta_k, \mu_k(\mathbf{I}_k)) \mid \mathbf{I}_k]] \\
&= \min_{\mu} \ \mathsf{E}\left[\sum_{k=0}^{\infty} \mathsf{E}[c_k(\Theta_k, \mu_k(\mathbf{I}_k)) \mid \mathbf{I}_k]\right] \qquad (5.7)
\end{aligned}$$

where the last step follows from monotone convergence theorem. For each $\theta = 0, 1$, and 2, we define the posterior probability of state θ, $\Pi_{k,\theta} := \mathsf{E}\left[\mathbf{1}_{\{\Theta_k=\theta\}} \,\middle|\, \mathbf{I}_k\right]$. From Eqn. 5.4, we see that

$$c_k(\Theta_k, \mu_k(\mathbf{I}_k)) \;=\; \lambda_f \mathbf{1}_{\{\Theta_k=0\}} \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=1\}} + \mathbf{1}_{\{\Theta_k\neq 0\}} \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=0\}}$$

and hence,

$$\begin{aligned}
\mathsf{E}[c_k(\Theta_k, \mu_k(\mathbf{I}_k)) \mid \mathbf{I}_k] \;&=\; \mathsf{E}\big[\lambda_f \mathbf{1}_{\{\Theta_k=0\}} \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=1\}} + \mathbf{1}_{\{\Theta_k\neq 0\}} \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=0\}} \mid \mathbf{I}_k\big] \\
&=\; \lambda_f \cdot \mathsf{E}\big[\mathbf{1}_{\{\Theta_k=0\}} \mid \mathbf{I}_k\big] \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=1\}} + \mathsf{E}\big[\mathbf{1}_{\{\Theta_k\neq 0\}} \mid \mathbf{I}_k\big] \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=0\}} \\
&=\; \lambda_f \cdot \Pi_{k,0} \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=1\}} + (1 - \Pi_{k,0}) \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=0\}}.
\end{aligned}$$

From the above equation, it is easy to see that at time k, a sufficient statistic ([Bertsekas, 2005]) is the probability vector $\mathbf{\Pi}_k = [\Pi_{k,0}, \Pi_{k,1}, \Pi_{k,2}]$. We call $\mathbf{\Pi}_k$ as the information state at time k. We also have a special information state called the terminal (or absorbing) state t, to which the system enters when an alarm is raised. Thus, the state space of the system is $\mathcal{S} = \mathcal{P} \cup \{\mathsf{t}\}$, where $\mathcal{P}$ is a 2–dimensional simplex defined as $\mathcal{P} := \{(p_0, p_1, p_2) \in [0,1]^3 : p_0 + p_1 + p_2 = 1\}$. We define the cost function $\widetilde{c} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$ as

$$\widetilde{c}(s, a) \;=\; \begin{cases} \lambda_f \cdot p_0, & \text{if } s = [p_0, p_1, p_2] \in \mathcal{P}, a = 1 \\ 1 - p_0, & \text{if } s = [p_0, p_1, p_2] \in \mathcal{P}, a = 0 \\ 0, & \text{if } s = \mathsf{t}. \end{cases} \tag{5.8}$$

Since $\mathbf{\Pi}_k$ is a sufficient statistic for $\mathbf{I}_k$, for any policy μ_k there exists a corresponding policy $\widetilde{\mu}_k$ such that $\widetilde{\mu}_k(\mathbf{\Pi}_k) = \mu_k(\mathbf{I}_k)$, and hence, Eqn. 5.6 can be written as

$$\begin{aligned}
R(\tau^{\mathsf{MinD}}) \;&=\; \min_{\mu} \; \mathsf{E}\left[\sum_{k=0}^{\infty} \mathsf{E}[c_k(\Theta_k, \mu_k(\mathbf{I}_k)) \mid \mathbf{I}_k]\right] \\
&=\; \min_{\widetilde{\mu}} \; \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{c}(\mathbf{\Pi}_k, \widetilde{\mu}_k(\mathbf{\Pi}_k))\right]
\end{aligned} \tag{5.9}$$

We now show that given the previous state $\mathbf{\Pi}_{k-1}$, the previous action A_{k-1}, and the current observation $\mathbf{X}_k$, the current state $\mathbf{\Pi}_k$ can be computed as $\mathbf{\Pi}_k = \mathbf{\Phi}\left(\mathbf{\Pi}_{k-1}, A_{k-1}, \mathbf{X}_k\right)$, where

$$\mathbf{\Phi}(s,a,\mathbf{x}) \;:=\; \begin{cases} \mathsf{t}, & \text{if } s = \mathsf{t} \\ \mathsf{t}, & \text{if } a = 1 \\ \Big[\phi_0(s,\mathbf{x}), \phi_1(s,\mathbf{x}), \phi_2(s,\mathbf{x})\Big], & \text{otherwise,} \end{cases}$$

where for any $\mathbf{p} = [p_0, p_1, p_2] \in \mathcal{P}$, the functions $\phi_\theta(\mathbf{p},\mathbf{x})$, $\theta = 0, 1, 2$ are defined as follows:

$$\begin{aligned}
\phi_0(\mathbf{p},\mathbf{x}) &:= \frac{p_0\rho_{00} f_0(\mathbf{x})}{(p_0\rho_{00} + p_1\rho_{12} + p_2)\, f_0(\mathbf{x}) + (p_0\rho_{01} + p_1\rho_{11})\, f_1(\mathbf{x})}, \\
\phi_1(\mathbf{p},\mathbf{x}) &:= \frac{(p_0\rho_{01} + p_1\rho_{11})\, f_1(\mathbf{x})}{(p_0\rho_{00} + p_1\rho_{12} + p_2)\, f_0(\mathbf{x}) + (p_0\rho_{01} + p_1\rho_{11})\, f_1(\mathbf{x})}, \text{ and} \\
\phi_2(\mathbf{p},\mathbf{x}) &:= \frac{(p_1\rho_{12} + p_2)\, f_0(\mathbf{x})}{(p_0\rho_{00} + p_1\rho_{12} + p_2)\, f_0(\mathbf{x}) + (p_0\rho_{01} + p_1\rho_{11})\, f_1(\mathbf{x})}.
\end{aligned}$$

It is to be noted that though the one stage cost function $\widetilde{c}$, at time k, depends only on $\Pi_{k,0}$, the transition requires the vector of a posteriori probabilities $\mathbf{\Pi}_k$. Thus, the information state process $\{\mathbf{\Pi}_k\}$ is a controlled Markov process. Thus, the tuple $(\mathcal{S}, \mathcal{A}, \mathbf{\Phi}(\cdot,\cdot,\cdot), \widetilde{c})$ defines a Markov decision process (MDP). We thus solve the problem defined in Eqn. 5.6 as follows. Since the one stage cost function and the transition kernel are time–invariant, it is sufficient to look for stationary policies. Let $\widetilde{\mu} : \mathcal{S} \to \mathcal{A}$ be a stationary policy. Then the total cost on using the policy $\widetilde{\mu}$ is given by

$$J_{\widetilde{\mu}}(\boldsymbol{\pi}) \;=\; \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{c}(\mathbf{\Pi}_k, \widetilde{\mu}(\mathbf{\Pi}_k)) \mid \mathbf{\Pi}_0 = \boldsymbol{\pi}\right]. \qquad (5.10)$$

Let $\mu^* : \mathcal{S} \to \mathcal{A}$ be an optimal stationary policy. The optimal total cost is then given by

$$\begin{aligned}
J^*(\boldsymbol{\pi}) &= \min_{\widetilde{\mu}} \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{c}(\mathbf{\Pi}_k, \widetilde{\mu}(\mathbf{\Pi}_k)) \mid \mathbf{\Pi}_0 = \boldsymbol{\pi}\right] \\
&= \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{c}(\mathbf{\Pi}_k, \mu^*(\mathbf{\Pi}_k)) \mid \mathbf{\Pi}_0 = \boldsymbol{\pi}\right] \qquad (5.11)
\end{aligned}$$

The DP that solves Eqn. 5.11 is given by Bellman's equation as follows

$$\begin{aligned} J^*(\boldsymbol{\pi}) &= \min\left\{\widetilde{c}(\boldsymbol{\pi},1), \widetilde{c}(\boldsymbol{\pi},0) + \underbrace{\mathsf{E}\left[J^*\left(\boldsymbol{\Phi}(\boldsymbol{\pi},0,\mathbf{X})\right)\right]}_{A_{J^*}(\boldsymbol{\pi})}\right\} \\ &=: \min\left\{\lambda_f \cdot \pi_0, \underbrace{(1-\pi_0) + A_{J^*}(\boldsymbol{\pi})}_{B_{J^*}(\boldsymbol{\pi})}\right\} \end{aligned} \tag{5.12}$$

We see from Eqn. 5.12 that the optimal stopping time τ^{MinD} is given by

$$\tau^{\mathsf{MinD}} = \inf\left\{k \geqslant 0 : \lambda_f \cdot \Pi_{k,0} \leqslant (1-\Pi_{k,0}) + A_{J^*}(\boldsymbol{\Pi}_k)\right\}. \tag{5.13}$$

We now show the structural properties of J^* in the following theorem.

Theorem 5.1 *$J^*(\boldsymbol{\pi})$ is concave in $\boldsymbol{\pi}$.*

We provide the detection delay analysis as $\alpha \to 0$, in the following theorem.

Theorem 5.2 *For any stopping rule τ with $\mathsf{P}_{\mathsf{FA}}(\tau) \leqslant \alpha$, as $\alpha \to 0$,*

$$\mathsf{E}_{k_1,k_2}\left[\tau - k_1 \mid \tau \geqslant k_1\right] \geqslant \frac{|\ln(\alpha)|}{|\ln(\rho_{00})|}\left(1 + o(1)\right), \tag{5.14}$$

where the $o(1)$ term goes to 0 as $\alpha \to 0$. As the P_{FA} constraint $\alpha \to 0$, the number of samples required for detection becomes large. However, the system remains in the in–change state only for a finite amount of time, and hence, stopping occurs due to the samples in the out–of–change state.

From Theorem 5.2, it is clear that the observations are not required for optimal detection. Let us now consider the following ad hoc detection rule which stops at time τ^{adhoc} when the P_{FA} constraint is met based on the distribution of the change time T.

$$\begin{aligned} \tau^{\mathsf{adhoc}} &:= \inf\left\{k : \mathsf{P}\left\{T > k\right\} \leqslant \alpha\right\} \\ &= \inf\left\{k : \sum_{k'=k+1}^{\infty} \rho_{00}^{k'-1}\rho_{01} \leqslant \alpha\right\}. \end{aligned}$$

Note that

$$\begin{aligned} \sum_{k=\tau^{\mathsf{adhoc}}+1}^{\infty} \rho_{00}^{k-1}\rho_{01} &\leqslant \alpha \\ \implies \qquad \rho_{00}^{\tau^{\mathsf{adhoc}}} &\leqslant \alpha \\ \implies \qquad \tau^{\mathsf{adhoc}} \ln(\rho_{00}) &\leqslant \ln(\alpha) \\ \implies \qquad \tau^{\mathsf{adhoc}} &\geqslant \frac{\ln(\alpha)}{\ln(\rho_{00})} \end{aligned}$$

Thus, $\mathsf{E}_{k_1,k_2}\left[\tau^{\mathsf{adhoc}} - k_1 \mid \tau^{\mathsf{adhoc}} \geqslant k_1\right] \geqslant \frac{|\ln(\alpha)|}{|\ln(\rho_{00})|} - k_1$. Also, note that $\tau^{\mathsf{adhoc}} \in \Delta(\alpha)$, and as $\alpha \to 0$, the mean detection delay of τ^{adhoc} is no worse than that given by Theorem 5.2. Hence, τ^{adhoc} is asymptotically delay optimal.

5.4 Asymptotic Minimal Detection Delay Policy (A–MinD)

In the previous section, we have obtained MinD, an optimal transient change detection procedure whose mean detection delay is the smallest in the class of policies having $\mathsf{P}_{\mathsf{FA}} \leqslant \alpha$. From Eqn. 5.13, we see that at any time k, MinD requires the computation of $A_{J^*}(\mathbf{\Pi}_k)$. We would be interested in a simpler detection rule based on a constant threshold (i.e., the threshold Γ is a constant). In this section, we obtain a change detection procedure $\mathsf{A} - \mathsf{MinD}$ which is obtained as a limiting policy of MinD as the average time for the change to occur goes to ∞, i.e., when $\rho_{01} \to 0$. Also, we show that the stopping time of $\mathsf{A} - \mathsf{MinD}$ is a simple (constant) threshold rule.

For each time k, we define the following one–to–one transformation (see [Veeravalli, 2001], [Raghavan and Veeravalli, 2010])

$$Q_{k,\theta} \ := \ \frac{\Pi_{k,\theta}}{\rho_{01}\Pi_{k,0}}, \ \theta = 0, 1, 2. \tag{5.15}$$

Note that $Q_{k,0} := \frac{1}{\rho_{01}}$ and $Q_{k,0} + Q_{k,1} + Q_{k,2} = \frac{1}{\rho_{01}\Pi_{k,0}}$. Hence, the inverse transformation for $\theta = 1, 2$, is given by

$$\begin{aligned}\Pi_{k,\theta} &= \frac{Q_{k,\theta}}{Q_{k,0} + Q_{k,1} + Q_{k,2}}, \\ &= \frac{Q_{k,\theta}/Q_{k,0}}{(Q_{k,0} + Q_{k,1} + Q_{k,2})/Q_{k,0}}, \\ &= \frac{\rho_{01} Q_{k,\theta}}{1 + \rho_{01}(Q_{k,1} + Q_{k,2})}. \end{aligned} \tag{5.16}$$

The last step follows as $Q_{k,0} := 1/\rho_{01}$. It can be showed that the transformed statistic $Q_{k,\theta}$ can also be computed recursively as follows.

Theorem 5.3 *The statistic $Q_{k,\theta}$ for $\theta = 1, 2$, can be computed in a recursive manner as*

$$Q_{k,\theta} = \frac{L_{k,\theta}}{\rho_{00}} \left(\sum_{\theta'=0}^{2} Q_{k-1,\theta'} \rho_{\theta'\theta} \right) \tag{5.17}$$

where $L_{k,\theta}$ is the likelihood–ratio between $f_\theta(\mathbf{X}_k)$ and $f_0(\mathbf{X}_k)$ and is given by $L_{k,\theta} = \frac{f_\theta(\mathbf{X}_k)}{f_0(\mathbf{X}_k)}$, and $f_2(\mathbf{X}_k) = f_0(\mathbf{X}_k)$.

Using this transformation, we can show that the stopping rule MinD given in Eqn. 5.13 can be written as

$$\tau^{\mathsf{MinD}} = \inf \left\{ k : Q_{k,1} + Q_{k,2} \geqslant \frac{1 - A_{J^*}(\mathbf{Q}_k)}{\rho_{01}(\lambda_f + A_{J^*}(\mathbf{Q}_k))} \right\} \tag{5.18}$$

We now consider the asymptotic behaviour of the procedure τ^{MinD} as the probability of occurrence of change goes to zero.

Theorem 5.4 [**[Raghavan and Veeravalli, 2010]**] *In the case of rare events, i.e., as $\rho_{01} \to 0$, the optimal stopping rule τ^{MinD} converges to the following simple threshold rule*

$$\tau^{\mathsf{A-MinD}} = \inf \{ k : Q_{k,1} + Q_{k,2} \geqslant \Gamma \}$$

where the threshold Γ is chosen such that the $\mathsf{P_{FA}}$ constraint is met.

5.5 Maximum Probability of Detection Policy (MaxP)

In this section, we consider the problem **P5.2** defined in Eqn. 5.3. Recall that at time k, $\Theta_k \in \{0,1\}$ is the state of nature and $A_k \in \mathcal{A} = \{0,1\}$ is the decision (or control or action) chosen by the decision maker after having observed $\mathbf{X}_k$. Let τ be a stopping time with respect to the sequence $\mathbf{I}_1, \mathbf{I}_2, \cdots$, where we recall that $\mathbf{I}_k = \left[\mathbf{X}_{[1:k]}, A_{[0:k-1]}\right]$ is the information available to the decision maker at the beginning of slot k. For a state–action pair (θ, a), at any time k, we define the gain function $g_k : \{0,1\} \times \mathcal{A} \to \mathbb{R}_+$ as follows. For time $k \leqslant \tau$,

$$g_k(\theta, a) \quad := \quad \begin{cases} 0, & \text{if } \theta = 0, a = 0 \\ 0, & \text{if } \theta = 1, a = 0 \\ 0, & \text{if } \theta = 2, a = 0 \\ -\lambda_f, & \text{if } \theta = 0, a = 1 \\ 1, & \text{if } \theta = 1, a = 1 \\ 0, & \text{if } \theta = 2, a = 1 \end{cases} \tag{5.19}$$

and for $k > \tau$, $g_k(\cdot, \cdot) := 0$. Note that the gain is $-\lambda_f$ if the stopping happens in state 0 and is 1 if the stopping happens in state 1. Thus, for any $k \leqslant \tau$, we have

$$g_k(\Theta_k, A_k) \quad := \quad \left[-\lambda_f \mathbf{1}_{\{\Theta_k = 0\}} + \mathbf{1}_{\{\Theta_k = 1\}}\right] \mathbf{1}_{\{A_k = 1\}}. \tag{5.20}$$

We are interested in obtaining an optimum stopping time τ that maximises the probability of detection subject to a constraint on the probability of false alarm.

$$\begin{aligned} \max_{\tau} \qquad & \mathsf{P}\left\{T \leqslant \tau < E\right\} \\ \text{subject to} \quad & \mathsf{P}\left\{\tau < T\right\} \leqslant \alpha. \end{aligned} \tag{5.21}$$

Recall that λ_f is the cost of false alarm. We are interested in obtaining a stopping time

τ^{MaxP} that maximises the expected gain (Bayesian reward), i.e.,

$$\begin{aligned}
G(\tau^{\mathsf{MaxP}}) &:= \max_{\tau} \ \mathsf{E}\big[-\lambda_f \cdot \mathbf{1}_{\{\tau<T\}} + \mathbf{1}_{\{T\leqslant\tau<E\}}\big] \\
&= \max_{\tau} \ \mathsf{E}\big[-\lambda_f \cdot \mathbf{1}_{\{\Theta_\tau=0\}} + \mathbf{1}_{\{\Theta_\tau=1\}}\big] \\
&= \max_{\tau} \ \mathsf{E}[g_\tau(\Theta_\tau, A_\tau)] \\
&= \max_{\tau} \ \mathsf{E}\left[\sum_{k=0}^{\tau} g_k(\Theta_k, A_k)\right] \\
&= \max_{\tau} \ \mathsf{E}\left[\sum_{k=0}^{\infty} g_k(\Theta_k, A_k)\right] \\
&= \max_{\tau} \ \sum_{k=0}^{\infty} \mathsf{E}[g_k(\Theta_k, A_k)] \quad \text{(by monotone convergence theorem)}
\end{aligned} \tag{5.22}$$

Note that λ_f is a Lagrange multiplier and is chosen such that the false alarm constraint is satisfied with equality, i.e., $\mathsf{P}_{\mathsf{FA}}(\tau^{\mathsf{MinD}}) = \alpha$ (see [Shiryaev, 1978]). Also, as before, we note that for every stopping time τ, there exists a policy $\mu = (\mu_1, \mu_2, \cdots)$ such that for any k, when $\tau = k$, $A_{k'} = \mu_{k'}(\mathbf{I}_{k'}) = 0$ for all $k' < k$ and $A_{k'} = \mu_{k'}(\mathbf{I}_{k'}) = 1$ for all $k' \geqslant k$. Hence, the optimal Bayesian gain given by Eqn. 5.22 becomes

$$\begin{aligned}
G(\tau^{\mathsf{MaxP}}) = \max_{\tau} \ \sum_{k=0}^{\infty} \mathsf{E}[g_k(\Theta_k, A_k)] &= \max_{\mu} \ \sum_{k=0}^{\infty} \mathsf{E}[\mathsf{E}[g_k(\Theta_k, A_k) \mid \mathbf{I}_k]] \\
&= \max_{\mu} \ \sum_{k=0}^{\infty} \mathsf{E}[\mathsf{E}[g_k(\Theta_k, \mu_k(\mathbf{I}_k)) \mid \mathbf{I}_k]] \\
&= \max_{\mu} \ \mathsf{E}\left[\sum_{k=0}^{\infty} \mathsf{E}[g_k(\Theta_k, \mu_k(\mathbf{I}_k)) \mid \mathbf{I}_k]\right]
\end{aligned} \tag{5.23}$$

where the last step follows from monotone convergence theorem. From Eqn. 5.20, we see that

$$g_k(\Theta_k, \mu_k(\mathbf{I}_k)) = -\lambda_f \mathbf{1}_{\{\Theta_k=0\}} \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=1\}} + \mathbf{1}_{\{\Theta_k=1\}} \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=1\}}$$

and hence,

$$\begin{aligned}\mathsf{E}[g_k(\Theta_k,\mu_k(\mathbf{I}_k)) \mid \mathbf{I}_k] &= \mathsf{E}\big[-\lambda_f \mathbf{1}_{\{\Theta_k=0\}} \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=1\}} + \mathbf{1}_{\{\Theta_k=1\}} \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=1\}} \mid \mathbf{I}_k\big] \\ &= \big(-\lambda_f \cdot \mathsf{E}\big[\mathbf{1}_{\{\Theta_k=0\}} \mid \mathbf{I}_k\big] + \mathsf{E}\big[\mathbf{1}_{\{\Theta_k=1\}} \mid \mathbf{I}_k\big]\big) \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=1\}} \\ &= (-\lambda_f \cdot \Pi_{k,0} + \Pi_{k,1}) \cdot \mathbf{1}_{\{\mu_k(\mathbf{I}_k)=1\}}.\end{aligned}$$

From the above equation, it is easy to see that at time k, a sufficient statistic ([Bertsekas, 2005]) is the probability vector $\mathbf{\Pi}_k = [\Pi_{k,0}, \Pi_{k,1}, \Pi_{k,2}]$ for this problem also. We define the reward function $\widetilde{g} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$ as

$$\widetilde{g}(s,a) = \begin{cases} -\lambda_f \cdot p_0 + p_1, & \text{if } s = [p_0,p_1,p_2] \in \mathcal{P}, a = 1 \\ 0, & \text{otherwise.}\end{cases} \tag{5.24}$$

Since $\mathbf{\Pi}_k$ is a sufficient statistic for $\mathbf{I}_k$, for any policy μ_k there exists a corresponding policy $\widetilde{\mu}_k$ such that $\widetilde{\mu}_k(\mathbf{\Pi}_k) = \mu_k(\mathbf{I}_k)$, and hence, Eqn. 5.22 can be written as

$$\begin{aligned}G(\tau^{\mathsf{MaxP}}) &= \max_{\mu} \ \mathsf{E}\left[\sum_{k=0}^{\infty} \mathsf{E}[g_k(\Theta_k,\mu_k(\mathbf{I}_k)) \mid \mathbf{I}_k]\right] \\ &= \max_{\widetilde{\mu}} \ \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{g}(\mathbf{\Pi}_k, \widetilde{\mu}_k(\mathbf{\Pi}_k))\right] \end{aligned} \tag{5.25}$$

The tuple $(\mathcal{S}, \mathcal{A}, \mathbf{\Phi}(\cdot,\cdot,\cdot), g)$ defines a Markov decision process (MDP). We thus solve the problem defined in Eqn. 5.21 as follows. Since the one stage cost function and the transition kernel are time–invariant, it is sufficient to look for stationary policies. Let $\widetilde{\mu} : \mathcal{S} \to \mathcal{A}$ be a stationary policy. Then the total reward on using the policy $\widetilde{\mu}$ is given by

$$J_{\widetilde{\mu}}(\boldsymbol{\pi}) = \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{g}(\mathbf{\Pi}_k, \widetilde{\mu}(\mathbf{\Pi}_k)) \mid \mathbf{\Pi}_0 = \boldsymbol{\pi}\right]. \tag{5.26}$$

Let $\mu^* : \mathcal{S} \to \mathcal{A}$ be an optimal stationary policy. The optimal reward is then given by

$$\begin{aligned} J^*(\boldsymbol{\pi}) &= \max_{\widetilde{\mu}} \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{g}(\mathbf{\Pi}_k, \widetilde{\mu}(\mathbf{\Pi}_k)) \mid \mathbf{\Pi}_0 = \boldsymbol{\pi}\right] \\ &= \mathsf{E}\left[\sum_{k=0}^{\infty} \widetilde{g}(\mathbf{\Pi}_k, \mu^*(\mathbf{\Pi}_k)) \mid \mathbf{\Pi}_0 = \boldsymbol{\pi}\right] \qquad (5.27) \end{aligned}$$

The DP that solves Eqn. 5.27 is given by Bellman's equation as follows

$$\begin{aligned} J^*(\boldsymbol{\pi}) &= \max\left\{\widetilde{g}(\boldsymbol{\pi}, 1),\ \widetilde{g}(\boldsymbol{\pi}, 0) + \underbrace{\mathsf{E}\left[J^*\left(\mathbf{\Phi}(\boldsymbol{\pi}, 0, \mathbf{X})\right)\right]}_{A_{J^*}(\boldsymbol{\pi})}\right\} \\ &=: \max\left\{-\lambda_f \cdot \pi_0 + \pi_1,\ \underbrace{\mathsf{E}\left[J^*\left(\mathbf{\Phi}(\boldsymbol{\pi}, 0, \mathbf{X})\right)\right]}_{A_{J^*}(\boldsymbol{\pi})}\right\} \qquad (5.28) \end{aligned}$$

We now show the structural properties of J^* in the following theorem.

