Title
Large Monitoring Systems: Data Analysis, Design and Deployment

Author
Rajagopal, Ram

Publication Date
2009

Peer reviewed|Thesis/dissertation

Large Monitoring Systems:
Data Analysis, Design and Deployment

by

Ram Rajagopal

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences
and the Designated Emphasis

in

Communication, Computation, and Statistics

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor Pravin Varaiya, Chair
Professor John Rice
Professor Martin Wainwright
Professor Jean Walrand

Fall 2009

The dissertation of Ram Rajagopal, titled Large Monitoring Systems: Data Analysis, Design and Deployment, is approved:

Chair ____ Date __________

____ Date __________

____ Date __________

____ Date __________

University of California, Berkeley

Large Monitoring Systems:
Data Analysis, Design and Deployment

Abstract

Large Monitoring Systems:
Data Analysis, Design and Deployment

by

Ram Rajagopal

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

and the Designated Emphasis in Communication, Computation, and Statistics

University of California, Berkeley

Professor Pravin Varaiya, Chair

The emergence of pervasive sensing, high bandwidth communications and inexpensive data storage and computation systems makes it possible to drastically change how we design, monitor and regulate very large-scale physical and human networks. Even small performance gains in the way we operate these networks translate into large savings. There are many critical challenges to create functional monitoring systems, such as data reliability, computational efficiency and proper system design, including choices of sensors, communication protocols, and analysis approaches.

In this dissertation introduces a framework to design a monitoring system, deploy it, maintain it and process the incoming heterogeneous sources of information, resulting in new applications. The framework is applied to urban traffic monitoring and road infrastructure sensing. We develop various state of the art statistical inference algorithms, compute performance guarantees and study some of the fundamental limits of the proposed ideas. We illustrate the methodology using experimental deployments we have built and are currently in use.

To my mother and father: *for sharing with me their love for life.*

Contents

List of Figures

List of Tables

Acknowledgments

This has been a long and wonderful journey, impossible without the many people who supported me. When I look back all I can see are the faces of mentors, friends and family that brought me here.

Pravin Varaiya has been the best advisor I could ask for. He encouraged me and was very patient with my intellectual wanderings. I learnt from him that research can be both very rigorous, with keen attention for details, and practical. But more importantly, I learnt the importance of independence in academic pursuits. His approach to mentoring, teaching and nurturing students is my ideal as an academic.

The dissertation owes a lot to interactions with Martin Wainwright. He encouraged me to be rigorous, and to investigate connections between computation, communication and statistics. His research approach inspired me at various points. I will always remember his enthusiasm and friendly advice on science and soccer. I learnt from Prof. Alexander Kurzhanski how to model robust uncertainties and what makes a good systems researcher. I would like to thank Prof. John Rice for advice on Statistics, interesting discussions on modeling sensor faults, and for serving on my dissertation committee. Prof. Jean Walrand has been very kind with sharing his time, and serving on both my dissertation and quals committees. Special thanks to Profs. David Aldous and Anant Sahai, whose probability theory and stochastic systems courses, kept me awake long hours and shaped my research interests.

I would like to thank Dr. Jakka Sairamesh, from IBM Research T.J. Watson, for hosting me for a year in New York. I had a chance to live in Manhattan and think about monitoring design and stochastic systems. Karric Kwong and Robert Kavaler from Sensys Networks were instrumental in creating various deployments used in the thesis. Ruth Gjerde deserves a huge special thanks, for patience and kindness. Without her, I would have been unable to satisfy requirements, register in time and add a Stats M.A. to my degree.

Some of the best work, and fun, in my dissertation resulted from collaborations with fellow students, who have become dear friends: Andreas Krause shared his expertise in approximation algorithms and good writing; Long Nguyen shared his profound statistical insight while we spent countless hours trying to grasp change point sequential analysis. During some breaks, I met his wonderful family, Bong and Lan; Shankar Bhamidi enthusiastically supported bad jokes and shared deep probabilistic insights; Sebastien Roch, the most awesome collaborator, who taught me all about short quartets; Ronnie Bajwa for many field trips to deploy sensors and to Viks; Charis Kaskiris, who shared his expertise in economics; Alex Kurzhanskiy for simulating politics and traffic; and Sinem Ergen, who patiently heard my heated arguments for fault detection, and helped making them into papers. Special thanks to Guilherme Rocha, for random philosophical discussions, but more importantly for guiding me through measure theory in Stat 205. When I freshly arrived at Berkeley, the warm reception in Pravin's lab, by Duke Lee, Rahul Jain and Tunc Simsek made me feel at home. They also advised me on courses and general Berkeley life.

Thanks to my EECS classmates Anand Sarwate, Alvaro Padilla, Hansen Bow, Jana Van Gruenen and Kofi Boakye, who shared trips, prelims and memories. Sheilon Wunder and Juan Lizarrazo are current (and great) work mates and friends. My life at Berkeley wouldn't have been remotely sane without the brazilian crowd. Mauricio Mancio (and then

Jo), Fernando Gonçalves (and then Flávia), Alfredo Cesar Melo and Bruno Salama made me feel at home at the I-House. Gregório and Carol Caetano were my family away from home, and a great source of debates on science, life and statistics. Felipe de Barros has become a close friend, and shared his knowledge on music and PDEs. Andres Donangelo gave me tips on everything from bikes to finance. Rodrigo Fonseca (the other brazilian in EECS) and Paula, Flávio Oliveira and Roberta are kind, fun and know how to have a good time.

I also have to thank all the support and encouragement I received from my friends in Rio de Janeiro: Luis Arthur Pinto, Janaina Tavares, Rafael Castro, Rafael Lima and Daniel Malaguti. Luis Eduardo is a true brother, and shared many advices on literature and Rio. Luis Antônio has been a mentor and friend, and helped me in critical stages. Swami Nithyananda has served as a mentor on all things spiritual, guiding me through the huge cultural forest of indian philosophy.

Thais Sakuma has been my sweetheart and close companion. I am grateful for her encouragement and love. I reserved for last thanking my family: mom, dad and Lakshmi. They serve as my inspiration for persisting on my pursuits, and their love and support mean the world to me. My dad's love for science, my mom's openness to learning and experimenting, and my sister's upbeat attitude to life and friendship have made a deep impression in me. Thank you with all my heart.

Chapter 1

Introduction

"Scientists investigate that which already is;
Engineers create that which has never been. "
Albert Einstein

One of the main challenges in modern systems engineering is to build *adaptive signal and information systems* that monitor and regulate very large dynamic networks, such as urban traffic, patient care and road infrastructure networks. The dynamics of these systems is determined by the aggregate behavior of a large collection of individual agents that locally sense the environment, while performance objectives are based on global requirements for this behavior. Furthermore, local action can give rise to unexpected global behavior.

Creating a monitoring system for such systems comprises designing the sensing and control architecture, as well as the methods for operating on the sensed information to measure and optimize the performance objective. Existing systems that accomplish these tasks in specific application domains are based on ad-hoc choices resulting in a lack of robustness, reliability and performance guarantees. In this dissertation we address these challenges by building an adaptable and reusable *information architecture* accounting for data, sensing and communication constraints. The architecture relies on *novel sensing* approaches; efficient algorithms to *design sensing systems*; *heterogeneous signal representation and inference* for information fusion and reliability; and *novel theoretical frameworks* to analyze solutions.

The chapter starts by describing urban traffic and infrastructure monitoring, a challenging application domain that illustrates concepts and requirements for building large monitoring systems. The dissertation is concerned with creating and deploying solutions for this problem. The chapter continues by proposing a general framework to address other large monitoring systems. It concludes by discussing the contributions of the dissertation resulting from application of the framework. The contributions are to both, transportation systems monitoring and to the more general problem of building large monitoring systems.

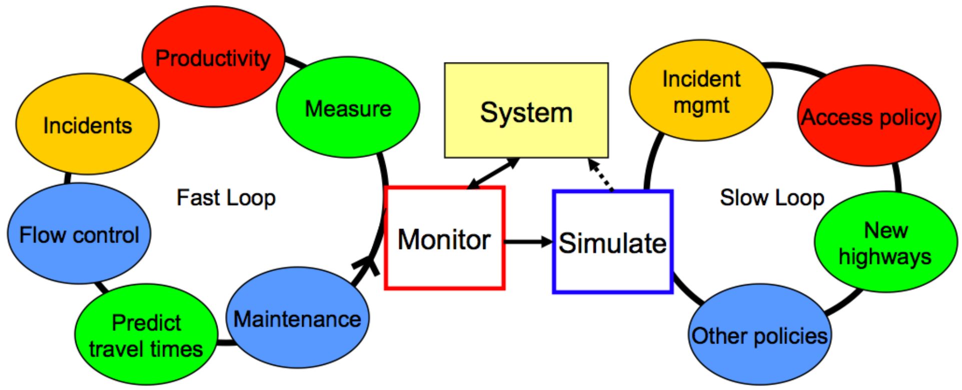


Figure 1.1: Traffic management: a fast operations loop and a slow planning loop.

1.1 Intelligent transportation systems for urban traffic and infrastructure

Urban traffic management [Sussman, 2000] is a challenging engineering and policy problem. Traffic networks are large systems composed of various subsystems whose behavior is primarily governed by the decision making of individual agents, such as drivers, and coordination mechanisms such as traffic signals. Technologies and policies to optimize or increase performance require monitoring the system to understand its behavior, measure the impact of decisions and incorporate feedback as part of the decision making.

Figure 1.1 shows the relationship between monitoring and traffic management. There are two main control loops in traffic management: a fast operations loop and a slow planning loop. The operations loop comprises activities such as measuring the *current* state of the system, evaluating productivity (e.g., total delay hours of congestion), detecting incidents in traffic, performing flow control (e.g., traffic signal control), predicting travel times for user routes and inferring whether sensors are functioning properly. The operations loop requires reliable real-time measurements of the traffic system. The slow planning loop includes actions such as evaluating incident management policies, creating road access policies (e.g., High Occupancy Vehicle special lanes) and deciding where to construct new roads and lanes. The planning loop typically relies on simulations of various different scenarios to understand how long term decisions affect system performance. Tuning the various simulators requires reliable data from a monitoring system.