Theorem 5.5 *$J^*(\boldsymbol{\pi})$ is convex in $\boldsymbol{\pi}$.*

Also, it is easy to see from Eqn. 5.28 that the optimal stopping rule τ^{MaxP} is given by

$$\tau^{\mathsf{MaxP}} = \inf\left\{k \geqslant 0 : -\lambda_f \cdot \Pi_{k,0} + \Pi_{k,1} \geqslant A_{J^*}(\mathbf{\Pi}_k)\right\}. \qquad (5.29)$$

5.6 Numerical Results

In this section, we study the mean detection delay, ADD and the probability of detection, $\mathsf{P_D}$ performance of the transient change detection procedures, MinD, A–MinD, and MaxP. We compare the ADD and the $\mathsf{P_D}$ performance with the well known CUSUM procedure ([Basseville and Nikiforov, 1993]). Note that $\mathsf{A} - \mathsf{MinD}$ and CUSUM are simple threshold rules whereas the procedures MinD and MaxP require the solution of the DPs defined in Eqns. 5.12 and 5.28.

We assume the following parameters for numerical studies, $f_0 \sim \mathcal{N}(0, 1)$, $f_1 \sim \mathcal{N}(1, 1)$, $\rho_{01} = 0.01$, $\rho_{12} = 0.1$, and $\rho = 0$. We study the ADD, $\widetilde{\mathsf{ADD}} := \mathsf{E}\left[(\tau - \Gamma_1)^+ | \Theta_\tau = 1\right]$ and the $\mathsf{P_D}$ performance of all the detection procedures we propose, for various values of

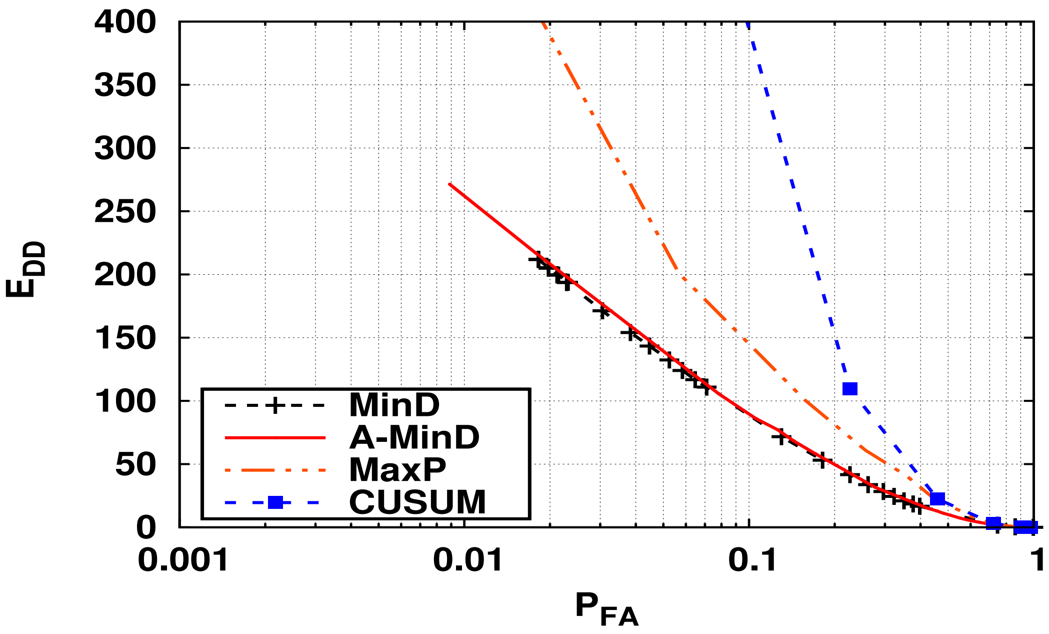


Figure 5.3: Mean detection delay (ADD) vs probability of false alarm ($\mathsf{P_{FA}}$) for $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(1,1)$, $\rho_{01} = 0.01, \rho_{12} = 0.1$, and $\rho = 0$.

$\mathsf{P_{FA}}$. We obtain the optimal policies using the value–iteration technique (see [Bertsekas, 2000a]), the number of iterations being taken as 1000. For each of the optimal policies we thus obtained, we obtain the ADD , $\widetilde{\mathsf{ADD}}$, and $\mathsf{P_D}$ for a range of $\mathsf{P_{FA}}$ by simulation, the number of simulation runs being made is 100. Thus, we obtain the ADD , $\widetilde{\mathsf{ADD}}$, and $\mathsf{P_D}$ for a range of $\mathsf{P_{FA}}$ and plot the results in Figs. 5.3 – 5.5. Our numerical results show that while the MinD procedure achieves the least ADD across all events (whether stopped in the in–change state or in the out–of–change state), the CUSUM procedure outperforms the MinD procedure when we consider only the events that are stopped in the in–change state. Also, we see from Figs. 5.3–5.5 that the ADD of MinD is approximately equal to that of $\mathsf{A - MinD}$. We also observe that the MaxP procedure results in the largest value of $\mathsf{P_D}$ as expected.

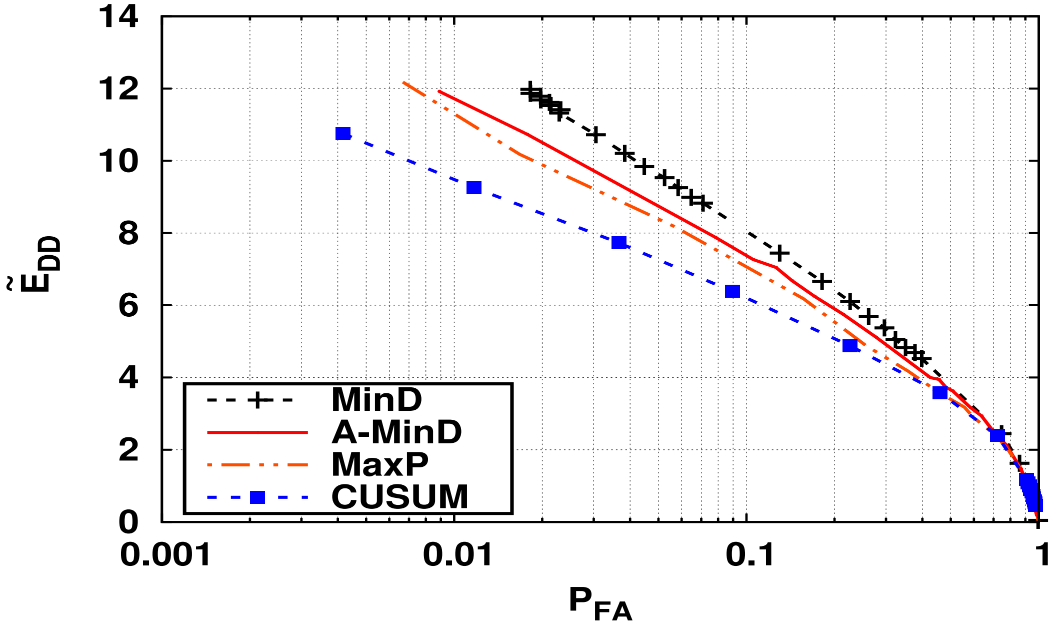


Figure 5.4: Mean detection delay of events stopped in state 1 ($\widetilde{\mathsf{ADD}}$) vs probability of false alarm ($\mathsf{P_{FA}}$) for $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(1,1)$, $\rho_{01} = 0.01, \rho_{12} = 0.1$, and $\rho = 0$.

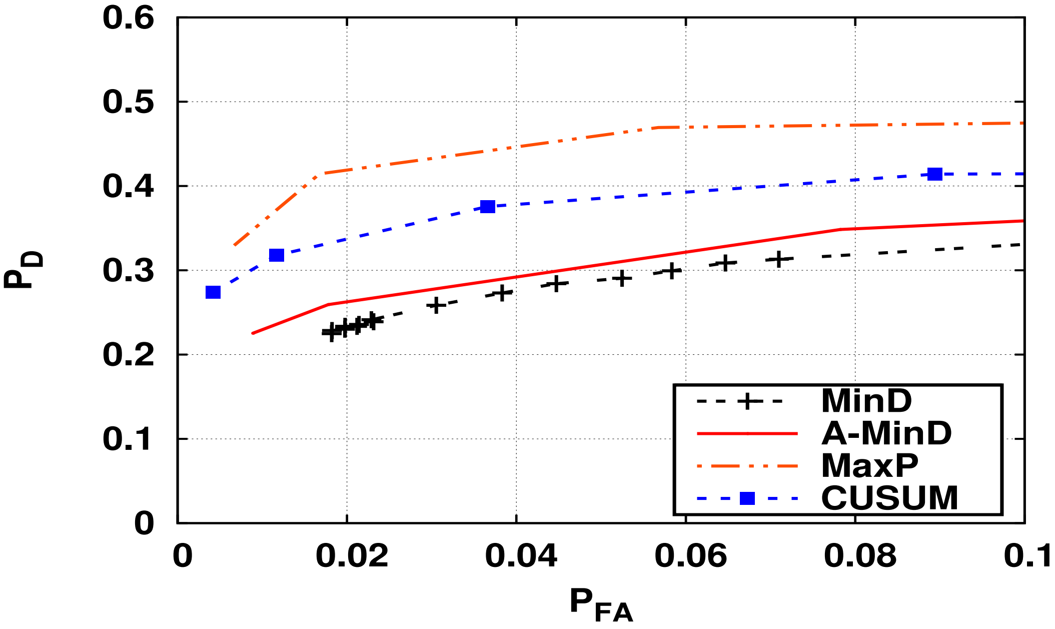


Figure 5.5: Mean probability of detection ($\mathsf{P_D}$) vs probability of false alarm ($\mathsf{P_{FA}}$) for $f_0 \sim \mathcal{N}(0,1)$, $f_1 \sim \mathcal{N}(1,1)$, $\rho_{01} = 0.01, \rho_{12} = 0.1$, and $\rho = 0$.

5.7 Conclusion

We consider the change detection problem when the event that cause the change is transient (i.e., not persistent). For a given constraint on the probability of false alarm, we model the transient change detection problem as an MDP and obtain the following Bayesian transient change detection procedures: 1) MinD, 2) $\mathsf{A-MinD}$ and 3) MaxP. We show that at any time k, the posterior probability vector (of states) up to time k is sufficient to detect the transient change. We also show some structural results of these optimum policies. Also, we compare the transient change detection procedures, MinD, $\mathsf{A-MinD}$, and MaxP with the well known CUSUM procedure. Our numerical results show that the ADD of the MinD procedure is approximately equal to that of the $\mathsf{A-MinD}$ procedure, the later procedure being much easier to implement. We also showed that the MaxP procedure results in the largest value of $\mathsf{P_D}$ as expected. Finally, we observed that $\widetilde{\mathsf{ADD}}$ is the smallest for the CUSUM procedure.

5.8 Appendix

Proof of Theorem 5.1

We show by induction that J^* is concave in $\mathbf{p}$. The value functions for the DP defined in Eqn. 5.12 are given by

$$\begin{aligned} V_0(\mathbf{p}) &= p_0 \\ V_{k+1}(\mathbf{p}) &= \min\left\{p_0, c(1-p_0) + \mathsf{E}V_k\left(\mathbf{\Phi}\big(\mathbf{p}, 0, \mathbf{X}\big)\right)\right\}, k = 1, 2, \cdots \end{aligned}$$

V_0, being an affine function, is concave in $\mathbf{p}$. For some $k \geqslant 1$, assume that V_k is concave in $\mathbf{p}$. Let $\mathbf{p} = [p_0, p_1, p_2]$ and $\mathsf{DN} = (p_0\rho_{00} + p_1\rho_{12} + p_2)\, f_0(\mathbf{x}) + (p_0\rho_{01} + p_1\rho_{11})\, f_1(\mathbf{x})$.

Then

$$\begin{aligned} V_k(\mathbf{p}) &= \inf_{(a_0,a_1,a_2,a_3)\in\mathcal{C}} \{a_0 + a_1 p_0 + a_2 p_1 + a_3 p_2\} \quad \text{(see [Rockafellar, 1997])} \\ \Longrightarrow\; V_k\big(\mathbf{\Phi}\left(\mathbf{p},0,\mathbf{X}\right)\big) \cdot \mathsf{DN} &= \inf_{a_0,a_1,a_2,a_3\in\mathcal{C}} \{\text{an affine function }\} \end{aligned}$$

where $\mathcal{C} := \{(a_0,a_1,a_2,a_3) \in \mathbb{R}^4 : a_0 + a_1 p_0 + a_2 p_1 + a_3 p_2 \geqslant V_k(\mathbf{p})\}$. Since integration is a linear operator, the above implies that $\mathsf{E} V_k\big(\mathbf{\Phi}\left(\mathbf{p},0,\mathbf{X}\right)\big)$ is concave in $\mathbf{p}$ or $V_{k+1}(\mathbf{p})$ is concave in $\mathbf{p}$. Hence, in the same way as was shown in the Appendix of Chapter 3, we can show that $J^*(\mathbf{p})$ and $A_{J^*}(\mathbf{p})$ are concave in $\mathbf{p}$. ■

Proof of Theorem 5.2

The proof of this theorem is along the same lines as in [Tartakovsky and Veeravalli, 2005].

$$\begin{aligned} \mathsf{E}\left[\tau - T \,\middle|\, \tau \geqslant T\right] &= \sum_{k_1=0}^{\infty} \sum_{k_2=k_1+1}^{\infty} \mathsf{P}\left\{T = k_1, E = k_2 \mid \tau \geqslant T\right\} \cdot \mathsf{E}\left[\tau - T \,\middle|\, \tau \geqslant T, T = k_1, E = k_1\right] \\ &= \sum_{k_1=0}^{\infty} \sum_{k_2=k_1+1}^{\infty} \mathsf{P}\left\{T = k_1, E = k_2 \mid \tau \geqslant T\right\} \cdot \mathsf{E}_{k_1,k_2}\left[\tau - k_1 \,\middle|\, \tau \geqslant k_1\right]. \end{aligned}$$

Thus, to obtain a lower bound for $\mathsf{E}\left[\tau - T \,\middle|\, \tau \geqslant T\right]$, it is sufficient to obtain a lower bound for $\mathsf{E}_{k_1,k_2}\left[\tau - k_1 \,\middle|\, \tau \geqslant k_1\right]$.

For any $D > 0$, we have

$$\begin{aligned}
\mathsf{E}_{k_1,k_2}\left[\tau - k_1 \,\middle|\, \tau \geqslant k_1\right] &= \frac{\mathsf{E}_{k_1,k_2}\left[(\tau - k_1)^+\right]}{\mathsf{P}_{k_1,k_2}\left\{\tau \geqslant k_1\right\}} \\
&\geqslant \frac{\mathsf{E}_{k_1,k_2}\left[(\tau - k_1)^+ \mid \tau - k_1 \geqslant D\right] \cdot \mathsf{P}_{k_1,k_2}\left\{\tau - k_1 \geqslant D\right\}}{\mathsf{P}_{k_1,k_2}\left\{\tau \geqslant k_1\right\}} \\
&\geqslant \frac{D \cdot \mathsf{P}_{k_1,k_2}\left\{\tau \geqslant k_1 + D\right\}}{\mathsf{P}_{k_1,k_2}\left\{\tau \geqslant k_1\right\}} \\
&= D\,\frac{\mathsf{P}_{k_1,k_2}\left\{\tau \geqslant k_1\right\} - \mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D\right\}}{\mathsf{P}_{k_1,k_2}\left\{\tau \geqslant k_1\right\}} \\
&= D\left[1 - \frac{\mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D\right\}}{\mathsf{P}_{k_1,k_2}\left\{\tau \geqslant k_1\right\}}\right] \\
&= D\left[1 - \frac{\mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D\right\}}{1 - \mathsf{P}_{k_1,k_2}\left\{\tau < k_1\right\}}\right] \\
&= D\left[1 - \frac{\mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D\right\}}{1 - \mathsf{P}_{\infty}\left\{\tau < k_1\right\}}\right], \qquad (5.30)
\end{aligned}$$

where the equality in the last step, $\mathsf{P}_{k_1,k_2}\left\{\tau < k_1\right\} = \mathsf{P}_{\infty}\left\{\tau < k_1\right\}$ is justified by the following argument. The event $\{\tau < k_1\} = \{\tau \leqslant k_1 - 1\} \in \sigma\left(\mathbf{X}_{[1:k_1-1]}\right)$, and since $T = k_1$, $\mathbf{X}_k \sim f_0$ for all $k \leqslant k_1 - 1$, we have the equality. Also, note that $\tau \in \Delta(\alpha)$. Hence,

$$\begin{aligned}
\mathsf{P}\left\{\tau < T\right\} &\leqslant \alpha \\
\iff \mathsf{P}\left\{\tau < T, T > k_1\right\} + \mathsf{P}\left\{\tau < T, T \leqslant k_1\right\} &\leqslant \alpha \\
\implies \mathsf{P}\left\{\tau < T, T > k_1\right\} &\leqslant \alpha \\
\implies \mathsf{P}\left\{\tau < k_1, T > k_1\right\} &\leqslant \alpha \quad (\because \{\tau < k_1, T > k_1\} \subseteq \{\tau < T, T > k_1\}) \\
\implies \mathsf{P}\left\{T > k_1\right\} \cdot \mathsf{P}\left\{\tau < k_1 \,\middle|\, T > k_1\right\} &\leqslant \alpha \\
\implies \mathsf{P}\left\{T > k_1\right\} \cdot \mathsf{P}_{\infty}\left\{\tau < k_1\right\} &\leqslant \alpha \quad (\because \text{the same argument as above}) \\
\implies \mathsf{P}\left\{T > k_1\right\} \cdot \mathsf{P}_{\infty}\left\{\tau < k_1\right\} &\leqslant \alpha \\
\implies \mathsf{P}_{\infty}\left\{\tau < k_1\right\} &\leqslant \frac{\alpha}{\rho_{00}^{k_1}}, \qquad (5.31)
\end{aligned}$$

From Eqns. 5.30 and 5.31, we have

$$\begin{aligned}\mathsf{E}_{k_1,k_2}\left[\tau - k_1 \,\middle|\, \tau \geqslant k_1\right] &\geqslant D\left[1 - \frac{\mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D\right\}}{1 - \mathsf{P}_\infty\left\{\tau < k_1\right\}}\right] \\ &\geqslant D\left[1 - \frac{\mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D\right\}}{1 - \frac{\alpha}{\rho_{00}^{k_1}}}\right]. \qquad (5.32)\end{aligned}$$

We are interested in obtaining a lower bound for the delay, or an upper bound for $\mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D\right\}$. We use the change of measure argument to obtain the probability, $\mathsf{P}_\infty\left\{k_1 \leqslant \tau < k_1 + D\right\}$ from $\mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D\right\}$, and use the upper bound for $\mathsf{P}_\infty\left\{k_1 \leqslant \tau < k_1\right\}$ from Eqn. 5.31. We define the sum of the log–likelihood ratios of the observations up to k, when the change points are $T = k_1$, $E = k_2$, as

$$\Lambda_k^{(k_1,k_2)} = \sum_{k'=k_1}^{\min\{k_2-1,k\}} \ln\left(\frac{f_1(\mathbf{X}_{k'})}{f_0(\mathbf{X}_{k'})}\right).$$