Recent studies shows poor traffic management costs the United States alone, 78 billion dollars, 4.2 billion lost hours and 2.9 billion gallons of wasted gas annually [Schrank and Lomax, 2007]. If highways were operated at 100% efficiency, traffic congestion could potentially be reduced by 40% [Chen et al., 2005]. The main obstacle to obtain such an improvement is the need for reliable real-time data from monitoring the traffic network. If such data were available, they could be incorporated in optimizing decisions for traffic operations.

Urban traffic management also requires road infrastructures be well preserved and prop-

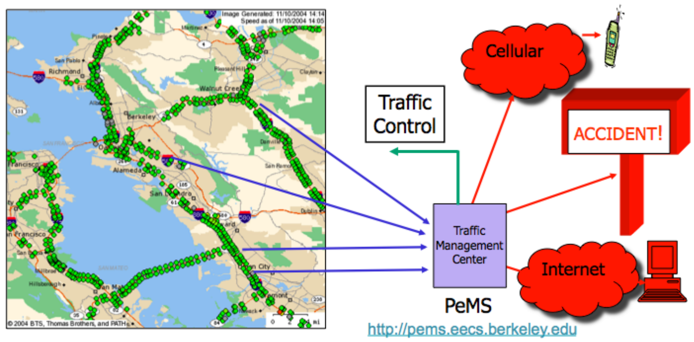


Figure 1.2: Traffic monitoring application (PeMS) and its uses.

erly functional. Four million miles of highway roads and 600,000 bridges form the core vehicular traffic infrastructure in the United States, need monitoring for safety and performance. Current costs for deploying monitoring are extremely high. A single one-time study of a bridge costs $40,000 and installing a single weigh-station to estimate road pavement damage from trucks, costs more than $500,000. Accomplishing such tasks in a cost feasible way requires creating new sensors and new statistical inference methods that can effectively use the sensed data.

1.1.1 Main challenges

There are several important challenges to overcome in order to successfully deploy a urban traffic and infrastructure monitoring system:

Sensing architecture. Due to the sheer size of the system, monitoring traffic requires integrating a large amount of heterogeneous sensors. PeMS is a system created for monitoring traffic in real-time in California highways [PeMS, 2009]. PeMS aggregates data from more than 25,000 traffic sensors, and computes various performance metrics for the fast operational loop. Urban streets do not have the same level coverage due to limitations on types of measurements readily possible with existing sensors. Furthermore, expected deployment lifetimes and available communication channels limit the amount of data that can be transferred from a battery operated sensor, such as a wireless sensor or mobile phone. Usually, most energy consumption is due to data transmission. The architecture should seek to balance local computation with data transmission and accuracy. Options for road infrastructure monitoring are even more limited, the main obstacles being installation costs and longevity.

Data reliability. The harsh operating conditions cause many of the sensors to fail or report incorrect measurements, so it is important to infer the reliability of the reported data. Moreover, due to the large and distributed nature of the sensing architecture, there are several fault points, such as the various data communication links. Statistical approaches to address data reliability and evaluate the quality of the various architecture components is a fundamental requirement to deploy a working sensing system.

State inference. Current classes of sensing are mainly designed for monitoring highways that have continuous traffic flows, and do not work well for monitoring urban street traffic, that has a stop and go nature due to signalized intersections. Addressing how to monitor urban streets requires considering new sensor platforms and methods to infer the state of traffic. Similarly, travel time information from urban links can be inferred from mobile sensor measurements, but require energy aware methods. Finally, inference of the state of infrastructure requires a combination of new sensing modalities and careful physical systems models inferred from data.

System design and optimization. System design encompasses deployment and architectural choices. An important problem is deciding where new sensors need to be deployed in the network. Another problem is determining characteristics of the communication system to obtain best measurement and inference performance.

Performance guarantees. It is important to measure how distant is a monitoring system's performance from its optimum with respective to objectives of interest. Such measurements or guarantees lead to robustness and a principled way of identifying the performance bottlenecks of the monitoring system.

1.2 Constructing large monitoring systems

In this section, we describe how to address challenges in the prior section using a principled approach. We divide our approach to building large monitoring systems into two parts: a high level design methodology, and the underlying technological, statistical and theoretical techniques to achieve this design more concretely. In this section we detail both parts that set the research questions which will be explored by the dissertation.

1.2.1 Design approach

Figure 1.3 depicts a high level view of our approach to create a large monitoring system, which we follow closely to build an autonomous monitoring solution for urban traffic and road infrastructure. The first step is to identify the key state variables that drive the behavior of the system. If they cannot be measured directly, identify measurable surrogate variables, from which the state variables can be inferred. For highways, for example, average speed, flow and number of vehicles per mile of highway are key variables and loop detectors are in-pavement sensors for measuring them. For urban traffic, wireless magnetic sensors capable of local computation are an alternative, but do not directly measure traffic state.

Once we have identified the variables to be measured, we can proceed to the next step in the procedure: creating the sensing platform. In most cases, this requires designing new sensors and deploying them, as well as collecting data from existing sensors. The sensors are heterogeneous, and can vary from standard transducers that report data over wireless links,to text messages from users, images from cameras and intermittent sensing from mobile units. The dissertation proposes new sensors for traffic and infrastructure monitoring, both

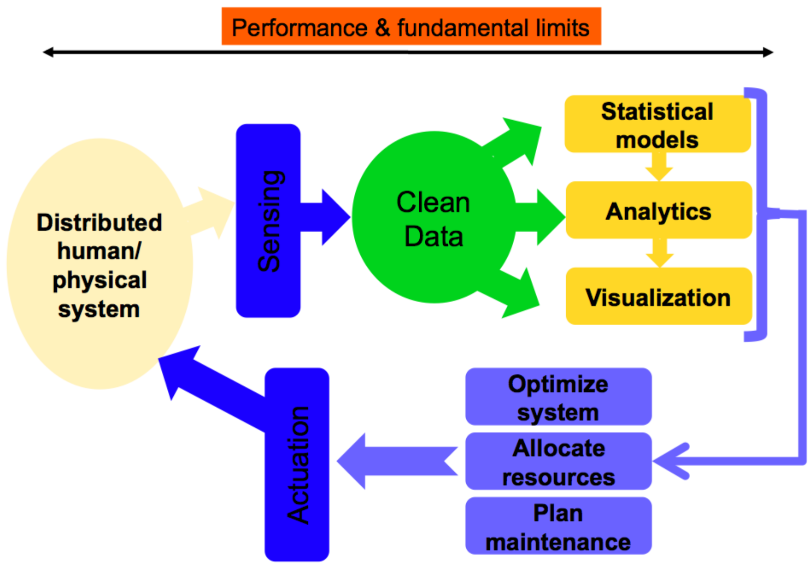


Figure 1.3: Creating a large monitoring and control system for dynamic networks.

autonomous nodes capable of measurement, local computation and wireless communication, but with battery limited lifetime.

Given a fixed budget, *deployment plans* to optimize *where* to place sensors and *when* to sample them need to be created to obtain best performance. For example, selecting time of the day and highway lanes to sample to obtain the maximum amount of information on congestion and accidents, without expending excessive energy.

Since the system is distributed over a large physical area, it is important to design the protocols for collecting and processing the data. Proper *communication protocols and computation protocols*, that determine tradeoffs between local computation and data transmission, need to be specified in order for the system to meet data performance requirements such as data error rates and power consumption targets. The California traffic sensor system has a hierarchical data infrastructure that uses wireless and wired links, and multiple protocols. System sensing performance needs to be measured from diagnostics based on received data and simple sanity checks.

The third step in the procedure is to cleanse and *normalize* the data acquired from the sensors. Deployed sensors might report plausible incorrect values, due to damage and loss of calibration, or values can be missing due to communication intermittency. *Data cleansing* (or normalization) consists of identifying such sensors, discarding the incorrect information, and inferring the incorrect or missing values to provide a proper stream of information. Any system that uses the data without accounting for these issues might perform very poorly.

We separate data normalization from the control and optimization of our system to increase robustness and reduce design complexity. From a statistical viewpoint performing such data completion independently from the final application goals is not guaranteed to be the most efficient. But it is unrealistic to require that every statistical model incorporates data validity explicitly. For example, consider an optimization procedure using N samples

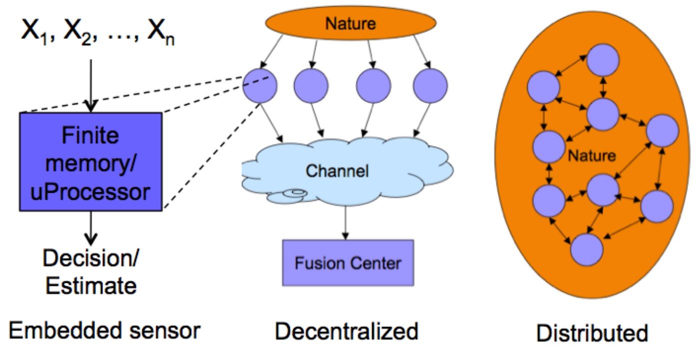


Figure 1.4: Typical monitoring architecture choices.

from a single sensor. There are $P = N!/(R!(N - R)!)$ ways in which R samples can be corrupted or missing. Thus, we may require P versions of the optimization procedure, making it impractical to design a proper solution. Moreover, the modeler of the data frequently is not the designer of the monitoring system, so he is unaware of the various types of possible faults.

We can now build statistical models based on the cleansed data to infer the desired key variables. Proper statistical modeling uses features from data and application related insights. Improperly constructed models do not capture the underlying phenomenon correctly, resulting in poorly estimated state variables, and in turn reduce the benefits of optimizing decision making. This phenomenon has been observed in various large networked systems, such as oil fields and urban traffic. Once key variables are inferred, they can be visualized, and related analytics can be computed. Operations engineers are typically interested in variables that capture a notion of productivity of the system. For example, detection of congestion hotspots and lane utilization statistics for urban traffic. Visualization is a very important aspect of the monitoring system, as it facilitates rapid communication of relevant information allowing effective decision making. Many systems fail to address this aspect, and the monitored data are not effectively used in the decision process.

The design process in Figure 1.3 is completed by *closing the loop*: using the monitored variables to actuate traffic and optimize it. Optimization includes using the inferred variables to plan maintenance of the monitoring and actuation system, allocate resources and optimize parameters affecting system behavior. Some examples are allocation of resources for emergency vehicle response and maintenance plans for traffic detectors.