Note that for $k < k_1$, $\Lambda_k^{(k_1,k_2)} = 0$, and for $k \geqslant k_2$, $\Lambda_k^{(k_1,k_2)} = \Lambda_{k_2}^{(k_1,k_2)}$. Hence,

$$\frac{1}{k}\Lambda_k^{(k_1,k_2)} \;\to\; 0, \mathsf{P}_{k_1,k_2} \text{ a.s. as } k \to \infty.$$

For any $C > 0$,

$$\begin{aligned}\mathsf{P}_\infty\left\{k_1 \leqslant \tau < k_1 + D\right\} &= \mathsf{E}_\infty\left[\mathbf{1}_{\{k_1 \leqslant \tau < k_1 + D\}}\right] \\ &= \mathsf{E}_{k_1,k_2}\left[\mathbf{1}_{\{k_1 \leqslant \tau < k_1 + D\}} e^{-\Lambda_\tau^{(k_1,k_2)}}\right] \quad (\because \text{by change of measure argument}) \\ &\geqslant \mathsf{E}_{k_1,k_2}\left[\mathbf{1}_{\left\{k_1 \leqslant \tau < k_1 + D, \Lambda_\tau^{(k_1,k_2)} < C\right\}} e^{-\Lambda_\tau^{(k_1,k_2)}}\right] \\ &\geqslant e^{-C} \cdot \mathsf{E}_{k_1,k_2}\left[\mathbf{1}_{\left\{k_1 \leqslant \tau < k_1 + D, \Lambda_\tau^{(k_1,k_2)} < C\right\}}\right] \\ &= e^{-C} \cdot \mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D, \Lambda_\tau^{(k_1,k_2)} < C\right\} \\ &\geqslant e^{-C} \cdot \mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D, \max_{k_1 \leqslant n < k_1 + D} \Lambda_n^{(k_1,k_2)} < C\right\} \\ &\geqslant e^{-C} \cdot \left[\mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D\right\} - \mathsf{P}_{k_1,k_2}\left\{\max_{k_1 \leqslant n < k_1 + D} \Lambda_n^{(k_1,k_2)} \geqslant C\right\}\right]\end{aligned}$$

Therefore, for any D and C,

$$\mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D\right\} \;\leqslant\; \mathsf{P}_{k_1,k_2}\left\{\max_{k_1 \leqslant n < k_1+D} \Lambda_n^{(k_1,k_2)} \geqslant C\right\} + e^{C} \cdot \mathsf{P}_\infty\left\{k_1 \leqslant \tau < k_1 + D\right\}$$

We choose $D = D_\alpha$ in such a way that $D_\alpha \to \infty$ as $\alpha \to 0$, and then we choose $C = \epsilon D_\alpha$ for some $\epsilon > 0$. For any $\epsilon > 0$, we have

$$\begin{array}{rcl}
\mathsf{P}_{k_1,k_2}\left\{\dfrac{1}{D_\alpha}\max\limits_{k_1 \leqslant n < k_1+D_\alpha} \Lambda_n^{(k_1,k_2)} \geqslant \epsilon\right\} & \to & 0, \text{ as } \alpha \to 0 \\
\text{or } \; \mathsf{P}_{k_1,k_2}\left\{\max\limits_{k_1 \leqslant n < k_1+D_\alpha} \Lambda_n^{(k_1,k_2)} \geqslant \epsilon D_\alpha\right\} & = & o(1), \;\; \alpha \to 0 \\
\text{and } \; e^{\epsilon D_\alpha} \cdot \mathsf{P}_\infty\left\{k_1 \leqslant \tau < k_1 + D_\alpha\right\} & \leqslant & e^{\epsilon D_\alpha} \cdot \mathsf{P}_\infty\left\{\tau < k_1 + D_\alpha\right\} \\
 & \leqslant & e^{\epsilon D_\alpha} \cdot \dfrac{\alpha}{\rho_{00}^{k_1+\lceil D_\alpha \rceil}} \\
 & \leqslant & e^{\epsilon D_\alpha} \cdot \dfrac{\alpha}{\rho_{00}^{k_1+1+D_\alpha}} \\
 & = & \dfrac{1}{\rho_{00}^{k_1+1}} \dfrac{\alpha}{\exp\left((\ln(\rho_{00}) - \epsilon) D_\alpha\right)}
\end{array} \tag{5.33}$$

Therefore, choosing $D_\alpha = \frac{q\ln(\alpha)}{\ln(\rho_{00})-\epsilon}$ for some $0 < q < 1$, we have

$$\begin{array}{rcl}
\mathsf{P}_{k_1,k_2}\left\{k_1 \leqslant \tau < k_1 + D_\alpha\right\} & \leqslant & o(1) + \dfrac{\alpha^{1-q}}{\rho_{00}^{k_1+1}} \\
 & = & \dfrac{o(1) + \alpha^{1-q}}{\rho_{00}^{k_1+1}}
\end{array} \tag{5.34}$$

Therefore, from Eqn. 5.30, we have

$$\mathsf{E}_{k_1,k_2}\left[\tau - k_1 \,\middle|\, \tau \geqslant k_1\right] \;\geqslant\; \frac{q\ln(\alpha)|}{\ln(\rho_{00}) - \epsilon} \cdot \left[1 - \frac{o(1) + \alpha^{1-q}}{\rho_{00}^{k_1+1} - \alpha\rho_{00}}\right].$$

Hence, as $\alpha \to 0$, we have

$$\mathsf{E}_{k_1,k_2}\left[\tau - k_1 \,\middle|\, \tau \geqslant k_1\right] \;\geqslant\; \frac{q\ln(\alpha)}{\ln(\rho_{00}) - \epsilon} \cdot \left[1 - o(1)\right], \quad \text{as } \alpha \to 0.$$

Note that $\epsilon > 0$ and $0 < q < 1$ can be arbitrarily chosen and hence,

$$\mathsf{E}_{k_1,k_2}\left[\tau - k_1 \,\middle|\, \tau \geqslant k_1\right] \;\geqslant\; \frac{\ln(\alpha)}{\ln(\rho_{00})} \cdot [1 - o(1)]\,, \quad \text{as } \alpha \to 0.$$

■

Proof of Theorem 5.3

Recall that

$$\begin{aligned}\Pi_{k,s} &= \frac{\sum_{i\in\mathcal{S}} \Pi_{k-1,i}\rho_{is} f_s(\mathbf{X}_k)}{\sum_{j\in\mathcal{S}}\sum_{i\in\mathcal{S}} \Pi_{k-1,i}\rho_{ij} f_j(\mathbf{X}_k)},\\ &:= \frac{\mathcal{N}_s}{\sum_{j\in\mathcal{S}}\mathcal{N}_j}\end{aligned}$$

where $f_2 = f_0$ and

$$\begin{aligned}\mathcal{N}_s &= \sum_{i=0}^{2} \Pi_{k-1,i}\rho_{is} f_s(\mathbf{X}_k),\\ &= \left(\sum_{i=0}^{2} \Pi_{k-1,i}\rho_{is}\right) \cdot L_{k,s} \cdot f_0(\mathbf{X}_k),\\ &= \left(\sum_{i=0}^{2} \frac{\rho_{01} Q_{k-1,i}\rho_{is}}{1 + \rho_{01}\sum_{j=1}^{2} Q_{k-1,j}}\right) \cdot L_{k,s} \cdot f_0(\mathbf{X}_k), \text{ (From Eqn. 6)}\end{aligned}$$

Therefore,

$$\begin{aligned}\Pi_{k,s} &= \frac{\left(\sum_{i=0}^{2} \rho_{01} Q_{k-1,i}\rho_{is}\right) \cdot L_{k,s} \cdot f_0(\mathbf{X}_k)}{\sum_{j=0}^{2}\left(\sum_{i=0}^{2} \rho_{01} Q_{k-1,i}\rho_{ij}\right) \cdot L_{k,j} \cdot f_0(\mathbf{X}_k)},\\ &= \frac{\left(\sum_{i=0}^{2} Q_{k-1,i}\rho_{is}\right) \cdot L_{k,s}}{\sum_{j=0}^{2}\left(\sum_{i=0}^{2} Q_{k-1,i}\rho_{ij}\right) \cdot L_{k,j}},\end{aligned}$$

Therefore,

$$
\begin{aligned}
Q_{k,s} &= \frac{\left(\sum_{i=0}^{2} Q_{k-1,i}\rho_{is}\right)\cdot L_{k,s}}{\rho_{01}\Pi_{k,0}\sum_{j=0}^{2}\left(\sum_{i=0}^{2} Q_{k-1,i}\rho_{ij}\right)\cdot L_{k,j}}, \\
&= \frac{\left(\sum_{i=0}^{2} Q_{k-1,i}\rho_{is}\right)\cdot L_{k,s}}{\rho_{01}\frac{\Pi_{k-1,0}\rho_{00}L_{k,0}f_0(\mathbf{X}_k)}{\sum_{j\in\mathcal{S}}\mathcal{N}_j}\sum_{j=0}^{2}\left(\sum_{i=0}^{2} Q_{k-1,i}\rho_{ij}\right)\cdot L_{k,j}}, \\
&= \frac{\left(\sum_{i=0}^{2} Q_{k-1,i}\rho_{is}\right)\cdot L_{k,s}}{\rho_{00}\rho_{01}\frac{\Pi_{k-1,0}f_0(\mathbf{X}_k)}{\sum_{j\in\mathcal{S}}\frac{\sum_{i\in\mathcal{S}}\rho_{01}Q_{k-1,i}\rho_{ij}L_{k,j}f_0(\mathbf{X}_k)}{1+\rho_{01}(Q_{k-1,1}+Q_{k-1,2})}}\sum_{j=0}^{2}\left(\sum_{i=0}^{2} Q_{k-1,i}\rho_{ij}\right)\cdot L_{k,j}}, \\
&= \frac{\left(\sum_{i=0}^{2} Q_{k-1,i}\rho_{is}\right)\cdot L_{k,s}}{\rho_{00}} \\
&= \frac{L_{k,s}}{\rho_{00}}\left(\sum_{i=0}^{2} Q_{k-1,i}\rho_{is}\right)
\end{aligned}
$$

$Q_{0,s}$ in terms of prior $\Pi_{0,s}$:

$$
\begin{aligned}
Q_{0,0} &= \frac{1}{\rho_{01}}, \\
Q_{0,1} &= \frac{\Pi_{0,1}}{\rho_{01}\Pi_{0,0}} \\
&= \frac{\mathsf{P}\{T=0, E>0\}}{\rho_{01}\mathsf{P}\{T>0\}} \\
&= \frac{\rho}{\rho_{01}(1-\rho)} \\
Q_{0,2} &= \frac{\Pi_{0,2}}{\rho_{01}\Pi_{0,0}} \\
&= \frac{\mathsf{P}\{T=0, E=0\}}{\rho_{01}\mathsf{P}\{T>0\}}, \\
&= 0
\end{aligned}
$$

■

Proof of Theorem 5.5

Let $\mathbf{p} = [p_0, p_1, p_2]$. If $V(\mathbf{p})$ is a convex function then

$$V(x) \quad = \quad \sup_{(a,b)\in\mathcal{C}} a_0 + a_1 p_0 + a_2 p_1 + a_3 p_2$$

where $\mathcal{C} := \{(a_0, a_1, a_2, a_3) \in \mathbb{R}^4 : a_0 + a_1 p_0 + a_2 p_1 + a_3 p_2 \leqslant V(\mathbf{p})\}$ (see [Rockafellar, 1997]). Also the point–wise maximum of two convex functions is convex. Using these facts, Theorem 5.5 can be proved along the same lines as that of Theorem 5.1. ■

Part II

Event Detection in Large Extent Networks

Chapter 6

Quickest Detection and Localisation of Events in Large Extent Networks

6.1 Introduction

In the previous chapters, we were concerned with event detection in small extent networks where the event affects all the sensor nodes in the same way. The event is considered as a source of a signal of some type (or "modality"), e.g., infrared, acoustic, etc., and hence, the measurement at a sensor node corresponds to the sum of the signal strength that the sensor receives from the event (which, in general, depends on the distance between the sensor node and the event), and the inherent sensor noise. In a small extent network, all the sensors cover the region of interest (ROI), and hence, all the sensors receive an appreciable signal strength from the event. We assumed, in Chapters 3, 4, and 5, that all the sensors receive the same signal strength. However, in a large extent network, the ROI is large compared to the coverage region of a sensor. Thus, in this case, an event affects only the sensors in the vicinity of where it occurs.

In this chapter, we consider the problem of detecting and locating an event, in a large extent network. This problem is also called *change detection and isolation* (see [Nikiforov, 1995]). Since the ROI is large, a large number of sensors are deployed to cover the ROI, making a centralised solution infeasible. In our work, *we seek distributed* *algorithms for detecting and locating an event, with small detection delay, subject to a constraint on false alarm and false isolation.* The distributed algorithms require only local information from the neighborhood of each node.

6.1.1 Summary of Contributions

1. We formulate the event detection/isolation problem in a large extent network as a worst case detection delay minimisation problem subject to a mean time to false alarm and mean time to false isolation constraints. Because of the large extent network, the postchange distribution is unknown, and the latter is a novel aspect of our problem formulation.

2. We propose distributed detection/isolation procedures MAX, ALL, and HALL (**H**ysteresis modified **ALL**) for large extent wireless sensor networks. The distributed procedures MAX and ALL are extensions of the decentralised procedures MAX [Tartakovsky and Veeravalli, 2003] and ALL [Mei, 2005], [Tartakovsky and Veeravalli, 2008], which were developed for small extent networks. The distributed procedures MAX, ALL, and HALL are computationally less complex and more energy–efficient compared to the centralised procedure given by Nikiforov [Nikiforov, 1995] (which can be applied only to the Boolean sensing model).

3. We analyse the supremum worst case detection delay (SADD) of MAX, ALL, and HALL when the mean time to false alarm (T_{FA}) and the mean time to false isolation (T_{FI}) are at least as large as a certain threshold γ. For the case of the Boolean sensing model, we compare the detection delay performance of the these distributed procedures with that of Nikiforov's procedure [Nikiforov, 1995] (a centralised procedure which is shown to be asymptotically delay optimal) and show that the distributed procedures ALL and HALL are asymptotically order optimal.

6.1.2 Discussion of Related Literature

The problem of sequential change detection/isolation with a finite set of postchange hypotheses was studied in a centralised setting by Nikiforov [Nikiforov, 1995], and Malladi and Speyer [Malladi and Speyer, 1999]. In [Nikiforov, 1995], Nikiforov formulated the non–Bayesian change detection/isolation problem and proposed a procedure which is shown to be worst case detection delay optimal, as $\min\{\mathsf{T}_{\mathsf{FA}}, \mathsf{T}_{\mathsf{FI}}\}$ goes to ∞. It is to be noted that the decision statistic of Nikiforov's procedure can not be computed in a recursive manner, and hence, the computational complexity of Nikiforov's procedure is high. In [Malladi and Speyer, 1999], Malladi and Speyer studied a Bayesian change detection/isolation problem and obtained a mean delay optimal centralised procedure which is a threshold based rule on the a posteriori probability of change corresponding to each post–change hypothesis.

Centralised procedures incur high communication costs and distributed procedures would be desirable. In this chapter, we study distributed procedures based on CUSUM detectors at the sensor nodes where the CUSUM detector at sensor node i is driven only by the observations made at node i.

The previous work on decentralised change detection [Tartakovsky and Veeravalli, 2003], [Mei, 2005], [Tartakovsky and Kim, 2006], etc., focuses on collocated networks where the postchange distribution of observations is known. However, in the case of large extent networks, we have a change detection/isolation problem where the postchange distribution of the observations of a sensor node, in general, depends on the distance between the event and the sensor node which is unknown.

6.1.3 Outline of the Chapter

The rest of the chapter is organised as follows. In Section 6.2, we provide the models for the occurrence of the event and the observations made by the sensors. We define the detection–coverage region of a sensor and partition the ROI into a minimum number of subregions $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_N$ where each subregion $\mathcal{A}_i$ is detection–covered by a unique set of sensors $\mathcal{N}_i$. We also define the detection subregion and the influence subregion of a set of sensor nodes, and motivate the problem by an example. In Section 6.3, we define the change detection/isolation problem as a multi–hypothesis testing problem where each hypothesis $\mathbf{H}_i$ corresponds to the detection of the event by the set of sensor nodes $\mathcal{N}_i$. We define the performance metrics (SADD, T_{FA}, and T_{FI}), and formulate the change detection/isolation problem. In Section 6.4, we propose distributed event detection/isolation procedures MAX, ALL and HALL which make a decision based on the local decisions of all sensor nodes in each set of sensors $\mathcal{N}_i$. We study the SADD, T_{FA}, and the T_{FI} performance of the distributed procedures, and discuss the asymptotic minimax delay optimality of these procedures. In Section 6.5, we provide numerical results and we conclude in Section 6.6.

6.2 System Model

Let $\mathcal{A} \subset \mathbb{R}^2$ be the region of interest (ROI) in which a WSN has to be engineered for event detection. We deploy n sensor nodes in the region $\mathcal{A}$. We assume that all nodes are equipped with the same type of sensor (e.g., acoustic or passive infrared). Let $\ell^{(i)} \in \mathcal{A}$ be the location of node i, and define $\boldsymbol{\ell} := [\ell^{(1)}, \ell^{(2)}, \cdots, \ell^{(n)}]$. We consider a discrete–time system, with the basic unit of time being one slot. The slots are indexed by non–negative integers. We assume that a slot is of unit length and that slot k is defined by the time interval $[k, k+1)$. The sensor nodes are assumed to be time–synchronised (see, for example, [Solis et al., 2006]), and at the beginning of every slot $k \geqslant 1$, each sensor node i samples its environment and obtains the observation $X_k^{(i)} \in \mathbb{R}$.

6.2.1 Event Model

An event occurs at an unknown time $T \in \mathbb{Z}_+$ and at an unknown location $\ell_e \in \mathcal{A}$. In this work, we consider only stationary (and permanent or persistent) point events, i.e., an event occurs at a point in the region of interest, and *having occurred, stays there forever.* Examples that would motivate such a model are 1) gas leakage in the wall of a large storage tank, 2) excessive strain at a point in a large flat structure. [Polunchenko and Tartakovsky, 2009] and [Premkumar et al., 2010] provide studies of change detection problems in which the event stays only for a finite random amount of time.

6.2.2 Sensing Model

An event is viewed as a source of some physical signal that can be sensed by the sensor nodes that have been deployed. Let h_e be the signal strength of the event[1]. A sensor at a distance d from the event senses a signal $h_e\rho(d)+W$, where W is random (sensing) zero mean noise and $\rho(d)$ is the model for distance dependent loss in signal strength which is a decreasing function of the distance d. Note that we have assumed an isotropic distance dependent loss model, whereby the mean signal received by all sensors at a distance d (from the event) is the same. We see some examples below for the sensing models.

Example 6.1 The Boolean model (see [Liu and Towsley, 2004]): In this model, the signal strength that a sensor receives is the same (which is given by μ_1) when the event occurs within a distance of r_s from the sensor and is 0 otherwise. Thus, for a Boolean sensing model[2],

$$\rho(d) \quad = \quad \begin{cases} 1, & \text{if } d \leqslant r_s \\ 0, & \text{otherwise.} \end{cases}$$

Example 6.2 The power law path–loss model (see [Liu and Towsley, 2004]) is given by

$$\rho(d) \quad = \quad d^{-\eta},$$

[1] In case, the signal strength of the event is not known, but is known to lie in an interval $[\underline{h}, \overline{h}]$, we work with $h_e = \underline{h}$ as this corresponds to the least Kullback–Leibler divergence between the "*event not occurred*" hypothesis and the "*event occurred*" hypothesis. See [Tartakovsky and Polunchenko, 2008] for change detection with unknown parameters for a collocated network.

[2] The detection region of a sensor under the Boolean sensing model is a disc of radius r_s. One can generalise the notion of the Boolean model by having an arbitrary shape for the detection region of a sensor. This model can be useful in studying sensors with directional sensitivity.

for some path loss exponent $\eta > 0$.

In Example 2, we see that the signal from an event varies continuously over the region. Hence, unlike the Boolean model, there is no clear demarcation between the sensors that observe the event and those that do not. Thus, in order to facilitate the design of a distributed detection scheme with some performance guarantees, in the remainder of this section, we will define certain regions around each sensor.

6.2.3 Detection Region and Detection Partition

Definition 6.1 The **Detection Range** of a sensor r_s, is defined as the distance from the sensor within which the occurrence of an event induces a signal level of at least $\mu_1 > 0$, i.e.,

$$r_s \quad := \quad \sup\left\{d : h_e \rho(d) \geqslant \mu_1\right\}.$$

■

The idea is that if an event occurs in the detection regions of some sensors then those sensors, seeing a large signal, can rapidly detect the change. We will see that μ_1 is a design parameter that defines the acceptable detection delay. For a given signal strength h_e, a large value of μ_1 results in a small detection range r_s (as $\rho(d)$ is non–increasing in d). Also, we will see in Section 6.4.4 (Eqn. (6.10)) that the detection delay (SADD) of the distributed change detection/isolation procedures we propose, depends on the detection range r_s, and that a small r_s (i.e., a large μ_1) results in a small detection delay (SADD). It will be clear from the following discussion that a small detection range r_s requires more sensors to be deployed.

We say that a location $x \in \mathsf{ROI}$ is *detection–covered* by sensor node i, if x lies within the detection range of sensor i, i.e., if $\|\ell^{(i)} - x\| \leqslant r_s$, where we recall that $\ell^{(i)}$ is the location of the sensor node i. For any sensor node i, let $\mathcal{S}^{(i)} \subseteq \mathcal{A}$ be the *detection–coverage region* of node i, i.e., any point $x \in \mathcal{S}^{(i)}$ is detection–covered by sensor node i. Thus, $\mathcal{S}^{(i)} = \{x \in \mathcal{A} : \|\ell^{(i)} - x\| \leqslant r_s\}$ (see Fig. 6.1).