Actuation might use conventional mechanisms, such as sending signals to actuators, but can also require less conventional mechanisms in the context of controls, such as information systems and incentives to affect the behavior of individual agents. Predictive routing services for vehicles, incident reports over mobile SMS and incentive mechanisms for changing user behavior, are some examples. Many of these methods require further statistical modeling.

1.2.2 Technical approach

Development of solutions for each step proposed in the previous subsection requires creating methodologies that can be implemented within the constraints of the existing architecture. Sensors are embedded, capable of local computation and communicate to a processing center (Figure 1.4). Typically, energy consumption is limited by lifetime requirements. Most energy consumption is due to data communications from the embedded sensor to a power unconstrained local or global fusion center. For example, for wireless traffic sensors, a local fusion computer is connected to the traffic signal control system, which receives regular power. Memory at the sensing units is moderately limited, so measurements for very long periods cannot be stored locally.

There are two typical processing configurations for the sensor network: distributed and decentralized processing [Tsitsiklis, 1993]. In *decentralized processing*, sensors process data locally, transmit information summaries, and the fusion center makes a global decision. The main purpose of local processing is to avoid transmitting all sensed information, and also to schedule the transduction system. Road infrastructure monitoring systems and PeMS are examples of decentralized processing architectures. In *distributed processing*, sensors communicate locally with other sensors or local fusion centers, and compute estimates and decisions. The two main goals are to avoid communication delays in the response and reduce the computational burden in a global fusion center. Distributed processing does not necessarily has to happen at a sensor level. For example, a fusion center could collect decentralized data from the system, but use a distributed method to compute individual decisions about sensor state, in order to reduce computational burden. An important challenge is to design optimization and statistical methods that can operate effectively in these architectures. We address this challenge by creating various distributed and decentralized methods for the components of the monitoring system.

Large networks typically do not have fully observable dynamics. Moreover, the available measurements have associated uncertainties, due to fundamental limitations. Therefore, any method used in a monitoring system requires incorporation of uncertainty in a principled way. *Statistical modeling forms the core of a well designed system.* Particular classes of statistical methods are well suited for this purpose:

Sequential. The real-time nature of monitoring problems and the limited memory in the sensing devices, implies that any local computations should preferably be sequential. In a sequential estimation or decision method, a partial decision is available after each new information is received. Moreover, decisions are update according to rules that only depend on short summaries of the data seen until that point. Sequentially performed estimation or stochastic optimization forms the class of *stochastic approximation methods* [Benveniste et al., 1990; Kushner and Yin, 1997]. An important concern is the speed at which estimates converge to a true value, under communications and noise constraints when calculations are performed sequentially. The dissertation uses a sequential approach for estimating real-time statistics for urban traffic. Another class of problems is to be able to make decisions sequentially, such as detecting which sensors are faulty. *Sequential analysis* [Siegmund, 1985] and *change point methods* [Shirayev, 1978] are two central fields of statistics that discuss the right performance metrics for such decision making and the means to design optimal strategies. The dissertation contributes to sequential change point detection, by proposing a novel type of multiple change point problem and the related analysis.

Spatial and temporal. Dynamics in networks are well characterized by spatial and temporal processes. For example, the vibration of a road caused by truck loads can be modeled using a spatial and temporal partial differential Euler beam equation. Similarly, spatial statistical models [Cressie, 1991] capture more general forms of uncertainties. Modeling, inference and prediction of spatial processes, in a decentralized or distributed computation context, is necessary to create real-time monitoring systems, but not yet fully explored area of research. We use spatial models in two areas, for prediction of traffic in optimization problems, and for prediction of infrastructure response to loading.

Non-parametric. Parametric statistics is concerned with uncertainty models that can be parameterized by a finite number of variables. In contrast, in non-parametric statistics [van der Vaart, 1998], the number of variables typically grows with the number of observations, as no a priori parameterization is used. For example, the empirical histogram is a non-parametric estimator of the distribution of a random variable. The principal advantage of non-parametric methodologies is that it can capture situations where uncertainties cannot be completely modeled a priori. The dissertation develops nonparametric decentralized optimal estimation of quantiles and creates non-parametric statistical measures for fault characterization [Nikulin, 2004].

Approximations. Obtaining optimal solutions for estimation and decision problems, in a decentralized or distributed scenario, is a computationally hard problem [Tsitsiklis, 1993]. In fact, even deciding whether a optimal approach exists can be intractable. Instead, if approximation methods are used, we can obtain deployable algorithms. The dissertation focuses on two types of approximation methods: *approximation algorithms* for combinatorial optimization and *approximate statistical decision* methods. Approximation algorithms [Vazirani, 2001] compute approximate solutions to intractable optimization problems, but are low complexity and provide guarantees on how far the approximation is from the true optimum. Similarly, one can pursue the development of *approximate statistical decision* methods. In this case, various important statistical methodologies are approximated so they operate in decentralized, distributed or computation constrained architectures. Performance guarantees are provided for the obtained solutions. The dissertation advances a new approximate min-max type optimization algorithm, and a novel approximate statistical decision methodology for change point detection.

Creating a solution that performs well in practice, but is also stable and easily maintained, requires that for each step we analyze the behavior of the proposed statistical models and methods, and compute if possible performance bounds and optimality criteria.