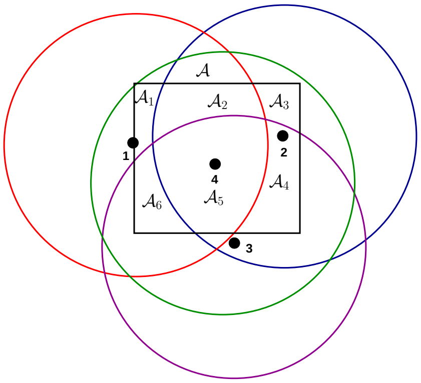


Figure 6.1: **Partitioning of $\mathcal{A}$ in a large WSN by detection regions**: A simple example of partitioning of $\mathcal{A}$ in a large WSN. The coloured solid circles around each sensor node denote their detection regions. The four sensor nodes, in the figure, divide the ROI, indicated by the square region, into regions $\mathcal{A}_1, \cdots, \mathcal{A}_6$ such that region $\mathcal{A}_i$ is detection–covered by a unique set of sensors $\mathcal{N}_i$.

We assume that the sensor deployment is such that every $x \in \mathcal{A}$ is detection–covered by at least one sensor (we see that this is the case in the example in Fig. 6.1). For each $x \in \mathcal{A}$, define $\mathcal{N}(x)$ to be the largest set of sensors by which x is detection–covered, i.e., $\mathcal{N}(x) := \{i : x \in S^{(i)}\}$. Let $\mathcal{C}(\mathcal{N})$ be the collection of all such sensor–sets, i.e., $\mathcal{C}(\mathcal{N}) = \{\mathcal{N}(x) : x \in \mathsf{ROI}\}$. Note that $\mathcal{C}(\mathcal{N})$ is a finite set and it can have at most $2^n - 1$ elements (since all points in the ROI are detection–covered). Let N be the number of elements of $\mathcal{C}(\mathcal{N})$. For each $\mathcal{N}_i \in \mathcal{C}(\mathcal{N})$, we denote the corresponding detection–covered region by $\mathcal{A}_i = \mathcal{A}(\mathcal{N}_i) := \{x \in \mathsf{ROI} : \mathcal{N}(x) = \mathcal{N}_i\}$. Evidently, the $\mathcal{A}_i, 1 \leqslant i \leqslant N$, partition the ROI. We say that the ROI is *detection–partitioned* into a *minimum number of subregions*, $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_N$, such that the subregion $\mathcal{A}_i$ is detection–covered by (each sensor in) a unique set of sensors $\mathcal{N}_i$, and $\mathcal{A}_i$ is the maximal detection–covered region of $\mathcal{N}_i$, i.e., $\forall i \neq i'$, $\mathcal{N}_i \neq \mathcal{N}_{i'}$ and $\mathcal{A}_i \cap \mathcal{A}_{i'} = \emptyset$. See Fig. 6.1 for a four sensor example.

6.2.4 Measurement Model

Before change, i.e., for $k < T$, the observation $X_k^{(i)}$ is just the zero mean sensor noise $W_k^{(i)}$, the probability density function (pdf) of which is denoted by $f_0(\cdot)$. After change, i.e., for $k \geqslant T$ and the location of the event is ℓ_e, the observation of sensor i is given by $h_e \rho(d_{ei}) + W_k^{(i)}$ where $d_{ei} := \|\ell^{(i)} - \ell_e\|$ is the distance of sensor node i from the event. We denote the postchange pdf of $X_k^{(i)}$ by $f_1(\cdot; d_{ei})$. The noise processes $\{W_k^{(i)}\}$ are independent and identically distributed (iid) across time and across sensor nodes.

6.2.5 CUSUM as the Local Detector

We shall propose distributed procedures for change detection/isolation based on using the CUSUM statistic [Basseville and Nikiforov, 1993], at *each node*. CUSUM is a non–Bayesian change detection procedure which is designed for single prechange and postchange pdfs. We outline the CUSUM procedure as follows. At each time $k \geqslant 1$, the CUSUM statistic S_k is given by

$$S_k \quad := \quad (S_{k-1} + Z_k)^+ \tag{6.1}$$

where $S_0 := 0$ and Z_k is the log likelihood–ratio (LLR) of the observation $\mathbf{X}_k$ between the postchange and the prechange pdfs. The CUSUM rule is to declare the change (in the distribution of the observations) when the statistic S_k crosses a threshold c, i.e.,

$$\tau^{\mathsf{CUSUM}} \quad = \quad \inf\left\{k : S_k \geqslant c\right\} \tag{6.2}$$

where the threshold c is chosen such that the mean time to false alarm exceeds a certain threshold.

We compute a CUSUM statistic at each sensor i based only on the observations of sensor i, the driving term of which should be the log likelihood–ratio (LLR) of the observation $X_k^{(i)}$ defined as $Z_k^{(i)}(d_{ei}) := \ln\left(\frac{f_1(X_k^{(i)};d_{ei})}{f_0(X_k^{(i)})}\right)$. As the location of the event ℓ_e is unknown, the distance of each sensor i from the event, d_{ei} is also unknown. Hence, one cannot work with the pdfs $f_1(\cdot; d_{ei})$. We propose to drive the CUSUM at each node i with $Z_k^{(i)}(r_s)$, where we recall that r_s is the detection range of a sensor, i.e., we define the statistic $C_k^{(i)}$ as follows.

$$C_k^{(i)} \quad := \quad \left(C_{k-1}^{(i)} + Z_k^{(i)}(r_s)\right)^+, \quad k = 1, 2, \cdots,$$

and $C_0^{(i)} := 0$. Based on the CUSUM statistic $C_k^{(i)}$, sensor i computes a local decision $D_k^{(i)} \in \{0,1\}$, where 0 represents no–change and 1 represents change. For each set of sensor nodes $\mathcal{N}_i$ that detection partitions the ROI, we define $\tau^{(\mathcal{N}_i)}$, the stopping time (based on the local decisions $D_k^{(j)}$s for all $j \in \mathcal{N}_i$) at which the set of sensors $\mathcal{N}_i$ detects the event. The way we obtain the local decision $D_k^{(j)}$ from the CUSUM statistic $C_k^{(j)}$ varies from rule to rule. Specific rules for local decision and the fusion of local decisions will be described in Section 6.4. Also, the event is isolated to a region associated with the sensor set that detects the event, called the influence region, which we define in the following subsection.

6.2.6 Influence Region

With the local detection statistic being as defined in the previous subsection, let us recall the two examples in Section 6.2.2. With the Boolean sensing model, after an event occurs, the CUSUMs of sensors within radius r_s of the event can rapidly cross their thresholds, while the sensors outside this range only observe noise. On the other hand with the power law path–loss sensing model, where the signal from the event varies continuously, in Example 2 (Section 6.2.2), the CUSUMs of some of the sensors beyond r_s also see a positive driving term. Lemma 1 shows that for a Gaussian noise model, there is a range $\bar{r} > r_s$ beyond which the driving term of the CUSUM has a negative mean.

Lemma 6.1 *For Gaussian noise (i.e., f_0 is Gaussian with mean μ_0 and variance σ^2), for some $r_s > 0$, let $f_1(\cdot; r_s)$ be taken as the postchange pdf in the* CUSUM *algorithm being used at a sensor. When the distance–loss function is $\rho(d)$, the mean of the driving term in the* CUSUM *statistic $C_k^{(i)}$ is* negative *if the event occurs beyond a distance of*

$$\overline{r} \quad := \quad \min\left\{d : 2\rho(d) \leqslant \rho(r_s)\right\}$$

from the sensor.

Proof 1 *Let $Z_k^{(i)}(d)$ be the LLR of the observation $X_k^{(i)}$ between pdfs $f_1(\cdot;d)$ and f_0, and let $Z_k^{(i)}(r_s)$ be the LLR between pdfs $f_1(\cdot;r_s)$ and f_0. It is easy to see for Gaussian f_0 (and hence for Gaussian $f_1(\cdot;\cdot)$) that*

$$\mathsf{E}_{f_1(\cdot;d)}[Z_k^{(i)}(r_s)] \quad = \quad \frac{(h_e\rho(r_s))^2}{2\sigma^2}\left(\frac{2\rho(d)}{\rho(r_s)} - 1\right). \tag{6.3}$$

Thus, $\mathsf{E}_{f_1(\cdot;d)}[Z_k^{(i)}(r_s)]$, the mean of the increment that drives CUSUM *statistic decreases with d and hits 0 at $\bar{r}$ given by $\bar{r} := \min\{d' : 2\rho(d') \leqslant \rho(r_s)\}$. Thus, $\mathsf{E}_{f_1(\cdot;d)}[Z_k^{(i)}(r_s)]$ is negative when an event occurs at a distance $d > \bar{r}$.*

We call this $\overline{r} := \min\{d : 2\rho(d) \leqslant \rho(r_s)\}$ the *sensing–range* of a sensor. Note that $\bar{r} > r_s$. Our viewpoint will be that a sensor can detect an event within an acceptable delay if it is within the detection–range, but that it can sense the event if it is within the sensing–range. In the case of the Boolean sensing model, the sensing range is the same as the detection–range, and is given by $\overline{r} = r_s$. In the case of the power law path–loss model, we can show that $\overline{r} = 2^{1/\eta}r_s$, which in the case of free–space (i.e., $\eta = 2$) is $\bar{r} = \sqrt{2}\, r_s$.

For $d > \bar{r}$ (i.e., the sensor is at a distance more than the sensing rage from the event), though the mean of the log–likelihood ratio $Z_k^{(i)}$ is negative, the CUSUM statistic $\{C_k^{(i)}\}$ (being a positive recurrent process) eventually crosses the CUSUM threshold c with probability 1 ([Meyn and Tweedie, 1993]). Let $\tau^{(i)} := \inf\left\{k : C_k^{(i)} \geqslant c\right\}$ be the time at which the CUSUM statistic $C_k^{(i)}$ crosses the threshold c. Let $T_E := \mathsf{E}_{f(\cdot;d)}\left[\tau^{(i)}\right]$ be the expected time that the statistic $C_k^{(i)}$ takes to cross the threshold c, where $\mathsf{E}_{f(\cdot,d)}[\cdot]$ is the expectation operator when the change happens at a distance d from sensor i and the distribution of all observations is $f(\cdot;d)$. We are interested in T_E when $d > \bar{r}$ which is shown in the lemma below.

Lemma 6.2 *For $d > \bar{r}$, $T_E \geqslant \exp(\omega_0 c)$, where $\omega_0 = 1 - \frac{2\rho(d)}{\rho(r_s)}$.*

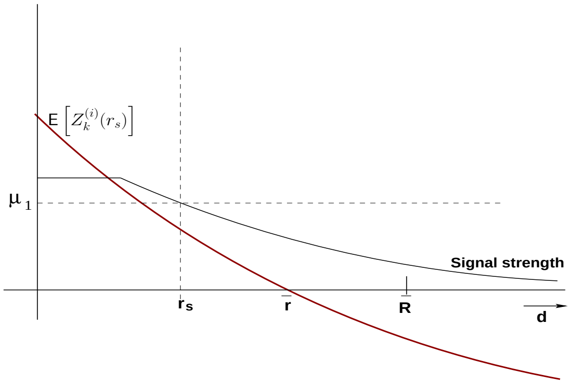


Figure 6.2: **Illustration of the detection range r_s, sensing range $\bar{r}$, and the influence range $\bar{R}$**: The signal strength of an event and the mean of the driving term of the CUSUM are plotted as a function of the distance d. At a distance $d \leqslant r_s$ from the event, the signal strength is at least μ_1, and for $d > \bar{r}$, the mean of the driving term $\mathsf{E}\left[Z_k^{(i)}(r_s)\right] < 0$.

Proof 2 *We recall that $\tau^{(i)}$ is the earliest time at which the* CUSUM *statistic $C_k^{(i)}$ crosses a threshold c. From (Eqn. 5.2.79 pg. 177 of) [Basseville and Nikiforov, 1993], we can show that $\mathsf{E}_{f_1(\cdot;d)}[\tau^{(i)}] \geqslant \exp(\omega_0 c)$ where ω_0 is the solution to the equation*

$$\mathsf{E}_{f_1(\cdot;d)}\left[e^{\omega_0 Z_k^{(i)}(r_s)}\right] = 0,$$

which is given by $\omega_0 = 1 - \frac{2\rho(d)}{\rho(r_s)}$ (see Eqn. (6.3)*).*

Note that from the definition of $\bar{r}$, $\omega_0 > 0$ for $d > \bar{r}$. We would be interested in $\mathsf{T_E} \geqslant \exp(\underline{\omega_0} \cdot c)$ for some $\underline{\omega_0} > 0$.

We now define the *influence range* of a sensor as follows.

Definition 6.2 Influence Range of a sensor, $\overline{R}$, is defined as the distance from the sensor within which the occurrence of an event can be detected within a mean delay of $\exp\left(\underline{\omega_0} c\right)$ where $\underline{\omega_0}$ is a parameter of interest and c is the threshold of the local CUSUM detector. Using Lemma 6.2, we see that $\bar{R} = \min\{d' : 2\rho(d') \leqslant (1 - \underline{\omega_0})\rho(r_s)\}$ (see Fig. 6.2 for an illustration). ■

From Lemma 6.2, we see that by having a large value of $\underline{\omega_0}$, the sensors that are beyond a distance of $\bar{R}$ from the event take a long time to cross the threshold. However, we see from the definition of influence range that a large value of $\underline{\omega}_0$ gives a large influence range $\bar{R}$. We will see from the discussion in Section 6.2.7 that a large influence range results in the isolation of the event to a large subregion of $\mathcal{A}$. On the other hand, from Section 6.4.6, we will see that a large $\underline{\omega}_0$ increases the time to false isolation, a performance metric of change detection/isolation procedure, which we define in Section 6.3.

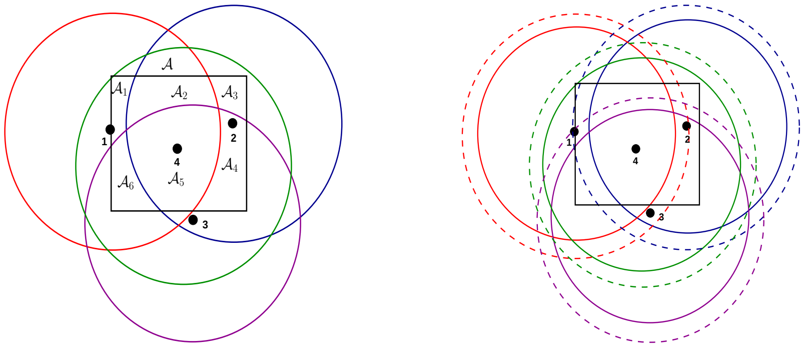
We define the *influence–region* of sensor i as $\mathcal{T}^{(i)} := \{x \in \mathcal{A} : \|\ell^{(i)} - x\| \leqslant \overline{R}\}$. For the Boolean sensing model, $\overline{R} = r_s$, and hence, $\mathcal{S}^{(j)} = \mathcal{T}^{(j)}$ for all $1 \leqslant j \leqslant n$, and for the power law path–loss sensing model, $\overline{R} > r_s$, and hence, $\mathcal{S}^{(j)} \subset \mathcal{T}^{(j)}$ for all $1 \leqslant j \leqslant n$.

Recalling the sets of sensors $\mathcal{N}_i$, $1 \leqslant i \leqslant N$, defined in Section 6.2.3, we define the *influence region of the set of sensors* $\mathcal{N}_i$ as the region $\mathcal{B}_i$ such that each $x \in \mathcal{B}_i$ is within the influence range of all the sensors in $\mathcal{N}_i$, i.e.,

$$\mathcal{B}_i \ := \ \mathcal{B}(\mathcal{N}_i) \quad := \quad \bigcap_{j \in \mathcal{N}_i} \mathcal{T}^{(j)}. \tag{6.4}$$

Note that $\mathcal{A}(\mathcal{N}_i) = \left(\bigcap_{j \in \mathcal{N}_i} \mathcal{S}^{(j)}\right) \bigcap \left(\bigcap_{j' \notin \mathcal{N}_i} \overline{\mathcal{S}^{(j')}}\right)$, where $\overline{\mathcal{S}}$ is the complement of the set $\mathcal{S}$, and $\mathcal{S}^{(j)} \subseteq \mathcal{T}^{(j)}$. Hence, $\mathcal{A}(\mathcal{N}_i) \subseteq \mathcal{B}(\mathcal{N}_i)$. For the power law path–loss sensing model, $\mathcal{S}^{(j)} \subset \mathcal{T}^{(j)}$ for all $1 \leqslant j \leqslant n$, and hence, $\mathcal{A}(\mathcal{N}_i) \subset \mathcal{B}(\mathcal{N}_i)$ for all $1 \leqslant i \leqslant N$. For the Boolean sensing model, $\mathcal{A}(\mathcal{N}_i) = \mathcal{B}(\mathcal{N}_i)$ only when $\mathcal{N}_i = \{1, 2, \cdots, n\}$. Thus, for a general sensing model, $\mathcal{A}(\mathcal{N}_i) \subseteq \mathcal{B}(\mathcal{N}_i)$. We note here that in the Boolean and the power law path loss models, an event which does not lie in the detection subregion of $\mathcal{N}_i$, but lies in its influence subregion (i.e., $\ell_e \in \mathcal{B}(\mathcal{N}_i) \setminus \mathcal{A}(\mathcal{N}_i)$) can be detected due to $\mathcal{N}_i$ because of the stochastic nature of the observations; in the power law path loss sensing model, this is also because of the difference in $\rho(d_{ei})$ between different sensors, where we recall that d_{ei} is the distance of the event from sensor i, and $\rho(\cdot)$ is the distance dependent loss function.

In Section 6.3, we formulate the problem of quickest detection of an event and *isolating the event to one of the influence subregions* $\mathcal{B}_1, \mathcal{B}_2, \cdots, \mathcal{B}_N$ under a false alarm and false isolation constraint.



(a) Detection and influence regions of the Boolean model

(b) Detection and influence regions of the power law path loss model

Figure 6.3: **Influence and detection regions**: A simple example of partitioning of $\mathcal{A}$ in a large WSN. The coloured solid circles around each sensor node denote their detection regions. The four sensor nodes, in the figure, divide the ROI, indicated by the square region, into regions $\mathcal{A}_1, \cdots, \mathcal{A}_6$ such that region $\mathcal{A}_i$ is detection–covered by a unique set of sensors $\mathcal{N}_i$. The dashed circles represent the influence regions. In the Boolean model, the influence region of a sensor coincides with its detection region.

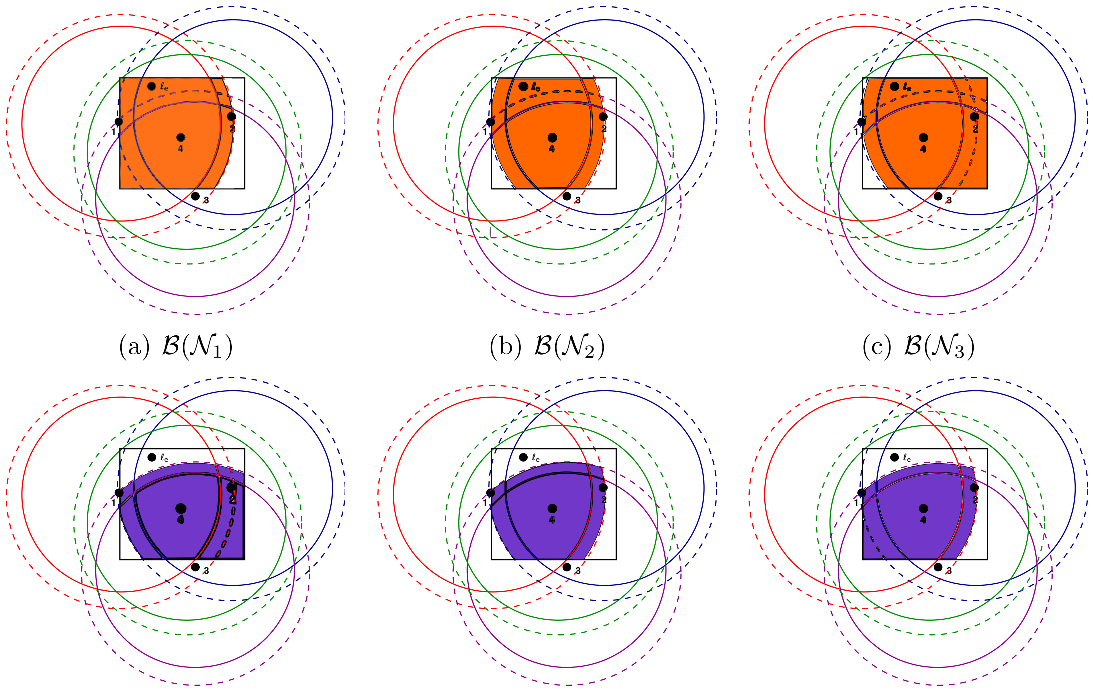
Remark: At this point, we recall that the definition of the three ranges, detection, sensing, and influence, have involved two design parameters μ_1 and $\underline{\omega_0}$ which can be used to "tune" the desired performance of the distributed detection schemes that we develop. ■

6.2.7 Discussion and Motivation for Formulation

We see from Example 1 in Section 6.2.2, that, in the case of the Boolean sensing model, the detection range is the same as the influence range, and hence, in Fig. 6.3(a), the influence region of each sensor coincides with its detection region. We consider an example with the same ROI (which is indicated by the square region) and the sensor deployment as in Fig. 6.3(a). The ROI is detection–partitioned by $\mathcal{N}_1 = \{1, 4\}$, $\mathcal{N}_2 = \{1, 2, 4\}$, $\mathcal{N}_3 = \{2, 4\}$, $\mathcal{N}_4 = \{2, 3, 4\}$, $\mathcal{N}_5 = \{1, 2, 3, 4\}$, and $\mathcal{N}_6 = \{1, 3, 4\}$. The event detection can be due to one of the sensor sets that detection cover the ROI, i.e., $\mathcal{N}_1$ or $\mathcal{N}_2$ or $\mathcal{N}_3$ or $\mathcal{N}_4$ or $\mathcal{N}_5$ or $\mathcal{N}_6$. We consider an event having occurred in subregion $\mathcal{A}_2$ (see Fig. 6.4; in the Boolean model, the dashed circles coincide with the corresponding solid ones), which is detection covered by $\mathcal{N}_2 = \{1, 2, 4\}$. We note here that $\mathcal{N}_2$ detection covers ℓ_e, the location of the event (i.e., $\ell_e \in \mathcal{A}(\mathcal{N}_2)$ and hence, $\ell_e \in \mathcal{B}(\mathcal{N}_2)$), and also that the sensor sets $\mathcal{N}_1$ and $\mathcal{N}_3$ are subsets of $\mathcal{N}_2$. As the distances of the event from different sensors are different and the observations $X_k^{(i)}$s are stochastic, it is possible that the event is detected by $\mathcal{N}_1$ or $\mathcal{N}_3$ (i.e., it might take a little longer time for all the sensors in $\mathcal{N}_2$ to detect the event). Since $\mathcal{N}_1 \subset \mathcal{N}_2$ (resp. $\mathcal{N}_3 \subset \mathcal{N}_2$), the influence subregion $\mathcal{B}(\mathcal{N}_1) \supset \mathcal{B}(\mathcal{N}_2)$ (resp. $\mathcal{B}(\mathcal{N}_3) \supset \mathcal{B}(\mathcal{N}_2)$). Thus, each of the influence subregions $\mathcal{B}(\mathcal{N}_1)$ and $\mathcal{B}(\mathcal{N}_3)$ contains the location of the event (see Fig. 6.4; recalling that, in the Boolean model, the dashed circles coincide with the corresponding solid ones), whereas the detection subregion $\mathcal{A}(\mathcal{N}_1)$ or $\mathcal{A}(\mathcal{N}_3)$ does not contain the location of the event. Hence, *we choose to isolate the event to the influence subregion of the sensor set that detects the event.* Thus, the detection due to $\mathcal{N}_1$ or $\mathcal{N}_2$ or $\mathcal{N}_3$ isolates the event (correctly) to the corresponding influence subregion. On the other hand, each of the sensor sets $\mathcal{N}_4$, $\mathcal{N}_5$, and $\mathcal{N}_6$ contains at least one sensor that does not influence cover ℓ_e, and hence, the detection due to $\mathcal{N}_4$ or $\mathcal{N}_5$ or $\mathcal{N}_6$ yields a *false isolation.*

In the case of the power law path–loss model (see Example 2 in Section 6.2.2), the influence range is strictly larger than the detection range, and hence, we have concentric discs around each sensor (the smaller one for the detection region and the larger one for the influence region, see Figs. 6.3(b) and 6.4). Let the event occur in the detection subregion of $\mathcal{N}_i$, i.e., $\ell_e \in \mathcal{A}(\mathcal{N}_i)$. In the path–loss model, we have the following two cases. In case 1, the location of the event does not belong to the influence region of any sensor $j \notin \mathcal{N}_i$ (i.e., $\ell_e \notin \mathcal{T}^{(j)}$ for all $j \notin \mathcal{N}_i$), and in case 2, the location of the event belongs to the influence region of some sensor $j \notin \mathcal{N}_i$ (i.e., $\ell_e \in \mathcal{T}^{(j)}$ for some $j \notin \mathcal{N}_i$). In the Boolean sensing model, as the influence region of a sensor coincides with its detection region, an event, which is detection covered by $\mathcal{N}_i$, can not lie in the influence region (which is also the detection region) of any other sensor $j \notin \mathcal{N}_i$, and hence, the case 2 described for the path loss model never arises in the Boolean model.

For the power law path–loss model, we consider the same detection/isolation example that we illustrated above for the Boolean sensing model. Also, we consider the case



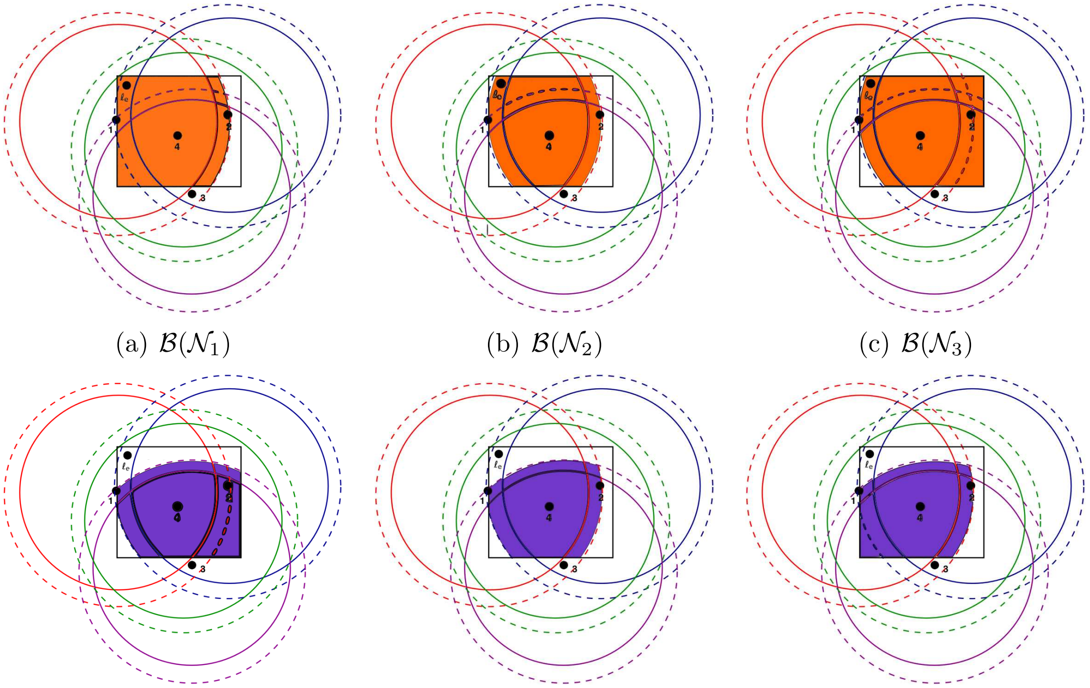
(a) $\mathcal{B}(\mathcal{N}_1)$ (b) $\mathcal{B}(\mathcal{N}_2)$ (c) $\mathcal{B}(\mathcal{N}_3)$

(d) $\mathcal{B}(\mathcal{N}_4)$ (e) $\mathcal{B}(\mathcal{N}_5)$ (f) $\mathcal{B}(\mathcal{N}_6)$

Figure 6.4: **An Illustration of the case: $\ell_e \in \mathcal{A}(\mathcal{N}_i)$ and $\ell_e \notin \mathcal{T}^{(j)}$, $\forall j \notin \mathcal{N}_i$**: The ROI, indicated by the square region, is detection–partitioned by $\mathcal{N}_1 = \{1,4\}$, $\mathcal{N}_2 = \{1,2,4\}$, $\mathcal{N}_3 = \{2,4\}$, $\mathcal{N}_4 = \{2,3,4\}$, $\mathcal{N}_5 = \{1,2,3,4\}$, and $\mathcal{N}_6 = \{1,3,4\}$. *In the Boolean sensing model, the influence region of a sensor represented by a dashed circle coincides with its detection region represented by a solid circle.* The set of sensors $\mathcal{N}_2$ detection covers the location of the event ℓ_e. If all the sensors in $\mathcal{N}_2$ or a subset of them as in $\mathcal{N}_1$ or $\mathcal{N}_3$ detect the event, then the corresponding influence region isolates the event (see Figs. 6.4(a)–6.4(c)). On the other hand, if the detection is due to $\mathcal{N}_i \not\subseteq \mathcal{N}_2$, then the influence region $\mathcal{B}(\mathcal{N}_i)$ does not contain the location of the event (see Figs. 6.4(d)–6.4(f)).

where the location of the event $\ell_e \in \mathcal{A}_2$ and $\ell_e \notin \mathcal{T}^{(3)}$, i.e., the event lies in the detection coverage region of $\mathcal{N}_2$ and does not lie in the influence region of sensor 3. In this case, the isolation region that corresponds to the detection due to each sensor set $\mathcal{N}_i$ can be explained in exactly the same manner as in the Boolean model, which we described above (see Fig. 6.4).

We consider an example for the case 2 of the power law path–loss model, where $\ell_e \in \mathcal{A}(\mathcal{N}_i)$ and $\ell_e \in \mathcal{T}^{(j)}$ for some $j \notin \mathcal{N}_i$. We consider the same ROI and the sensor deployment as in the previous example (i.e., the ROI is a square region and is detection–partitioned by $\mathcal{N}_1 = \{1,4\}$, $\mathcal{N}_2 = \{1,2,4\}$, $\mathcal{N}_3 = \{2,4\}$, $\mathcal{N}_4 = \{2,3,4\}$,



(a) $\mathcal{B}(\mathcal{N}_1)$ (b) $\mathcal{B}(\mathcal{N}_2)$ (c) $\mathcal{B}(\mathcal{N}_3)$

(d) $\mathcal{B}(\mathcal{N}_4)$ (e) $\mathcal{B}(\mathcal{N}_5)$ (f) $\mathcal{B}(\mathcal{N}_6)$

Figure 6.5: **An Illustration of the case: $\ell_e \in \mathcal{A}(\mathcal{N}_i)$ and $\ell_e \in \mathcal{T}^{(j)}$, for some $j \notin \mathcal{N}_i$ (corresponding to Fig. 6.3(b))**: The ROI, indicated by the square region, is detection–partitioned by $\mathcal{N}_1 = \{1,4\}$, $\mathcal{N}_2 = \{1,2,4\}$, $\mathcal{N}_3 = \{2,4\}$, $\mathcal{N}_4 = \{2,3,4\}$, $\mathcal{N}_5 = \{1,2,3,4\}$, and $\mathcal{N}_6 = \{1,3,4\}$. The set of sensors $\mathcal{N}_1$ detection covers the location of the event ℓ_e and $\{1,2,4\} = \mathcal{N}_2$ influence covers ℓ_e. If all the sensors in $\mathcal{N}_2$ or a subset of them as in $\mathcal{N}_1$ or $\mathcal{N}_3$ detect the event, then the corresponding influence region isolates the event (see Figs. 6.5(a)–6.5(c)). On the other hand, if the detection is due to $\mathcal{N}_i \not\subseteq \mathcal{N}_2$, then the influence region $\mathcal{B}(\mathcal{N}_i)$ does not contain the location of the event (see Figs. 6.5(d)–6.5(f)).

$\mathcal{N}_5 = \{1,2,3,4\}$, and $\mathcal{N}_6 = \{1,3,4\}$, see Fig. 6.3(b)). If $\ell_e \in \mathcal{A}_2$ and $\ell_e \in \mathcal{T}^{(3)}$ then ℓ_e lies in the influence region of all the sensors (a trivial example), and hence, the detection due to any $\mathcal{N}_i$ isolates the event (correctly) to the corresponding influence subregion. We consider another example where the ROI and the sensor sets are the same as in the previous example, and the location of the event $\ell_e \in \mathcal{A}(\mathcal{N}_1)$, $\ell_e \in \mathcal{T}^{(2)}$, and $\ell_e \notin \mathcal{T}^{(3)}$ (see Fig. 6.5). In this example, ℓ_e lies in the influence region of sensors $\{1,2,4\}$ and does not lie in the influence region of sensor 3. Hence, the detection due to the set of sensors $\mathcal{N}_1 = \{1,4\}$ or $\mathcal{N}_2 = \{1,2,4\}$ or $\mathcal{N}_3 = \{2,4\}$ (i.e., subsets of $\{1,2,4\}$ that influence cover ℓ_e) isolates the event (correctly) to the corresponding influence subregion, whereas the detection due to the set of sensors $\mathcal{N}_4 = \{2,3,4\}$ or $\mathcal{N}_5 = \{1,2,3,4\}$ or $\mathcal{N}_6 = \{1,3,4\}$ (sets which are not subsets of $\{1,2,4\}$) yields a false isolation.

6.3 Problem Formulation

We are interested in studying the *problem of distributed event detection/isolation* in the setting developed in Section 6.2. Given a sample node deployment (i.e., given $\boldsymbol{\ell}$), and *having chosen a value of the detection range,* r_s, we partition the ROI, $\mathcal{A}$ into the detection–subregions, $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_N$. Let $\mathcal{N}_i$ be the set of sensors that detection–cover the region $\mathcal{A}_i$. Let $\mathcal{B}_i$ be the influence region of the set of sensor nodes $\mathcal{N}_i$. We define the following set of hypotheses

$$\begin{aligned}\mathbf{H}_0 &: \text{ event not occurred,}\\ \mathbf{H}_i &: \text{ event occurred in subregion } \mathcal{A}_i, \quad i = 1,2,\cdots,N.\end{aligned}$$

The event occurs in one of the detection subregions $\mathcal{A}_i$, but we will only be able to isolate it to one of the influence subregions $\mathcal{B}_i$ that is consistent with the $\mathcal{A}_i$ (as explained in Section 6.2.7). We study distributed procedures that detect and locate an event (to any of the $\mathcal{B}_i$s) subject to a false alarm and false isolation constraint. The *false alarm constraint* considered is the mean time to false alarm $\mathsf{T}_{\mathsf{FA}i}$, and the *false isolation constraint* considered is the mean time to false isolation $\mathsf{T}_{\mathsf{FI}ij}$, each of which we define as follows.

Definition 6.3 For each $1 \leqslant i \leqslant N$, the **Mean time to false alarm** $\mathsf{T}_{\mathsf{FA}i}$ (due to sensors $\mathcal{N}_i$) is defined as the expected number of samples taken under the null hypothesis $\mathbf{H}_0$ to raise an alarm, and declare the hypothesis $\mathbf{H}_i \neq \mathbf{H}_0$ as true, i.e.,

$$\mathsf{T}_{\mathsf{FA}i} \;:=\; \mathsf{E}_\infty\left[\tau^{(\mathcal{N}_i)}\right],$$

where $\mathsf{E}_\infty[\cdot]$ is the expectation operator when the change occurs at infinity. ■

Definition 6.4 For each i, j, $1 \leqslant i \neq j \leqslant N$, the **Mean time to false isolation** $\mathsf{T}_{\mathsf{FI}ij}$, is defined as the expected number of samples taken under the hypothesis $\mathbf{H}_i \neq \mathbf{H}_0$ to raise an alarm, and declare the hypothesis $\mathbf{H}_j$ as true where $\mathcal{B}_j$, the influence subregion corresponding to $\mathbf{H}_j$, is not consistent with the subregion $\mathcal{A}_i$ in which the event actually occurred, i.e.,

$$\mathsf{T}_{\mathsf{FI}ij} \quad := \quad \sup_{\{\mathbf{s}:\ell_e \in \mathcal{A}_i\}} \mathsf{E}_1^{(\mathbf{s})} \left[\tau^{(\mathcal{N}_j)}\right],$$

where $\mathsf{E}_1^{(\mathbf{s})}[\cdot]$ is the expectation operator when the change occurs at time 1, and in the subregion $\mathcal{A}_i$ such that the vector of distances between the event and the sensor nodes is $\mathbf{s} = [s_1, s_2, \cdots, s_n]$. ■

In the classical change detection problem, there is one prechange hypothesis and only one postchange hypothesis (and hence, there is no question of a false isolation). The CUSUM procedure was proposed by Page in [Page, 1954] as a non–Bayesian change detection procedure. The optimality of CUSUM was shown by Lorden in [Lorden, 1971]. Lorden proposed a detection delay metric[3],

$$\mathsf{ADD}(\tau) := \sup_{t \geqslant 1} \operatorname{ess\,sup} \mathsf{E}_t \left[(\tau - t + 1)^+ | \mathbf{X}_{[1:t-1]}\right],$$

where ess sup (essential supremum) of a random variable X is defined as ess sup $X := \inf\{a : \mathsf{P}\{X > a\} = 0\}$, and E_t is the corresponding expectation operator corresponding to the probability measure P_t, when the change occurs at time t, conditioned on $\mathbf{X}_{[1:t-1]}$. Lorden showed that as $\gamma \to \infty$, $\mathsf{ADD}(\tau^{\mathsf{CUSUM}}) = \inf_{\{\tau:\mathsf{T}_{\mathsf{FA}}(\tau) \geqslant \gamma\}} \mathsf{ADD}(\tau)$, i.e., the CUSUM procedure (see Eqns. (6.1) and (6.2)) is asymptotically worst case delay optimal.

In this work, we are interested in obtaining a detection/isolation procedure τ (for a multihypothesis postchange problem) that defines a stopping time and the location of the event. The error events that can occur are the false alarm and the false isolation which are measured in terms of the mean time to false alarm $\mathsf{T}_{\mathsf{FA}i}$ and the mean time

[3] For $k_2 \geqslant k_1$, the notation $Y_{[k_1:k_2]}$ is defined as $[Y_{k_1}, Y_{k_1+1}, \cdots, Y_{k_2}]$.

to false isolation $\mathsf{T}_{\mathsf{FI}ij}$ (defined above). We define the supremum average detection delay SADD performance for the procedure τ in the same sense as Lorden [Lorden, 1971] (also see [Nikiforov, 1995]) as the worst–case average number of samples taken under any hypothesis $\mathbf{H}_i$, $i = 1, 2, \cdots, N$, to raise an alarm, i.e.,

$$\mathsf{SADD}(\tau) := \sup_{\{\mathbf{s}:\ell_e \in \mathcal{A}\}} \sup_{t \geqslant 1} \operatorname{ess\,sup} \ \mathsf{E}_t^{(\mathbf{s})} \left[(\tau - t + 1)^+ | \mathbf{X}_{[1:t-1]} \right],$$

where $\mathbf{s} := [s_1, s_2, \cdots, s_n]$, s_j being the distance between sensor j and the event. Note that $\mathsf{E}_t^{(\mathbf{s})}$ is the conditional expectation operator (and $\mathsf{P}_t^{(\mathbf{s})}$ is the corresponding probability measure) given $\mathbf{X}_{[1:t-1]}$ when the change happens at time t and at the location $\ell_e \in \mathcal{A}$ such that the distance between sensor j and the event is s_j. Thus, we are interested in obtaining an optimal procedure τ that minimises the SADD subject to the mean time to false alarm and the mean time to false isolation constraints,

$$\begin{array}{rl} \inf & \sup_{\{\mathbf{s}:\ell_e \in \mathcal{A}\}} \sup_{t \geqslant 1} \operatorname{ess\,sup} \ \mathsf{E}_t^{(\mathbf{s})} \left[(\tau - t + 1)^+ | \mathbf{X}_{[1:t-1]} \right] \\ \text{such that} & \mathsf{T}_{\mathsf{FA}i}(\tau) \geqslant \gamma, \quad i = 1, 2, \cdots, N \\ & \mathsf{T}_{\mathsf{FI}ij}(\tau) \geqslant \gamma, \quad i \neq j, \ i, j = 1, 2, \cdots, N. \end{array}$$

The change detection/isolation problem that we pose here is motivated by the framework of Nikiforov, [Nikiforov, 1995] which we discuss in the next subsection.

6.3.1 Centralised Solution for the Boolean Sensing Model

In [Nikiforov, 1995], Nikiforov studied a change detection/isolation problem that involves $N > 1$ postchange hypotheses (and one prechange hypothesis). Thus, Nikiforov's formulation can be applied to our problem. But, Nikiforov considered a model in which under hypothesis $\mathbf{H}_i$, the joint pdf of the observation vector $\mathbf{X}_k$, g_i is completely known. It should be noted that in our problem, in the case of power law path–loss sensing model, the pdf of the observations under any postchange hypothesis is unknown as the location of the event is unknown. The worst case detection delay of a multi–hypothesis change detection problem is given by $\sup_{1 \leqslant i \leqslant N} \sup_{t \geqslant 1} \operatorname{ess\,sup} \mathsf{E}_t^{(i)} \left[(\tau - t + 1)^+ | \mathbf{X}_{[1:t-1]} \right]$. The problem posed by Nikiforov is

$$\begin{aligned} \inf \quad & \sup_{1 \leqslant i \leqslant N} \sup_{t \geqslant 1} \operatorname{ess\,sup} \mathsf{E}_t^{(i)} \left[(\tau - t + 1)^+ | \mathbf{X}_{[1:t-1]} \right] \\ \text{such that} \quad & \mathsf{T}_{\mathsf{FA}i}(\tau) \geqslant \gamma, \quad i = 1, 2, \cdots, N \\ & \mathsf{T}_{\mathsf{FI}ij}(\tau) \geqslant \gamma, \quad i \neq j, \; i, j = 1, 2, \cdots, N \end{aligned} \tag{6.5}$$

where $\mathsf{T}_{\mathsf{FA}i}$ and $\mathsf{T}_{\mathsf{FI}ij}$ are as defined in Definitions 6.3 and 6.4. Nikiforov obtained an asymptotically optimum *centralised change detection/isolation* procedure $\tau^{\mathsf{Nikiforov}}$ when $\gamma \to \infty$, i.e., as $\gamma \to \infty$, $\mathsf{SADD}(\tau^{\mathsf{Nikiforov}}) = \inf_{\{\tau : \min\{\mathsf{T}_{\mathsf{FA}i}, \mathsf{T}_{\mathsf{FI}ij}\} \geqslant \gamma\}} \mathsf{SADD}(\tau)$. The SADD of the asymptotically optimal procedure $\tau^{\mathsf{Nikiforov}}$ is given by the following theorem.

Theorem 6.1 ([Nikiforov, 1995]) *For the N–hypotheses change detection/isolation problem (for the Boolean sensing model) defined in Eqn.* (6.5), *the asymptotically worst case delay optimal detection/isolation procedure $\tau^{\mathsf{Nikiforov}}$ has the property,*

$$\mathsf{SADD}(\tau^{\mathsf{Nikiforov}}) \quad \sim \quad \frac{\ln \gamma}{\min_{0 \leqslant i \leqslant N} \min_{1 \leqslant j \neq i \leqslant N} KL(g_i, g_j)}, \quad \text{as } \gamma \to \infty, \tag{6.6}$$

where $KL(f, g)$ is the Kullback–Leibler divergence between the pdfs f and g (and we recall that g_i is the joint pdf of the observation $\mathbf{X}_k$ under hypothesis $\mathbf{H}_i$). ■

We note here that the decision statistic of Nikiforov's procedure $\tau^{\mathsf{Nikiforov}}$ cannot be computed in a recursive manner, making the procedure computationally prohibitive.

In the case of Boolean sensing model, for any postchange hypothesis $\mathbf{H}_i$, after the change occurs, only the set of sensor nodes that detection cover (which is the same as influence cover) the subregion $\mathcal{A}_i$ switch to a postchange pdf f_1 (and the distribution of other sensor nodes continues to be f_0). Since the pdf of the sensor observations are conditionally i.i.d., the pdf of the observation vector, in the Boolean sensing model, corresponds to the postchange pdf g_i of Nikiforov's problem. Thus, the problem considered by Nikiforov directly applies to our setting with the Boolean sensing model. In our work, however, we propose algorithms for the change detection/isolation problem for any sensing model. Also, Nikiforov's procedure is centralised and computationally prohibitive, whereas we propose distributed procedures which are computationally simple.