1.3 Dissertation organization

The dissertation is organized in chapters that follow the general methodology outlined in Section 1.2. There are three major parts to the dissertation: the first part identifies the variables to be measured and the sensors that will be used (Chapter 2), the second part is concerned with data quality, management and deployment of the sensing infrastructure

(Chapters 3, 4 and 5) and the third part is concerned with applications and measurement of relevant variables (Chapters 6, 7 and 8). The remainder of this section summarizes each chapter.

1.3.1 Chapter 2: Sensing Traffic and Road Infrastructure

The first part of Chapter 2 reviews a system model for a traffic network. We identify the important variables that drive the dynamic behavior of traffic, as well as, the reliability of the traffic infrastructure. A model is then used to show the need for measuring different types of variables for highway networks and urban city networks. For highways, periodic aggregates of traffic variables (e.g., mean) characterize traffic, whereas individual vehicle properties need to be measured for urban streets.

In the second part we review various sensing technologies and their characteristics. We review a wireless magnetic sensor, that is capable of local processing, and is deployed as a device embedded in the road. The sensor can be used to measure aggregate variables for highways and we consider the possibility of using it to measure the state of traffic for urban streets. In Chapter 7 we introduce the statistical method and algorithm to infer the key traffic variables for urban streets from these mesurements. We also propose a new embedded wireless accelerometer to measure road vibration generated by traffic movement. In Chapter 8 we propose the statistical method to convert the vibration measurement into truck loads, which are identified as the key factor that cause road damage. This work was jointly developed with Ronnie Bajwa and Pravin Varaiya.

1.3.2 Chapter 3: Measuring Reliability of a Large Sensor Network

The California Department of Transportation (Caltrans) freeway *sensor network* has two components: the *sensor system* of 25,000 inductive loop sensors grouped into 8,000 vehicle detector stations (VDS) and covering 30,500 freeway direction-miles; and the *communication network* over which the sensor measurements are transported to Caltrans Traffic Management Centers. This sensor network is virtually the only source of data for use in traffic operations, performance measurement, planning and traveler information. However, the value of these data are greatly reduced by the poor reliability of the sensor network: On a typical day in 2005, only 60% of the statewide sensor network provided reliable measurements.

This chapter is an empirical study of the reliability of the sensor network based on data obtained from PeMS. We propose and calculate four non-parametric metrics of system performance: *productivity*, *stability* and *lifetime* and *fixing time*. Based on these metrics we compare the performance of the different districts and verify the limited effectiveness of the DFP. We interpret these metrics to lead to conclusions on how to design a sensor network solution for a large transportation system. This work was jointly developed with Pravin Varaiya.

1.3.3 Chapter 4: Simultaneous Fault Detection for Multiple Sensors

Monitoring its health by detecting its failed sensors is essential to the reliable functioning of any sensor network. We are interested in the detection of sensors that report plausible but incorrect values, as these occur rather frequently in the transportation detector network. This chapter present a distributed, online, sequential algorithm for detecting multiple faults in a sensor network. The algorithm works by detecting change points in the correlation statistics of neighboring sensors, requiring only neighbors to exchange information.

Using sequential analysis, we compute performance guarantees on detection delay and false alarm probability for the algorithm. This appears to be the first work to offer such guarantees for a multiple sensor network. The theoretical framework and resulting algorithm are also useful to explain the performance and improve various correlation tracking methods proposed in the literature. Based on the performance guarantees, we compute a tradeoff between sensor node density, detection delay and energy consumption. We also address synchronization, finite storage and data quantization. We validate our approach using data from the loop detector network. This work was jointly developed with Xuanlong Nguyen, Sinem Ergen and Pravin Varaiya.

1.3.4 Chapter 5: Simultaneous Placement and Scheduling of Sensors

We consider the problem of monitoring spatial phenomena, such as road speeds on a highway, using wireless sensors with limited battery life. A central question is to decide *where* to locate these sensors to best predict the phenomenon at the unsensed locations. However, given the power constraints, we also need to determine *when* to selectively activate these sensors in order to maximize the performance while satisfying lifetime requirements. Traditionally, these two problems of sensor placement and scheduling have been considered separately: one first decides where to place the sensors, and then when to activate them.

In this chapter we present an efficient algorithm, eSPASS, that simultaneously optimizes the placement and the schedule. We prove that eSPASS provides a constant-factor approximation to the optimal solution of this NP-hard optimization problem. A salient feature of our approach is that it obtains "balanced" schedules that perform uniformly well over time, rather than only on average. We then extend the algorithm to allow for a smooth power-accuracy tradeoff. Our algorithm applies to complex settings where the sensing quality of a set of sensors is measured, e.g., in the improvement of prediction accuracy (more formally, to situations where the sensing quality function is *submodular*). Two important applications are *privacy-preserving* sensing using mobile sensors, such as personal mobile phones, and power limited magnetic sensor node deployment. We present extensive empirical studies on these tasks, and our results show that simultaneously placing and scheduling greatly improves performance compared to separate placement and scheduling (e.g., a 33% improvement in network lifetime on the traffic prediction task). This work was jointly developed with Andreas Krause, Anupam Gupta and Carlos Guestrin.