In Section 6.4, we propose a distributed detection procedure HALL and analyse its false alarm (T_{FA}), false isolation (T_{FI}) and the detection delay (SADD) properties. We also discuss the false alarm (T_{FA}), false isolation (T_{FI}) and the detection delay (SADD) properties of the procedures MAX and ALL.

6.4 Distributed Change Detection/Isolation Procedures

In this section, we study the procedures MAX and ALL for change detection/isolation in a distributed setting. Also, we propose a distributed detection procedure "HALL" for a large WSN, and analyse the SADD, the T_{FA}, and the T_{FI} performance.

6.4.1 The MAX Procedure

Tartakovsky and Veeravalli proposed a decentralised procedure MAX for a collocated scenario in [Tartakovsky and Veeravalli, 2003]. We extend the MAX procedure to a large WSN under the T_{FA} and T_{FI} constraints. Recalling Section 6.2, each sensor node i employs CUSUM for local change detection between pdfs f_0 and $f_1(\cdot; r_s)$. Let $\tau^{(i)}$ be the random time at which the CUSUM statistic of sensor node i crosses the threshold c. The local decision of sensor node i, $D_k^{(i)}$ is defined as

$$D_k^{(i)} \ := \ \begin{cases} 0, & \text{for } k < \tau^{(i)} \\ 1, & \text{for } k \geqslant \tau^{(i)}. \end{cases}$$

The global decision rule τ^{MAX} declares an alarm at the earliest time slot k at which all sensor nodes $j \in \mathcal{N}_i$ for some $i = 1, 2, \cdots, N$ have crossed the threshold c. Thus,

$$\begin{aligned} \tau^{\mathsf{MAX},(\mathcal{N}_i)} \ &:= \ \max\left\{\tau^{(j)}, \ j \in \mathcal{N}_i\right\}, \\ \tau^{\mathsf{MAX}} \ &:= \ \min\left\{\tau^{\mathsf{MAX},(\mathcal{N}_i)} : 1 \leqslant i \leqslant N\right\}, \end{aligned}$$

i.e., the MAX procedure declares an alarm at the earliest time instant when the CUSUM statistic of all the sensor nodes $\mathcal{N}_i$ corresponding to hypothesis $\mathbf{H}_i$ of some i crosses the threshold at least once. Note that, when the alarm is due to the sensor nodes in the set $\mathcal{N}_i$, at the time of alarm τ^{MAX}, the CUSUM statistic of some nodes $j \in \mathcal{N}_i$ can be less than the threshold, but the CUSUM statistic of each of these nodes j has crossed the threshold at some time $k_j < \tau^{\mathsf{MAX}}$.

The isolation rule is to declare the event having occurred in the influence region $\mathcal{B}_i = \mathcal{B}(\mathcal{N}_i)$ corresponding to the set of sensors $\mathcal{N}_i$ that raised the alarm.

6.4.2 ALL Procedure

Mei, [Mei, 2005], and Tartakovsky and Kim, [Tartakovsky and Kim, 2006], proposed a decentralised procedure ALL, again for a collocated network. We extend the ALL procedure to a large extent network under the $\mathsf{T_{FA}}$ and the $\mathsf{T_{FI}}$ constraints. Here, each sensor node i employs CUSUM for local change detection between pdfs f_0 and $f_1(\cdot; r_s)$. Let $C_k^{(i)}$ be the CUSUM statistic of sensor node i at time k. *The* CUSUM *in the sensor nodes is allowed to run freely even after crossing the threshold c.* Here, the local decision of sensor node i is

$$D_k^{(i)} \quad := \quad \begin{cases} 0, & \text{if } C_k^{(i)} < c \\ 1, & \text{if } C_k^{(i)} \geqslant c. \end{cases}$$

The global decision rule τ^{ALL} declares an alarm at the earliest time slot k at which the local decision of all the sensor nodes corresponding to a set $\mathcal{N}_i$, for some $i = 1, 2, \cdots, N$, are 1, i.e.,

$$\begin{aligned} \tau^{\mathsf{ALL},(\mathcal{N}_i)} &:= \inf\left\{k : D_k^{(j)} = 1,\ \forall j \in \mathcal{N}_i\right\} \\ &= \inf\left\{k : C_k^{(j)} \geqslant c, \forall j \in \mathcal{N}_i\right\} \\ \tau^{\mathsf{ALL}} &:= \min\left\{\tau^{\mathsf{ALL},(\mathcal{N}_i)} : 1 \leqslant i \leqslant N\right\}. \end{aligned}$$

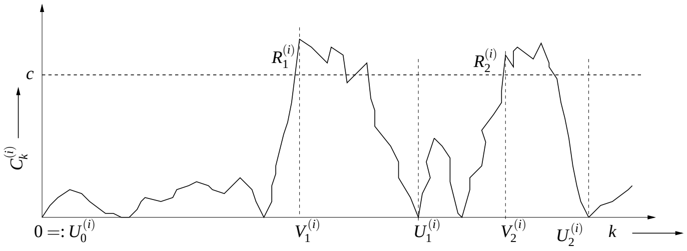


Figure 6.6: ALL and HALL: Evolution of CUSUM statistic $C_k^{(i)}$ of node i plotted vs. k. Note that at time $k = V_j^{(i)}$, $R_j^{(i)}$ is the excess above the threshold.

The isolation rule is to declare the event having occurred in region $\mathcal{B}_i = \mathcal{B}(\mathcal{N}_i)$ corresponding to the set of sensors $\mathcal{N}_i$ that raised the alarm. Note that ALL declares an alarm only when the CUSUM statistic of all the nodes $j \in \mathcal{N}_i$ are above the threshold.

6.4.3 HALL Procedure

Motivated by the fact that sensor noise can make the CUSUM statistic fluctuate around the threshold, we propose a local decision rule which is 0 when the CUSUM statistic has visited zero and has not crossed the threshold yet and is 1 otherwise. We explain the HALL procedure below.

The following discussion is illustrated in Fig. 6.6. Each sensor node i computes a CUSUM statistic $C_k^{(i)}$ based on the LLR of its own observations between the pdfs $f_1(\cdot; r_s)$ and f_0. Define $U_0^{(i)} := 0$. Define $V_1^{(i)}$ as the time at which $C_k^{(i)}$ crosses the threshold c (for the first time) as:

$$V_1^{(i)} := \inf\left\{k : C_k^{(i)} \geqslant c\right\}$$

(see Fig. 6.6 where the "overshoots" $R_k^{(i)}$, at $V_k^{(i)}$, are also shown). Note that $\inf \emptyset := \infty$. Next define

$$U_1^{(i)} := \inf\left\{k > V_1^{(i)} : C_k^{(i)} = 0\right\}.$$

Now starting with $U_1^{(i)}$, we can recursively define $V_2^{(i)}, U_2^{(i)}$ etc. in the obvious manner (see Fig. 6.6). We define the *quiet–times* and the *active–times* of the CUSUM process $C_k^{(i)}$ as $Q_j^{(i)} := V_j^{(i)} - U_{j-1}^{(i)}$ and $A_j^{(i)} := U_j^{(i)} - V_j^{(i)}$. Each node i computes the local decision $D_k^{(i)}$ based on the CUSUM statistic $C_k^{(i)}$ as follows:

$$D_k^{(i)} = \begin{cases} 1, & \text{if } V_j^{(i)} \leqslant k < U_j^{(i)} \text{ for some } j \\ 0, & \text{otherwise.} \end{cases} \tag{6.7}$$

The global decision rule[4] is a stopping time τ^{HALL} defined as the earliest time slot k at which all the sensor nodes in a region have a local decision 1, i.e.,

$$\begin{aligned} \tau^{\mathsf{HALL},(\mathcal{N}_i)} &:= \inf\left\{k : D_k^{(j)} = 1, \ \forall j \in \mathcal{N}_i\right\}, \\ \tau^{\mathsf{HALL}} &:= \min\left\{\tau^{\mathsf{HALL},(\mathcal{N}_i)} : 1 \leqslant i \leqslant N\right\}. \end{aligned}$$

The isolation rule is to declare the event having occurred in the region $\mathcal{B}(\mathcal{N}_i)$ corresponding to the set of sensors $\mathcal{N}_i$ which raised the alarm.

For the distributed procedures MAX, ALL, and HALL, we analyse the SADD performance in Section 6.4.4, the $\mathsf{T}_{\mathsf{FA}i}$ in Section 6.4.5 and the $\mathsf{T}_{\mathsf{FI}ij}$ in Section 6.4.6.

6.4.4 Supremum Average Detection Delay (SADD)

In this section, we analyse the SADD performance of the distributed detection/isolation procedures. We observe that for any sample path of the observation process, for a given threshold c, the MAX rule raises an alarm first, followed by the HALL rule and then by the ALL rule. This ordering is explained as follows. We note that for any node i, the local decision statistic of the MAX procedure, $D_k^{(i)} = 0$ for $k < \tau^{(i)}$ and $D_k^{(i)} = 1$ for $k \geqslant \tau^{(i)}$. It can happen that the CUSUM statistic $C_k^{(i)}$ at some $k > \tau^{(i)}$ goes below c, the CUSUM threshold, in which case, the local decision of the ALL rule is 0 (but that of MAX is 1). Let $V_{j_k}^{(i)} < k$ be the last time instant before k at which the CUSUM statistic of the ith

[4] The procedures $\mathsf{HALL}, \mathsf{MAX}$ and ALL differ only in their local decision rule; the global decision rule as a function of $D_k^{(i)}$s is the same for $\mathsf{HALL}, \mathsf{MAX}$ and ALL.

node crossed the threshold c. If at all time instants between (and including) $V_{j_k}^{(i)}$ and k, the CUSUM statistic is strictly greater than 0, then the local decision statistic of HALL is 1; otherwise, it is 0. Thus, the local decisions of MAX, HALL, and ALL are ordered as $D_k^{(i)}(\mathsf{MAX}) \geqslant D_k^{(i)}(\mathsf{HALL}) \geqslant D_k^{(i)}(\mathsf{ALL})$, and hence, we have,

$$\tau^{\mathsf{MAX}} \leqslant \tau^{\mathsf{HALL}} \leqslant \tau^{\mathsf{ALL}}. \tag{6.8}$$

We recall that each of the stopping times MAX, HALL, or ALL is the minimum of stopping times corresponding to the sets of sensors $\{\mathcal{N}_r : r = 1, 2, \cdots, N\}$, i.e.,

$$\begin{aligned} \tau^{\mathsf{rule}} &= \min\{\tau^{\mathsf{rule},(\mathcal{N}_r)} : r = 1, 2, \cdots, N\} \\ &\leqslant \tau^{\mathsf{rule},(\mathcal{N}_r)}, \text{ for any } r = 1, 2, \cdots, N \end{aligned}$$

where "rule" can be MAX or HALL or ALL.

Let $\mathbf{H}_i$ be the true hypothesis, i.e., the location of the event $\ell_e \in \mathcal{A}_i$. Since $\mathbf{H}_i$ is the true hypothesis, we have $\forall j \in \mathcal{N}_i,\ s_j \leqslant r_s$. Also, we have $\tau^{\mathsf{ALL}} \leqslant \tau^{\mathsf{ALL},(\mathcal{N}_i)}$. Hence,

$$\begin{aligned} \mathsf{SADD}\left(\tau^{\mathsf{ALL}}\right) &:= \sup_{\{\mathbf{s}:\ell_e \in \mathcal{A}_i\}} \sup_{t \geqslant 1} \operatorname{ess\,sup} \mathsf{E}_t^{(\mathbf{s})} \left[\left(\tau^{\mathsf{ALL}} - t + 1\right)^+ \mid \mathbf{X}_{[1:t-1]}\right] \\ &\leqslant \sup_{\{\mathbf{s}:\ell_e \in \mathcal{A}_i\}} \sup_{t \geqslant 1} \operatorname{ess\,sup} \mathsf{E}_t^{(\mathbf{s})} \left[(\tau^{\mathsf{ALL},(\mathcal{N}_i)} - t + 1)^+ \mid \mathbf{X}_{[1:t-1]}\right]. \end{aligned}$$

We note from [Tartakovsky and Kim, 2006], as the CUSUM threshold $c \to \infty$,

$$\begin{aligned} \sup_{t \geqslant 1} \operatorname{ess\,sup} \mathsf{E}_t^{(\mathbf{s})} \left[(\tau^{\mathsf{ALL},(\mathcal{N}_i)} - t + 1)^+ \mid \mathbf{X}_{[1:t-1]}\right] &= \frac{c}{\min_{j \in \mathcal{N}_i} \mathsf{E}_{f_1(\cdot;s_j)}\left[Z\left(s_j\right)\right]} \left(1 + o(1)\right), \\ &\leqslant \frac{c}{\mathsf{E}_{f_1(\cdot;r_s)}\left[Z\left(r_s\right)\right]} \left(1 + o(1)\right), \\ &= \frac{c}{\mathrm{KL}(f_1(\cdot;r_s), f_0)} \left(1 + o(1)\right). \end{aligned} \tag{6.9}$$

The inequality above follows from $\mathsf{E}_{f_1(\cdot;s_j)}\left[Z\left(s_j\right)\right] = \frac{(h_e \rho(s_j))^2}{2\sigma^2} \geqslant \frac{(h_e \rho(r_s))^2}{2\sigma^2} = \mathsf{E}_{f_1(\cdot;r_s)}\left[Z\left(r_s\right)\right]$ as $f_1(\cdot;d) \sim \mathcal{N}(h_e\rho(d) + \mu_0, \sigma^2)$, and $f_0 \sim \mathcal{N}(\mu_0, \sigma^2)$, $s_j \leqslant r_s\ \forall j \in \mathcal{N}_i$, and $\rho(d)$ decreases in d. Since the inequality in Eqn. (6.9) does not depend on i, we conclude that as $c \to \infty$,

$$\mathsf{SADD}(\tau^{\mathsf{ALL}}) \;\leqslant\; \frac{c}{\mathrm{KL}\big(f_1(\cdot;r_s),f_0\big)}(1+o(1)),$$

and from Eqn. (6.8), and the definition of SADD, we have

$$\mathsf{SADD}(\tau^{\mathsf{MAX}}) \;\leqslant\; \mathsf{SADD}(\tau^{\mathsf{HALL}}) \;\leqslant\; \mathsf{SADD}(\tau^{\mathsf{ALL}}) \;\leqslant\; \frac{c}{\mathrm{KL}(f_1(\cdot;r_s),f_0)}(1+o(1)), \text{ as } c \to \infty, \tag{6.10}$$

where $\mathrm{KL}(f_1(\cdot;r_s),f_0)$ is the Kullback–Leibler divergence between the pdfs $f_1(\cdot;r_s)$ and f_0.

Remark: Recall from Section 6.2.3 that $\mu_1 = h_e\rho(r_s)+\mu_0$. We now see that μ_1 governs the detection delay performance which can be chosen such that a requirement on SADD is met. Thus, to achieve a requirement on SADD, we need to choose r_s appropriately. A small value of r_s gives less detection delay compared to a large value of r_s. But, a small r_s requires more sensors to detection–cover the ROI.

6.4.5 Mean Time to False Alarm due to $\mathcal{N}_r$ ($\mathsf{T}_{\mathsf{FA}r}$)

From [Tartakovsky and Veeravalli, 2008], we note that for a CUSUM threshold c, the expected number of samples to raise a false alarm and locate the event in subregion $\mathcal{B}_r$ for the MAX procedure is given by

$$\mathsf{T}_{\mathsf{FA}r} \;\geqslant\; \exp(c). \tag{6.11}$$

Note that $\tau^{(r)}$, the time at which the last sensor in $\mathcal{N}_r$ crosses the threshold c, is at least as large as the stopping time of the CUSUM procedure at any sensor node $j \in \mathcal{N}_r$. As the mean time for each CUSUM to cross the threshold is $\exp(c)$ ([Basseville and Nikiforov, 1993]), $\mathsf{T}_{\mathsf{FA}r}$ of MAX procedure is at least as large as $\exp(c)$. Thus, for MAX procedure to achieve a $\mathsf{T}_{\mathsf{FA}r}$ of γ, we can choose the CUSUM threshold $c = \ln\gamma$.

From [Mei, 2005], the expected number of samples to raise a false alarm and locate the event in subregion $\mathcal{B}_r$ for the ALL procedure is given by

$$\mathsf{T}_{\mathsf{FA}r}\left(\tau^{\mathsf{ALL}}\right) \;\; \geqslant \;\; \exp\left(|\mathcal{N}_r|c\right). \tag{6.12}$$

In ALL, an alarm is made in $\mathcal{B}_r$ only when all the $|\mathcal{N}_r|$ CUSUM processes are above the threshold c. Since, these processes are independent, the probability of all sensor nodes $j \in \mathcal{N}_r$ in alarm state can be shown to be $1/\exp(|\mathcal{N}_r|c)$ (see [Mei, 2005]), and hence, the result follows. Thus, for ALL we choose $c \geqslant \frac{\ln\gamma}{|\mathcal{N}_r|}$ to achieve a $\mathsf{T}_{\mathsf{FA}r}$ of γ.

The $\mathsf{T}_{\mathsf{FA}r}$ performance of HALL is given by the following theorem.

Theorem 6.2 $\mathsf{T}_{\mathsf{FA}r}\left(\tau^{\mathsf{HALL}}\right) \;\geqslant\; \gamma$ *when the threshold* c *is chosen such that* $c\left(1-\frac{1}{\beta+o(1)}\right) \geqslant \frac{\ln(\gamma+1)}{|\mathcal{N}_r|} + \ln\frac{\beta+o(1)}{KL(f_0, f_1(\cdot;r_s))}$, *where* $\beta > 1$ *is a constant that depends on the pdfs* $f_0(\cdot), f_1(\cdot;r_s)$.

6.4.6 Mean Time to False Isolation ($\mathsf{T}_{\mathsf{FI}ij}$)

Here, we consider the scenario in which the hypothesis $\mathbf{H}_i$ is true and the hypothesis $\mathbf{H}_j$ is declared to be true at the time of alarm, *and* the event does not lie in the region $\mathcal{B}(\mathcal{N}_j)$. We are interested in finding the mean time to alarm due to $\mathcal{N}_j$ when $\ell_e \in \mathcal{B}(\mathcal{N}_i)$ and $\ell_e \notin \mathcal{B}(\mathcal{N}_j)$, i.e., there exists a node $j' \in \mathcal{N}_j$ such that $\ell_e \notin \mathcal{T}^{(j')}$, the influence region of node j'. We are interested in obtaining the corresponding mean time to false isolation, $\mathsf{T}_{\mathsf{FI}ij}$ which we show in the following theorem.

Theorem 6.3 *Define the parameters* λ_{ij} *as follows: for the Boolean sensing model,* $\lambda_{ij} = |\mathcal{N}_j \setminus \mathcal{N}_i|$ *and for the path–loss sensing model,* $\lambda_{ij} = 1$. *The* $\mathsf{T}_{\mathsf{FI}ij}$ *for the change detection/isolation procedures is given by*

1. $\mathsf{T}_{\mathsf{FI}ij}\left(\tau^{\mathsf{MAX}}\right) \geqslant \gamma$, *when the threshold* c *is chosen such that* $c \geqslant \frac{\ln\gamma}{\underline{\omega_0}}$.

2. $\mathsf{T}_{\mathsf{FI}ij}\left(\tau^{\mathsf{ALL}}\right) \geqslant \gamma$, *when the threshold* c *is chosen such that* $c \geqslant \frac{\ln\gamma}{\underline{\omega_0}\lambda_{ij}}$.

3. $\mathsf{T}_{\mathsf{FI}ij}\left(\tau^{\mathsf{HALL}}\right) \;\geqslant\; \gamma$, *when the threshold* c *is chosen such that* $c\left(1-\frac{1}{\beta+o(1)}\right) \;\geqslant\; \frac{\ln(\gamma+1)}{\underline{\omega_0}\lambda_{ij}} + \ln\frac{\beta+o(1)}{\underline{\omega_0}KL(f_0, f_1(\cdot;r_s))}$, *where* $\beta > 1$ *is a constant that depends upon the distribution of the LLR of the observation.*

where we recall that $\underline{\omega_0}$ is the parameter we defined in Section 6.2.6 that defines the lower bound on the mean time to cross the threshold for sensors beyond a distance of $\bar{R}$ from the event.

In the next subsection, we discuss the asymptotic minimax delay optimality of the distributed procedures in relation to Theorem 6.1.

6.4.7 Asymptotic Order Optimality

Define the parameter $\widetilde{M}$ as

$$\widetilde{M} \;=\; \min_{1\leqslant i\leqslant N}\;\min_{1\leqslant j\neq i\leqslant N}\;\min\left\{|\mathcal{N}_i|, \lambda_{ij}\right\}.$$

Choosing the local CUSUM threshold, $c = \frac{\ln\gamma}{\underline{\omega_0}\widetilde{M}}$ for ALL, $c = \frac{\ln(\gamma+1)}{\left(1-\frac{1}{\beta}\right)\underline{\omega_0}\widetilde{M}} + \frac{1}{1-\frac{1}{\beta}}\ln\frac{\beta}{\underline{\omega_0}\mathrm{KL}(f_0,f_1(\cdot;r_s))}$ for HALL, and $c = \frac{\ln\gamma}{\underline{\omega_0}}$ for MAX, we see from Eqns. (6.11) and (6.12), and Theorems 6.2 and 6.3, that as $\gamma\to\infty$, $\min\{\mathsf{T}_{\mathsf{FA}i}(\tau^{\mathsf{rule}}), \mathsf{T}_{\mathsf{FA}ij}(\tau^{\mathsf{rule}})\} \;\geqslant\; \gamma$ for ALL, HALL, and MAX. From Eqn. (6.9), we see that

$$\begin{aligned}
\mathsf{SADD}(\tau^{\mathsf{ALL}}) &\leqslant \frac{\ln\gamma}{\underline{\omega_0}\widetilde{M}\cdot\mathrm{KL}(f_1(\cdot;r_s),f_0)}\left(1+o(1)\right),\\
\mathsf{SADD}(\tau^{\mathsf{HALL}}) &\leqslant \frac{\ln(\gamma+1)}{\left(1-\frac{1}{\beta}\right)\underline{\omega_0}\widetilde{M}\cdot\mathrm{KL}(f_1(\cdot;r_s),f_0)}\left(1+o(1)\right)+C+o(1),\\
\mathsf{SADD}(\tau^{\mathsf{MAX}}) &\leqslant \frac{\ln\gamma}{\underline{\omega_0}\cdot\mathrm{KL}(f_1(\cdot;r_s),f_0)}\left(1+o(1)\right),
\end{aligned}\tag{6.13}$$

where $o(1)\to 0$ as $\gamma\to\infty$ and the constant C is given by $C = \frac{\ln(\beta)}{\left(1-\frac{1}{\beta}\right)\underline{\omega_0}\cdot\mathrm{KL}(f_0,f_1(\cdot;r_s))\cdot\mathrm{KL}(f_1(\cdot;r_s),f_0)}$. Note that as we decrease the detection–range r_s, $\mathrm{KL}(f_1(\cdot;r_s),f_0)$ and $\underline{\omega_0}$ increases. But $\widetilde{M}$ decreases as r_s decreases. Thus, to achieve a smaller detection delay, the detection range r_s can be decreased, and the number of sensors n can be increased.

We can compare the asymptotic SADD performance of the distributed procedures HALL, MAX and ALL against the optimal centralised scheme of Nikiforov for the Boolean sensing model. We recall that in the Boolean sensing model, the signal strength of the event remains a constant in a disk of radius r_s (which is the same as $\bar{R}$), and is 0 beyond the detection range. Hence, any sensor that is not in the detection range of the event observes only the sensor noise, and hence, the mean time for its CUSUM statistic to cross a threshold c is e^c, thus giving $\underline{\omega_0} = 1$. For Gaussian pdfs f_0 and f_1, the KL divergence between the hypotheses $\mathbf{H}_i$ and $\mathbf{H}_j$ is given by

$$\begin{aligned}
\mathrm{KL}(g_i, g_j) &= \int \ln\left(\frac{\prod_{s\in\mathcal{N}_i} f_1(x^{(s)}) \prod_{s'\notin\mathcal{N}_i} f_0(x^{(s')})}{\prod_{s\in\mathcal{N}_j} f_1(x^{(s)}) \prod_{s'\notin\mathcal{N}_j} f_0(x^{(s')})}\right) \prod_{s\in\mathcal{N}_i} f_1(x^{(s)}) \prod_{s'\notin\mathcal{N}_i} f_0(x^{(s')})\, d\mathbf{x} \\
&= \int \left(\ln\left(\prod_{s\in\mathcal{N}_i} \frac{f_1(x^{(s)})}{f_0(x^{(s)})}\right) - \ln\left(\prod_{s\in\mathcal{N}_j} \frac{f_1(x^{(s)})}{f_0(x^{(s)})}\right)\right) \prod_{s\in\mathcal{N}_i} f_1(x^{(s)}) \prod_{s'\notin\mathcal{N}_i} f_0(x^{(s')})\, d\mathbf{x} \\
&= \sum_{s\in\mathcal{N}_i} \mathrm{KL}(f_1, f_0) - \int \ln\left(\prod_{s\in\mathcal{N}_j\cap\mathcal{N}_i} \frac{f_1(x^{(s)})}{f_0(x^{(s)})}\right) \prod_{s\in\mathcal{N}_i} f_1(x^{(s)}) \prod_{s'\notin\mathcal{N}_i} f_0(x^{(s')})\, d\mathbf{x} \\
&\quad - \int \ln\left(\prod_{s\in\mathcal{N}_j\setminus\mathcal{N}_i} \frac{f_1(x^{(s)})}{f_0(x^{(s)})}\right) \prod_{s\in\mathcal{N}_i} f_1(x^{(s)}) \prod_{s'\notin\mathcal{N}_i} f_0(x^{(s')})\, d\mathbf{x} \\
&= \sum_{s\in\mathcal{N}_i} \mathrm{KL}(f_1, f_0) - \sum_{s\in\mathcal{N}_j\cap\mathcal{N}_i} \mathrm{KL}(f_1, f_0) + \sum_{s\in\mathcal{N}_j\setminus\mathcal{N}_i} \mathrm{KL}(f_1, f_0) \\
&= |\mathcal{N}_i \,\Delta\, \mathcal{N}_j| \;\mathrm{KL}(f_1, f_0)
\end{aligned}$$

where the operator Δ represents the symmetric difference between the sets. Thus, from Theorem 6.1 for Gaussian f_0 and f_1, we have

$$\begin{aligned}
\mathsf{SADD}(\tau^{\mathsf{Nikiforov}}) &\sim \frac{\ln\gamma}{M^* \cdot \mathrm{KL}(f_1, f_0)} \\
\text{where } M^* &= \min_{1\leqslant i\leqslant N} \min_{1\leqslant j\leqslant N, j\neq i} \min\left\{|\mathcal{N}_i|, |\mathcal{N}_i \,\Delta\, \mathcal{N}_j|\right\}.
\end{aligned}$$

Note that for the same $\min\{\mathsf{T}_{\mathsf{FA}}, \mathsf{T}_{\mathsf{FI}}\}$ requirement of γ, the SADD of the asymptotically optimum centralised procedure τ^*, and τ^{ALL} and τ^{HALL} scale as $\ln\gamma/\mathrm{KL}(f_1, f_0)$. Hence, ALL and HALL are asymptotically order optimal. The factor $1/(1-1/\beta)$ in the SADD of HALL makes this slightly larger than $\mathsf{SADD}(\tau^{\mathsf{ALL}})$. To achieve the desired false alarm performance for the MAX procedure, we choose the threshold $c = \ln\gamma$ and hence $\mathsf{SADD}\left(\tau^{\mathsf{MAX}}\right) \sim \frac{\ln\gamma}{\mathrm{KL}(f_1(\cdot; r_s), f_0)}$. Thus, the SADD performance of MAX is worse than that of ALL or HALL. Note that $\widetilde{M} \leqslant M^*$. This is because $|\mathcal{N}_j \setminus \mathcal{N}_i| \leqslant |\mathcal{N}_j \Delta \mathcal{N}_i|$. In the case of centralised setting, the observations from sensor nodes in $\mathcal{N}_j \setminus \mathcal{N}_i$ and in $\mathcal{N}_j \setminus \mathcal{N}_i$ can provide the location information of the event in a better way. In a

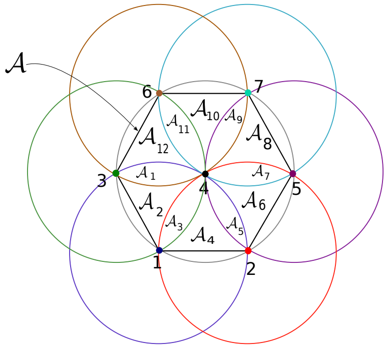


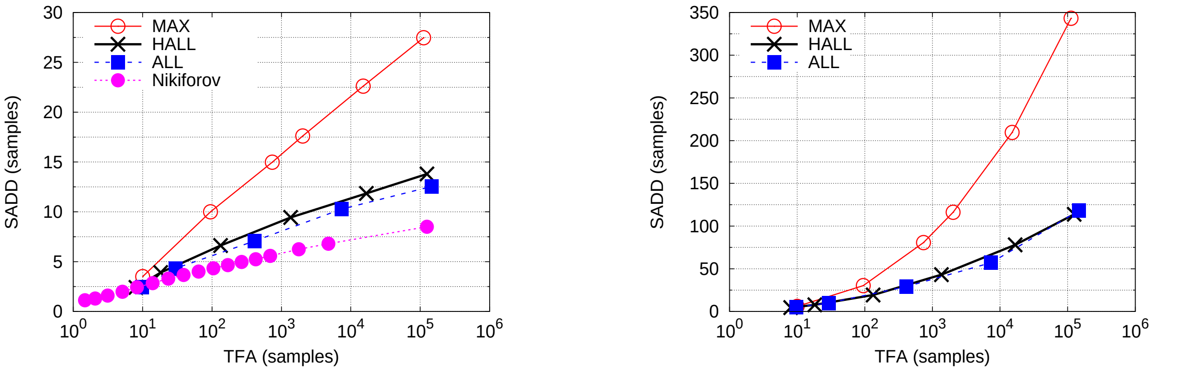
Figure 6.7: **Sensor nodes placement**: 7 sensor nodes (which are numbered 1,2,$\cdots$,7) represented by small filled circles are placed in the hexagonal ROI $\mathcal{A}$. The ROI is detection partitioned by the sets of sensor nodes, $\mathcal{N}_1 = \{1,3,4,6\}, \mathcal{N}_2 = \{1,3,4\}, \mathcal{N}_3 = \{1,2,3,4\}, \mathcal{N}_4 = \{1,2,4\}, \mathcal{N}_5 = \{1,2,4,5\}, \mathcal{N}_6 = \{2,4,5\}, \mathcal{N}_7 = \{2,4,5,7\}, \mathcal{N}_8 = \{4,5,7\}, \mathcal{N}_9 = \{4,5,6,7\}, \mathcal{N}_{10} = \{4,6,7\}, \mathcal{N}_{11} = \{3,4,6,7\}$, and $\mathcal{N}_{12} = \{3,4,6\}$ with the subregion $\mathcal{A}_i$ being the detection region (for both the Boolean and the power law path loss sensing models) of the sensor set $\mathcal{N}_i$.

distributed setting, this information is not available, and hence, the $\mathsf{T}_{\mathsf{FI}ij}$ performance of the distributed procedures is a little worse than that of the asymptotically optimum centralised procedure. To compensate for the decrease in $\mathsf{T}_{\mathsf{FI}ij}$, we increase the CUSUM threshold, which increases the SADD of the distributed procedures. But, the decrease in SADD of the centralised procedure is offset by substantially larger computation and communication costs, particularly in terms of the node energy expenditure, a critical issue in the context of WSNs.

6.5 Numerical Results

We compute the SADD and the $\mathsf{T}_{\mathsf{FA}i}$ performance of MAX, HALL, ALL, and Nikiforov's procedure for the Boolean sensing model with $f_0 \sim \mathcal{N}(0,1)$ and $f_1 \sim \mathcal{N}(1,1)$. We consider a deployment of 7 nodes in a hexagonal ROI (the detection region of each node being a disc of unit radius around it, see Fig. 6.7) such that we get $N = 12$ detection subregions, and $\mathcal{N}_1 = \{1,3,4,6\}, \mathcal{N}_2 = \{1,3,4\}, \mathcal{N}_3 = \{1,2,3,4\}, \mathcal{N}_4 = \{1,2,4\}, \mathcal{N}_5 = \{1,2,4,5\}, \mathcal{N}_6 = \{2,4,5\}, \mathcal{N}_7 = \{2,4,5,7\}, \mathcal{N}_8 = \{4,5,7\}, \mathcal{N}_9 = \{4,5,6,7\}, \mathcal{N}_{10} = \{4,6,7\}, \mathcal{N}_{11} = \{3,4,6,7\}$, and $\mathcal{N}_{12} = \{3,4,6\}$. From Eqns. (6.11) and (6.12), and from Theorem 6.2, it is clear that for a given threshold c, the $\mathsf{T}_{\mathsf{FA}i}$ is determined by $|\mathcal{N}_i|$, the number of sensors in $\mathcal{N}_i$. Thus, $\mathsf{T}_{\mathsf{FA}1} = \mathsf{T}_{\mathsf{FA}3} = \mathsf{T}_{\mathsf{FA}5} = \mathsf{T}_{\mathsf{FA}7} = \mathsf{T}_{\mathsf{FA}9} = \mathsf{T}_{\mathsf{FA}11}$, and $\mathsf{T}_{\mathsf{FA}2} = \mathsf{T}_{\mathsf{FA}4} = \mathsf{T}_{\mathsf{FA}6} = \mathsf{T}_{\mathsf{FA}8} = \mathsf{T}_{\mathsf{FA}10} = \mathsf{T}_{\mathsf{FA}12}$. Also (from Eqns. (6.11) and (6.12), and from Theorem 6.2), it is clear that larger the number of sensors in a sensor set, higher will be the T_{FA}. Hence, the T_{FA} performance of the change detection procedures are limited by the three sensor sets ($\mathsf{T}_{\mathsf{FA}2}$, $\mathsf{T}_{\mathsf{FA}4}$, $\mathsf{T}_{\mathsf{FA}6}$, $\mathsf{T}_{\mathsf{FA}8}$, $\mathsf{T}_{\mathsf{FA}10}$, $\mathsf{T}_{\mathsf{FA}12}$). Hence, we obtain the SADD and the $\mathsf{T}_{\mathsf{FA}i}$ (due to any of the 3 sensor sets) by simulation. For a detection/isolation procedure, we choose a range of the threshold c such that the $\mathsf{T}_{\mathsf{FA}i}$ varies from 10^0 to 10^5. For each threshold c of the local CUSUM chosen, we obtained the time to false alarm and the detection delay for 100 runs of the simulation, and took the average of these to be the $\mathsf{T}_{\mathsf{FA}i}$ and the SADD respectively. We note here that all the observations are sampled from Gaussian distribution with mean 0 and variance 1 for generating the $\mathsf{T}_{\mathsf{FA}i}$, and Gaussian distribution with mean 1 and variance 1 for generating the SADD (this assumes that the event has occurred at time 1, which corresponds to the worst case delay, see [Lorden, 1971]). We plot the SADD against the corresponding $\log(\mathsf{T}_{\mathsf{FA}i})$ in Fig. 6.8(a). We observe from Fig. 6.8(a) that SADD increases linearly with $\log(\mathsf{T}_{\mathsf{FA}i})$ and that the slope is $\approx 1/\mathrm{KL}(f_1, f_0)$ for MAX, and $\approx 1/(3 \cdot \mathrm{KL}(f_1, f_0))$ for HALL, ALL, and Nikiforov's procedure. Also, we observe that for a given $\mathsf{T}_{\mathsf{FA}i}$, Nikiforov's procedure has the smallest SADD and MAX has the largest SADD. For example, for a $\mathsf{T}_{\mathsf{FA}i}$ requirement of 2000 slots, Nikiforov's procedure gives an SADD of 6.2 slots, ALL gives an SADD of 8 slots, HALL gives an SADD of 9 slots, and MAX gives an SADD of 17.5 slots. Nikiforov's procedure is an optimal centralised procedure and hence it outperforms ALL, HALL and MAX in terms of SADD. Also, the SADD of MAX is the largest as it does not scale with $\widetilde{M}$. From Eqn. (6.13), it is clear that the SADD of ALL is little smaller than that of HALL.

For the same sensor deployment in Fig. 6.7, we compute the SADD and the T_{FA} for the square law path loss sensing model given in Section 6.2.2. Also, the sensing radius r_s and the signal strength h_e are taken to be unity. Thus, the sensor sets ($\mathcal{N}_i$s) and the detection subregions ($\mathcal{A}_i$s) are the same as in the Boolean model, we described above.



(a) SADD vs $\mathsf{T_{FA}}$ for the Boolean model

(b) SADD vs $\mathsf{T_{FA}}$ for the square law path loss model

Figure 6.8: SADD versus $\mathsf{T_{FA}}$ for MAX, HALL, ALL and Nikiforov's procedure for the Boolean and the square law path loss sensing models. In the Boolean sensing model, the system parameters are $f_0 \sim N(0,1)$, $f_1 \sim N(0,1)$, and in the case of path loss sensing model, the parameters are $f_0 \sim N(0,1)$, $r_s = 1.0$, $\bar{R} = 1.5$.

The prechange pdf f_0 is taken to be $\mathcal{N}(0,1)$, and since r_s is taken as 1, $f_1(\cdot; r_s) \sim \mathcal{N}(1,1)$. Thus, the LLR of observation $X_k^{(i)}$ is given by $\log\left(\frac{f_1(X_k^{(i)}; r_s)}{f_0(X_k^{(i)})}\right) = X_k^{(i)} - \frac{1}{2}$, which is the same as that in the Boolean sensing model. Hence, under *the event not occurred hypothesis*, as the pdf of all $X_k^{(i)}$s is $f_0(\cdot)$, the $\mathsf{T}_{\mathsf{FA}i}$s under the path loss sensing model is the same as that of the Boolean sensing model. We consider an event occurring in a detection subregion $\mathcal{A}_i$. Let the event be detected by $\mathcal{N}_j$ which influence covers the event, i.e., $\ell_e \in \mathcal{B}(\mathcal{N}_j)$. We assume a worst case distance between the event and sensors $s \in \mathcal{N}_j$, i.e., $d_{es} = \bar{R}$, the influence range. The influence range $\bar{R}$ is taken to be 1.5 (i.e., the influence region of a sensor overlaps with almost half of the detection region of each adjacent sensor). Due to the symmetry in the sensor locations (also in detection and influence subregions), we obtain the SADD and the $\mathsf{T}_{\mathsf{FA}i}$ due to any of the 3 sensor sets by simulation. For a distributed detection/isolation procedure, we choose a range of the threshold c such that the $\mathsf{T}_{\mathsf{FA}i}$ varies from 10^0 to 10^5. For each threshold c of the local CUSUM chosen, we obtained the time to false alarm and the detection delay for 100 runs of the simulation, and took the average of these to be the $\mathsf{T}_{\mathsf{FA}i}$ and the SADD respectively. We note here that all the observations are sampled from Gaussian distribution with mean 0 and variance 1 for generating the $\mathsf{T}_{\mathsf{FA}i}$, and Gaussian distribution with mean 1 and variance 1 for generating the SADD (this assumes that the event has occurred at time 1, which corresponds to the worst case delay, see [Lorden, 1971]). Also, we note that the location of the event is at a distance of $\bar{R}$ from all the nodes of $\mathcal{N}_j$ that influence covers the event. We plot the SADD against the corresponding $\log(\mathsf{T}_{\mathsf{FA}i})$ in Fig. 6.8(b). We observe from Fig. 6.8(b) that for large $\mathsf{T}_{\mathsf{FA}i}$, SADD increases linearly with $\log(\mathsf{T}_{\mathsf{FA}i})$. Also, we observe that for a given $\mathsf{T}_{\mathsf{FA}i}$, the smallest SADD is achieved by both ALL and HALL, and MAX has the largest SADD. For example, for a $\mathsf{T}_{\mathsf{FA}i}$ requirement of 2000 slots, ALL and HALL gives an SADD of 40 slots, and MAX gives an SADD of 110 slots. Thus, for a T_{FA} requirement of 2000 slots, the SADD performance is poorer in path loss model by 30 slots compared to the Boolean model in the case of ALL and HALL, and the SADD performance is worse by 92.5 slots in the case of MAX procedure. This is the price we pay for the uncertainty about the location of the event.

6.6 Conclusion

We consider the quickest distributed event detection/isolation problem in a large extent WSN with a practical sensing model which incorporates distance losses. We formulate the change detection/isolation problem in the optimality framework introduced by Nikiforov [Nikiforov, 1995]. We propose distributed detection/isolation procedures, MAX, ALL and HALL and show that as $\min\{\mathsf{T}_{\mathsf{FA}}, \mathsf{T}_{\mathsf{FI}}\} \to \infty$, the SADD performance of HALL and ALL is of the same order as that of the optimal centralised procedure of Nikiforov [Nikiforov, 1995].

6.7 Appendix

Proof of Theorem 6.2

We recall in the case of Boolean sensing model that the detection and influence subregions corresponding to a set of sensor nodes is the same. Note that for any $i = 1, 2, \cdots, n$, $\{U_j^{(i)}, j \geqslant 1\}$ is a renewal process, with the jth cycle being the time interval $(U_{j-1}^{(i)}, U_j^{(i)}]$.