1.3.5 Chapter 6: Estimating Traffic Statistics in a Data Communication-Constrained Setting

Data for urban traffic links are characterized by the *distribution* of the measurements, as opposed to a finite number of moments such as mean and variance. α-quantiles of a distribution are values θ^* so that the probability that the random variable is less than θ^* is α. When $\alpha = 0.5$, we obtain the median. Empirical estimates of quantiles from observed data capture well properties of general distributions. In real settings, quantiles are re-estimated for each link as more data become available, but such procedure usually requires transmission and storage of all the observed values by the sensors. For example, for a median computation, all observed values are required from each sensor. On the other hand, averages can be computed from the sum of the observed values and the number of observed values at each sensor. Due to power and privacy constraints, sensors are restricted on the amount of information they can communicate, and maybe unable to send all the actual measured values.

In Chapter 6 we formulate this problem for both mobile sensors, such as cellphones, and fixed wireless sensors. We state the problem as one of decentralized statistical inference: given i.i.d. samples from an unknown distribution, estimate an arbitrary quantile subject to limits on the number of bits exchanged. We analyze a standard fusion-based architecture, in which each of m sensors transmits a single bit to the fusion center, which in turn is permitted to send some number k bits of feedback. Supposing that each of m sensors receives n observations, the optimal centralized protocol yields mean-squared error decaying as $O(1/[nm])$. We develop and analyze the performance of various decentralized protocols in comparison to this centralized gold-standard. First, we describe a decentralized protocol based on $k = \log(m)$ bits of feedback that is strongly consistent, and achieves the same asymptotic MSE as the centralized optimum. Second, we describe and analyze a decentralized protocol based on only a single bit ($k = 1$) of feedback. For step sizes independent of m, it achieves an asymptotic MSE of order $O[1/(n\sqrt{m})]$, whereas for step sizes decaying as $1/\sqrt{m}$, it achieves the same $O(1/[nm])$ decay in MSE as the centralized optimum. Our theoretical results are complemented by simulations, illustrating the tradeoffs between these different protocols. This work was jointly developed with Martin Wainwright and Pravin Varaiya.

1.3.6 Chapter 7: Measuring Vehicle Travel Times

Chapter 7 describes a system for measuring the vehicle count and travel time in the links of a road network. The measurements require matching vehicle signatures recorded by the wireless magnetic sensor network described in Chapter 2. The matching algorithm is based on a statistical model of the signatures. The model itself is estimated from the data. The approach is first discussed for a single lane road, and extended to multiple lane roads.The algorithm yields a correct matching rate of 75% for a false matching rate of 5%, and reliably estimates the number of vehicles on each link and its travel time distribution.The system is tested on a 0.9 mile-long segment of San Pablo Avenue in Albany, CA. We also present exact and heuristic error calculations for the method, identifying an important class of stochastic shortest path problems. This work was jointly developed with Karric Kwong, Robert Kavaler and Pravin Varaiya.

1.3.7 Chapter 8: Monitoring Load Impact in Roads

Chapter 8 introduces a dynamical model and the statistical method for inferring the load of heavy vehicles on road pavement from measurements from an accelerometer sensor network. Heavy vehicle load cause the most damage to pavements, and monitoring loads and damage improves maintenance planning, incurring in fewer future closures and accidents.

We derive an *approximate* solution method for a distributed parameter system representing the dynamics of the road pavement. Based on the approximation we propose a method for measuring impulsive loading forces. We then analyze the method and conclude its optimality. To conclude the chapter we present some computational experiments as well as a comparison with data collected from a real deployment of a wireless accelerometer sensor network. This work was jointly developed with Alexander Kurzhanski and Pravin Varaiya.

1.4 Summary of contributions

There are three thrusts in the dissertation: a systems design thrust, focused on creating new sensors and deployments, an algorithmic thrust, creating various statistically principled methods for various problems and a theoretical thrust, focused on analyzing the performance of the proposed algorithms. Although some of the methods are presented in the context of transportation applications, their applicability is more general. Some of the theoretical frameworks and contributions are also more generally applicable. Some of the main contributions of the dissertation are

- Embedded accelerometer sensor network for measuring infrastructure system response;
- Metrics for monitoring sensing quality for sensor network applications;
- Theory and algorithms for multiple change-point detection problems where multiple changes are correlated;
- Theory and algorithms for sub-modular discrete min-max problems;
- Algorithm and analysis for distributed sequential quantile estimation;
- Algorithm and heuristic analysis for stochastic matching for travel time estimation;
- Theory and algorithms for estimating a finite parameter in a distributed parameter system.

Chapter 2

Sensing Traffic and Roads

2.1 Introduction

The dynamics of urban traffic is determined by the interaction between individual decisions of drivers, road conditions, traffic control signals and road capacity. Characterizing traffic involves identifying the key variables that affect the dynamics, and designing sensors and systems for measuring these key variables. Monitoring road infrastructure also involves a similar approach. We study in detail the key variables for characterizing the dynamics of traffic, and the response of road infrastructures when subject to an external excitation signal, in the first part of this chapter.

In the second part, we review how we can measure the required key variables using existing sensing systems. In the case of urban traffic, we identify a promising technology for monitoring: magnetic wireless sensor networks. In later chapters of the dissertation we discuss how to use this technology together with sophisticated algorithms to measure traffic very reliably.

For road infrastructure monitoring, we propose an wireless accelerometer sensor network that can be embedded in the pavement. We briefly discuss the steps required for using this technology on the field. We conclude this chapter by addressing the important issue of designing a communication system to connect the sensor systems in the field to a central processing unit.