Let $A_j^{(i)}$ be the reward in the jth cycle. By renewal theory, we can show that the probability of a node in local alarm state is given by $\frac{\mathsf{E}_\infty A_1^{(1)}}{\mathsf{E}_\infty A_1^{(1)}+\mathsf{E}_\infty Q_1^{(1)}}$ and, because the CUSUMs of nodes $i \in \mathcal{N}_r$ are independent, probability of false alarm in region $\mathcal{A}_r$ is

$$\mathsf{PFA_r} \;=\; \left(\frac{\mathsf{E}_\infty A_1^{(1)}}{\mathsf{E}_\infty A_1^{(1)} + \mathsf{E}_\infty Q_1^{(1)}}\right)^{|\mathcal{N}_r|}. \tag{6.14}$$

Let $\tau_1 < \tau_2 < \cdots$ be the random times at which the system enters false alarm in region $\mathcal{A}_r$ and let $0 =: \zeta_0 < \zeta_1 < \zeta_2 < \cdots$ be the random times at which the system comes out of the false alarm in region $\mathcal{A}_r$. We define TTF_r, the mean time to false alarms in $\mathcal{A}_r$, as

$$\mathsf{TTF}_r \;:=\; \lim_{m\to\infty} \frac{1}{m} \sum_{j=1}^{m} \left[\tau_j - \zeta_{j-1}\right].$$

We now show from sample path and coupling arguments that $\mathsf{TTF}_r \leqslant \mathsf{T}_{\mathsf{FA}r}$. At the times $\zeta_1, \zeta_2, \cdots$ at which the system comes out of the false alarm in region $\mathcal{A}_r$, the CUSUM statistic of some of the sensor nodes is non–zero. Thus, the time between the alarm τ_j and ζ_{j-1} is less than or equal to the mean time to false alarm $\mathsf{T}_{\mathsf{FA}r}$, the time required for a alarm when the CUSUM statistic of all the sensor nodes are reset to zero. Note that PFA_r, the fraction of time the system is in false alarm in region $\mathcal{A}_r$ is given by

$$\begin{aligned}
\mathsf{PFA}_r \;&=\; \lim_{m\to\infty} \frac{\sum_{j=1}^{m}\left[\zeta_j - \tau_j\right]}{\sum_{j=1}^{m}\left[(\tau_j - \zeta_{j-1}) + (\zeta_j - \tau_j)\right]} \\
&\geqslant\; \lim_{m\to\infty} \frac{m}{\sum_{j=1}^{m}\left[\tau_j - \zeta_{j-1}\right] + m} \\
&=\; \lim_{m\to\infty} \frac{1}{\frac{1}{m}\sum_{j=1}^{m}\left[\tau_j - \zeta_{j-1}\right] + 1} \qquad (6.15) \\
&=\; \frac{1}{\mathsf{TTF}_r + 1} \qquad (6.16)
\end{aligned}$$

The inequality in the second step is due to the fact that the time duration the system spends in false alarm state is at least one step and the function $g : \{1, 2, 3, \cdots\} \to \mathbb{R}$ defined by $g(x) := \frac{x}{K+x}$ is an increasing function, for a constant K. Therefore, combining

Eqn. (6.14) and Eqn. (6.15), we have

$$\begin{aligned}
\mathsf{TTF}_r &\geqslant \left(\frac{\mathsf{E}_\infty\left[A_1^{(1)} + Q_1^{(1)}\right]}{\mathsf{E}_\infty\left[A_1^{(1)}\right]}\right)^{|\mathcal{N}_r|} - 1 \\
&\geqslant \left(\frac{e^c}{\mathsf{E}_\infty\left[A_1^{(1)}\right]}\right)^{|\mathcal{N}_r|} - 1 \qquad (6.17) \\
&= \exp\left(|\mathcal{N}_r| c\left(1 - \frac{\ln \mathsf{E}_\infty A_1^{(1)}}{c}\right)\right) - 1. \qquad (6.18)
\end{aligned}$$

The inequality in the second step follows from $\mathsf{E}_\infty\left[Q_1^{(1)}\right] \geqslant e^c$ (see Eqn. 5.2.80 in [Basseville and Nikiforov, 1993]) and $\mathsf{E}_\infty A_1^{(1)} \geqslant 0$. In the same way as in Eqn. 5.2.67 in [Basseville and Nikiforov, 1993], we can show that

$$\mathsf{E}_\infty A_1^{(1)} \leqslant \frac{c + e_u + o(1) + e_d}{\mathrm{KL}(f_0, f_1(\cdot; r_s))},$$

where $e_u + o(1) = \mathsf{E}_\infty R_j^{(i)}$ is the mean excess above c (in up–crossing) and e_d is the mean excess in down–crossing. Note that the positive constants, e_u and e_d do not depend on the threshold c ([Woodroofe, 1982], [Basseville and Nikiforov, 1993]). Define $\beta = e_u + e_d$. Hence,

$$\begin{aligned}
\frac{\ln \mathsf{E}_\infty A_1^{(1)}}{c} &\leqslant \frac{\ln(c + \beta + o(1)) - \ln \mathrm{KL}(f_0, f_1(\cdot; r_s))}{c} \\
&= \frac{\ln(\beta + o(1))}{c} + \frac{\ln(1 + \frac{c}{\beta + o(1)})}{c} - \frac{\ln \mathrm{KL}(f_0, f_1(\cdot; r_s))}{c} \\
&\leqslant \frac{\ln(\beta + o(1))}{c} + \frac{1}{\beta + o(1)} - \frac{\ln \mathrm{KL}(f_0, f_1(\cdot; r_s))}{c}. \qquad (6.19)
\end{aligned}$$

Hence, Theorem 6.2 follows from Eqns. (6.17) and (6.19) as $\mathsf{TTF}_r \leqslant \mathsf{T}_{\mathsf{FA}r}$. ■

Proof of Theorem 6.3 The true hypothesis is $\mathbf{H}_i$, but false isolation has occurred by declaring the hypothesis $\mathbf{H}_j$ to be true, and hence isolating the event to $\mathcal{B}_j$ even though the event does not lie in $\mathcal{B}_j$. Note that $\mathcal{N}_j = (\mathcal{N}_j \cap \mathcal{N}_i) \bigcup (\mathcal{N}_j \setminus \mathcal{N}_i)$. Since we have assumed that $\mathbf{H}_i$ is true, the sensors in the set $\mathcal{N}_j \cap \mathcal{N}_i$ are within the influence range of the event. For the Boolean sensing model, the event is beyond the influence range of all the sensors in the set $\mathcal{N}_j \setminus \mathcal{N}_i$ (since the influence range is the same as the sensing range), and for the path–loss model the event is beyond the influence range of at least one sensor in $\mathcal{N}_j \setminus \mathcal{N}_i$ (otherwise, by definition, the event is in $\mathcal{B}(\mathcal{N}_j)$). Thus, we have defined $\lambda_{ij} = |\mathcal{N}_j \setminus \mathcal{N}_i|$ for the Boolean sensing model, and 1 for the path–loss sensing model. From sample path argument, it is clear that the time taken to raise an alarm in region $\mathcal{B}_j$ is at least as large as the time taken when we restrict the set $\mathcal{N}_j$ to those which are beyond the influence range of each of the sensors in $\mathcal{N}_j \cap \mathcal{N}_i$ (from Lemma 2 and the definition of influence range, we see that the set of sensors in the influence region take less than $\exp(\underline{\omega_0} c)$ samples to cross the threshold). There are at least λ_{ij} nodes in this restricted set. The observation of any of these sensor nodes is just the sensor noise and hence the theorem follows from Eqns. (6.11) and (6.12), from Lemma 6.2, and from Theorem 6.2. ■

Chapter 7

Conclusions

In this thesis, we studied event detection problems under various scenarios. Our major contributions in this thesis, are as follows. We identified a number of limitations of the classical change detection problem. For some of the limiting cases, we provided mathematical models, formulated the event detection problems, and obtained optimum detection procedures.

In the first part of the thesis, we focused on Bayesian event detection in a small extent network. As the network is a small extent one, our interest was in obtaining centralised detection procedures.

In Chapter 3, we studied the *problem of quickest event detection with* sleep–wake *scheduling.* At each time instant k, we wish to control the sleep–wake state of sensors, in addition to making a decision on stopping at time k or to continue sampling at time $k+1$. In order to control the number of sensors in the wake state, we added the sensing+computation+communication cost incurred for using each sensor in each time instant in the wake state to the Bayesian cost function of the classical change detection problem. We were interested in obtaining optimal detection procedures that minimise the Bayesian cost subject to the constraint, $\mathsf{P}_{\mathsf{FA}} \leqslant \alpha$. We showed that the optimization problem can be cast as a partially observable Markov decision process (POMDP). We also showed that at any time k, a sufficient statistic for this problem is the a posteriori probability of change, $\Pi_k = \mathsf{P}\left\{T \leqslant k \mid I_k\right\}$. Based on the theory of POMDP, we obtained quickest event detection rule along with optimal closed loop and open loop control policies for choosing a number of sensors in the wake state. In particular, at each time k, based on Π_k, we obtained

1. optimal (closed loop) control of the number of sensors in the wake state at time $k+1$,
2. optimal (closed loop) control of the probability of a sensor in the wake state at time $k+1$, and
3. optimal (open loop) control of the probability of a sensor in the wake state.

It is to be noted that the optimal open loop control policy does not depend on Π_k, and is computed at time $k = 0$. We argued that the optimal number of sensors to be kept in the wake state is small when Π_k is either small (close to 0) or large (close to 1), and is large when Π_k is neither small nor large (in the numerical example, we see that this happens around 0.5). Also, we observed from a sample path of the $\{\Pi_k\}$ process that for most of the time, the closed loop control policies keep only a few sensors in the wake state.

Several extensions to this work is possible. We considered a *centralised detection* problem. However, it will be interesting to extend our problem to the *decentralised detection* setting (see for example, [Veeravalli, 2001]). Also, one can try to obtain some interesting simple heuristic policies for sleep–wake control.

In Chapter 4, we studied the *problem of event detection on wireless ad hoc networks.* In this problem, the sensor nodes and the fusion centre are connected by a wireless ad hoc network like IEEE 802.11 or ZigBee. To the best of our knowledge, the problem of sequential change detection over a network has not been studied in the literature so far, and this is the first work to investigate this problem in great detail. It is to be noted that the sensors, in general, generate samples at a periodic sampling rate of r samples per time slot. Thus, we are also interested in studying the effect of sampling rate on the mean detection delay.

We proposed a decision strategy which we call Network Aware Decision Making (NADM). We now describe the NADM procedure in detail. We time–stamp the samples with the sampling instants. The samples contend for channel access and reach the fusion centre eventually. The fusion center is equipped with a sequencer buffer and a decision maker. We enqueue in the sequencer buffer, the out–of–time sequence samples that are successfully transmitted and release them to the decision maker as early as possible. By using the sequencer buffer and the queueing strategy, we ensured that the decision maker receives the samples in the non–decreasing order of time–stamps, as early as possible. In each time slot k, the decision maker makes a decision on stopping at time k or to wait for samples at time $k+1$, based on the samples it has received thus far.

We were interested in obtaining an optimal NADM procedure in the fusion centre that minimises the mean detection delay subject to the constraint, $\mathsf{P}_{\mathsf{FA}} \leqslant \alpha$. We modelled the MAC layer of the network as a generalized processor sharing queueing system. It is necessary to understand the network delay suffered by each sample for information fusion by the decision maker. We showed that the network delay of each sample can be computed from the state of the network, $\mathbf{Q}_k$. Since at each time k, a sample that the decision maker receives, if any, carries information about the state of nature delayed by Δ_k, we need to keep track of the state of nature at time instants $k-\Delta_k, k-\Delta_k+1, \cdots, k$. Thus, we appended the network–state, $\mathbf{Q}_k$ and $\mathbf{\Theta}_k$, where $\mathbf{\Theta} := [\Theta_{k-\Delta_k}, \Theta_{k-\Delta_k+1}, \cdots, \Theta_k$. We showed that the system with the state $[\mathbf{Q}_k, \mathbf{\Theta}_k]$ forms a discrete time dynamical system. We also showed that the distribution of samples that the decision maker receives in time slot $k+1$ is governed by the state of the system in time slot k. Thus, we formulated the problem of Bayesian quickest event detection using NADM strategies as a POMDP. We showed that at any time k, a sufficient statistic for this problem is the vector, $[\mathbf{Q}_k, \Pi_k]$, where $\Pi_k = \mathsf{P}\left\{T \leqslant k \mid I_k\right\}$. Based on the theory of POMDP, we obtained an optimal NADM policy. We showed that the optimal NADM policy is a threshold rule on Π_k where the threshold depends on the network–state, $\mathbf{Q}_k$.

We also studied the tradeoff between the mean detection delay of the optimal NADM procedure and the sampling rate r. We note that the detection delay has three components:

i) sampling delay, ii) network–delay, and iii) decision delay. We observed that a large (small) sampling rate r implies a large (small) network–delay and a small (large) sampling–delay. We showed numerically that an optimal sampling rate r^* exists which offers a minimum mean detection delay. We also studied the mean detection delay performance as a function of the number of sensor nodes deployed while keeping the total number of observations per time slot (nr) fixed at some value. We observed that the total number of observations per time slot determines the load on the queueing–network. We observed numerically for a set of parameters that when the load on the queueing–network is maximum ($nr = 1/3$), using 1 sensor node is optimal, and when $nr = 1/100$, using 10 sensor nodes is optimal.

In Chapter 5, we studied the *problem of transient change detection.* A transient change is a short–lived change, which disappears after staying for a finite time in the system. We proposed a Markov chain model for transient change which is an extension of the classical change model. In this work, we were interested in obtaining optimal detection procedures for transient change subject to the false alarm constraint, $\mathsf{P}_{\mathsf{FA}} \leqslant \alpha$.

As in Chapters 3 and 4, we formulated the optimal transient change detection problem as a POMDP. We showed that a sufficient statistic at any time instant k is the a posteriori probability vector $[\Pi_{k,0}, \Pi_{k,1}, \Pi_{k,2}]$ where 0 represents pre–change state, 1 represents in–change state, and 2 represents out–of–change state. Using the theory of POMDP, we obtained the following detection procedures:

1. MinD (**Min**imum Detection **D**elay) which minimises the mean detection delay under the constraint $\mathsf{P}_{\mathsf{FA}} \leqslant \alpha$

2. A–MinD (**A**symptotic – **Min**imum Detection **D**elay) which is asymptotically equivalent to MinD when the mean time until the occurrence of change goes to ∞ (i.e., for a rare event)

3. MaxP (**Max**imum **P**robability of change) which maximises the probability of stopping in the in–change state (called as probability of detection, P_{D}) under the constraint, $\mathsf{P}_{\mathsf{FA}} \leqslant \alpha$.

We observed numerically that the mean detection delay of A–MinD is approximately the same as that of MinD. This suggests that A-MinD is a better approximation to MinD. Also, A-MinD being a simple threshold rule is easy to implement. We also studied the detection delay and the probability of detection performance of only the events that are stopped in the in–change state. Here, we observed that CUSUM outperforms MinD in mean detection delay and performs close to MaxP in probability of detection.

In the second part of the thesis, we focused on large extent networks. In this part of the thesis, we studied the problem of detecting and locating an event (also called as isolation) in a large extent WSN. Since the network is large, our interest was obtaining quickest distributed detection/isolation procedures. To the best of our knowledge, our work is the first to consider the problem of distributed event detection/isolation in a large extent WSN.

In Chapter 6, we studied the *problem of event detection/isolation in large extent networks*. We formulated the problem as an optimization problem that minimises the worst case detection delay subject to time–to–false alarm, T_{FA} and time–to–false isolation, T_{FI} constraints. The problem formulation is done in the framework of [Nikiforov, 1995]. In [Nikiforov, 1995], Nikiforov considered a Boolean sensing model and proposed a centralised minimax optimal change detection procedure. In this work, we considered a realistic signal propagation model in which the signal strength decays with distance. Thus, the post–change mean of the distribution of observations, which differs across sensors, is unknown as the location of the event is unknown. We proposed the worst case mean based on the *detection–range* of the sensors for the post–change distribution.

Based on the *detection–region*s of the sensors, we partitioned the ROI into a minimal number of subregions, each subregion being detection–covered by a unique set of sensors. We use these sets of corroborating sensors to identify the location of the event. we proposed the following local decision rules based on CUSUM: MAX, ALL, and HALL, and a global detection/isolation rule at the corroborating sets of sensors. We analysed the detection delay performance of the distributed event detection/isolation procedures we proposed. We showed that as $\min\{\mathsf{T}_{\mathsf{FA}i}, \mathsf{T}_{\mathsf{FI}ij}\} \to \infty$ for all $i, j \neq i$, the distribution procedures based on ALL and HALL are approximately worst case detection delay optimal.

In this thesis, we studied a number of event detection problems that arise in a WSN and obtained optimal solutions for the problems. Our studies would help in improving the design and implementation of a WSN for event detection applications.

For future research, one can explore the problem of sequential event detection on ad hoc networks (NODM or NADM processing) with sleep–wake scheduling. Also, the problem of transient change detection on ad hoc networks will be an interesting problem to explore.

Bibliography

[Aldosari and Moura, 2004] Aldosari, S. A. and Moura, J. M. F. (2004). Detection in decentralized sensor networks. In *Proceedings of ICASSP*, pages II:277–280.

[Appadwedula et al., 2005] Appadwedula, S., Veeravalli, V. V., and Jones, D. L. (2005). Energy-efficient detection in sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):693–702.

[Baccelli and Makowski, 1990] Baccelli, F. and Makowski, A. (1990). Synchronization in queueing systems. In Takagi, H., editor, *Stochastic Analysis of Computer and Communication Systems*, pages 57–131. North–Holland.

[Basseville and Nikiforov, 1993] Basseville, M. and Nikiforov, I. V. (1993). *Detection of Abrupt Changes: Theory and Application.* Prentice Hall, Englewood Cliffs, NJ.

[Bertsekas, 2000a] Bertsekas, D. P. (2000a). *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, second edition.

[Bertsekas, 2000b] Bertsekas, D. P. (2000b). *Dynamic Programming and Optimal Control*, volume II. Athena Scientific, second edition.

[Bertsekas, 2005] Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control*, volume II. Athena Scientific, third edition.

[Bertsekas, 2007] Bertsekas, D. P. (2007). *Dynamic Programming and Optimal Control*, volume II. Athena Scientific, third edition.

[Hernández-Lerma and Lasserre, 1996] Hernández-Lerma, O. and Lasserre, J. B. (1996). *Discrete–Time Markov Control Processes:Basic Optimality Criteria*. Springer–Verlag, New York.

[Kumar et al., 2004] Kumar, A., Manjunath, D., and Kuri, J. (2004). *Communication Networking: An Analytical Approach*. Morgan-Kaufmann (an imprint of Elsevier), San Francisco.

[Kumar et al., 2008] Kumar, A., Manjunath, D., and Kuri, J. (2008). *Wireless Networking*. Morgan-Kaufmann (an imprint of Elsevier), San Francisco.

[Lai, 1998] Lai, T. L. (1998). Information bounds and quick detection of parameter changes in stochastic systems. *IEEE Transactions on Information Theory*, 44(7):2917–2929.

[Lehmann and Casella, 1998] Lehmann, E. L. and Casella, G. (1998). *Theory of Point Estimation*. Springer–Verlag, New York.

[Liu and Towsley, 2004] Liu, B. and Towsley, D. (2004). A study of the coverage of large-scale sensor networks. In *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*.

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

[Malladi and Speyer, 1999] Malladi, D. P. and Speyer, J. L. (1999). A generalized Shiryayev sequential probability ratio test for change detection and isolation. *IEEE Transactions on Automatic Control*, 44(8):1522–1534.

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

[Meyn and Tweedie, 1993] Meyn, S. and Tweedie, R. L. (1993). *Markov Chains and Stochastic Stability*, volume I. Cambridge University, first edition.

[Mote] Mote. http://www.xbow.com/.

[Moustakides, 1986] Moustakides, G. V. (1986). Optimal stopping times for detecting changes in distributions. *The Annals of Statistics*, 14(4):1379–1387.

[Nikiforov, 1995] Nikiforov, I. V. (1995). A generalized change detection problem. *IEEE Transactions on Information Theory*, 41(1):171–187.

[Niu and Varshney, 2005] Niu, R. and Varshney, P. K. (2005). Distributed detection and fusion in a large wireless sensor network of random size. *EURASIP Journal on Wireless Communications and Networking*, 4(7):462–472.

[Page, 1954] Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1/2):100–115.

[Pollak, 1985] Pollak, M. (1985). Optimal detection of a change in distribution. *The Annals of Statistics*, 13(1):206–227.

[Polunchenko and Tartakovsky, 2009] Polunchenko, A. S. and Tartakovsky, A. S. (2009). Sequential detection of transient changes in statistical distributions. Technical report, University of Southern California.

[Prasanthi and Kumar, 2006] Prasanthi, V. K. and Kumar, A. (2006). Optimizing delay in sequential change detection on ad hoc wireless sensor networks. In *IEEE SECON*, VA, USA.

[Premkumar and Anurag Kumar, 2008] Premkumar, K. and Anurag Kumar (2008). Optimal sleep-wake scheduling for quickest intrusion detection using sensor networks. In *Proc. IEEE Infocom*, Arizona, USA.

[Premkumar et al., 2010] Premkumar, K., Anurag Kumar, and Veeravalli, V. V. (2010). Bayesian quickest transient change detection. In *Proc. International Workshop on Applied Probability*, Colmenarejo, Spain.

[Premkumar et al.] Premkumar, K., Prasanthi, V. K., and Kumar, A. Delay optimal event detection on ad hoc wireless sensor networks. *ACM Transactions on Sensor Networks*. to appear.

[Raghavan and Veeravalli, 2010] Raghavan, V. and Veeravalli, V. V. (2010). Quickest change detection of a markov process across a sensor array. *IEEE Transactions on Information Theory*, 56(4).

[Rago et al., 1996] Rago, C., Willet, P., and Bar-Shalom, Y. (1996). Censoring sensors: A low communication-rate scheme for distributed detection. *IEEE Transactions on Aerospace and Electronic Systems*, 32(2):554–568.

[Ritov, 1990] Ritov, Y. (1990). Decision theoretic optimality of the cusum procedure. *Annals of Statistics*, 18(3):1464–1469.

[Roberts, 1966] Roberts, S. W. (1966). A comparison of some control chart procedures. *Technometrics*, 8(3):411–430.

[Rockafellar, 1997] Rockafellar, R. T. (1997). *Convex Analysis*. Princeton University Press, Princeton, New Jersy.

[Shiryaev, 1963] Shiryaev, A. N. (1963). On optimum methods in quickest detection problems. *Theory of Probability and its Applications*, 8(1):22–46.

[Shiryaev, 1978] Shiryaev, A. N. (1978). *Optimal Stopping Rules*. Springer, New York.

[Singh et al., 2008] Singh, C. K., Kumar, A., and Ameer, P. M. (2008). Performance evaluation of an IEEE 802.15.4 sensor network with star topology. *Wireless Networks*, 14(4):543–568.

[Solis et al., 2006] Solis, R., Borkar, V. S., and Kumar, P. R. (2006). A new distributed time synchronization protocol for multihop wireless networks. In *45th IEEE Conference on Decision and Control (CDC'06)*.

[Tartakovsky and Kim, 2006] Tartakovsky, A. G. and Kim, H. (2006). Performance of certain decentralized distributed change detection procedures. In *9th International Conference of Information Fusion*.

[Tartakovsky and Polunchenko, 2008] Tartakovsky, A. G. and Polunchenko, A. S. (2008). Quickest changepoint detection in distributed multisensor systems under unknown parameters. In *11th International Conference of Information Fusion*, Germany.

[Tartakovsky and Veeravalli, 2003] Tartakovsky, A. G. and Veeravalli, V. V. (2003). Quickest change detection in distributed sensor systems. In *Sixth International Conference of Information Fusion*, volume 2, pages 756–763.

[Tartakovsky and Veeravalli, 2005] Tartakovsky, A. G. and Veeravalli, V. V. (2005). General asymptotic bayesian theory of quickest change detection. *SIAM Theory of Probability and its Applications*, 49(3):458–497.

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

[Tenny and Sandell, 1981] Tenny, R. R. and Sandell, N. R. (1981). Detection with distributed sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 17:501–510.

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

[Woodroofe, 1982] Woodroofe, M. (1982). *Nonlinear Renewal Theory in Sequential Analysis*. SIAM, Philadelphia.

[Wu et al., 2007] Wu, Y., Fahmy, S., and Shroff, N. B. (2007). Energy efficient sleep/wake scheduling for multi-hop sensor networks: Non-convexity and approximation algorithm. In *Infocom*, Alaska, USA.

[Yakir, 1994] Yakir, B. (1994). Optimal detection of a change in distribution when the observations form a markov chain with finite state space. *IMS Lecture Notes - Monograph Series*, 23:346–358.

[Zacharias and Sundaresan, 2007] Zacharias, L. and Sundaresan, R. (2007). Decentralized sequential change detection using physical layer fusion. In *IEEE ISIT*, Nice, France.