2.2 What to measure?

2.2.1 Traffic

The typical traffic network model is shown in Figure 2.1. The road network is divided into sections. Each section is represented by an edge. Nodes represent the junction of two or more sections. The nodes may or may not correspond to intersections. Flow control of the network is executed at the nodes. For example, a typical node in an arterial is an intersection, and flow is controlled by a traffic signal. The state of traffic is the state of each edge.

Traffic behaves differently in highways and in signalized roads. In highways, traffic

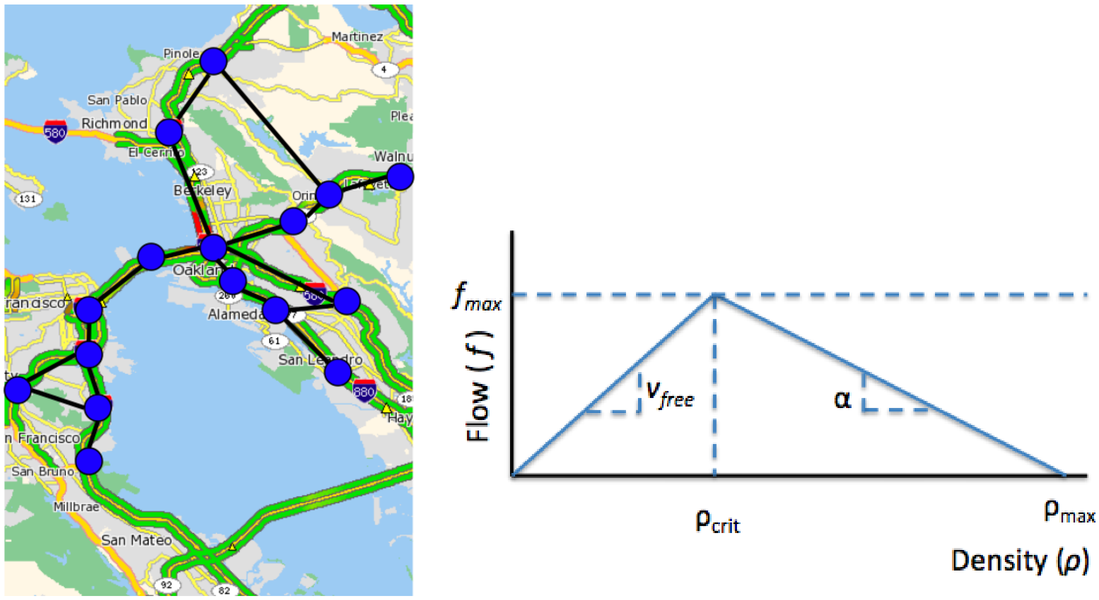


Figure 2.1: (left) Traffic network model and (right) Density, flow and speed relationship in a typical model

behaves like a fluid. Its state can be characterized by point measurements at each road section, such as average density ρ (cars/mile), average flow f (cars/hour) and average speed v (mph). The average is taken over a small period of time, typically 30 seconds. Several models have been proposed for highway traffic inspired by equivalent models in fluid dynamics. One popular model is the Cell Transmission Model (CTM) [Daganzo, 1994]. It assumes that to a first order, there is a relationship between density, flow and speed (Figure 2.1). When traffic is free flowing, $f = \rho\, v_{free}$, where v_{free} is the free flow speed. For highways in California, for example, the free flow speed is approximately 60 mph. Congestion happens when $\rho > \rho_{crit}$, where ρ_{crit} is the critical density, at which point we have the maximum flow $f_{max} = \rho_{crit}\, v_{free}$. In the congested regime, flow decreases with increasing density so that $f = f_{max} - \alpha\,(\rho - \rho_{crit})$. α (mph) is the speed of propagation of the congestion wave. At a certain density, the flow becomes 0. This is the jam density of the road. In the congested regime, we can compute the average speed of a section as $v = \{f_{max} - \alpha\,(\rho - \rho_{crit})\}/\rho$.

CTM builds upon this flow-density relationships to model the behavior over time of various connected highway sections. Observed density, flow and speeds need to be measured to calibrate such a model. Measurements become even more important in the presence of incidents, since then the relationship between these three variables may change in unexpected ways. More complicated micro-simulation models have been proposed, but their calibration also depends on accurately measuring these variables. We will call the triplet (ρ, f, v) the state of a highway section. Notice that in the CTM model, density (ρ) is the only state, but in general it is not known if CTM holds exactly. The state can be measured by looking at averages in a single fixed point in the highway section (space). Other variables such as vehicle type distribution and weather could potentially be added to a highway section state.

A variety of sensors can measure the state of a highway. Some of these sensors are described in Section 2.3.

Point measurements do not characterize traffic well for signalized roads. In such roads, the nature of traffic is 'stop and go' and therefore it is not very meaningful to look at point averages from measurements at a single fixed point. Local averages will be affected by events such as slowdowns, turns and traffic signal changes. Signalized roads behave more similar to a 'store and forward' communication packet networks. Each vehicle is a packet, and it follows a route composed of consecutive links. Its flow is controlled by flow control mechanisms such as traffic lights. By following this analogy, we can consider that the state of a signalized road section is characterized by the number of vehicles (n) stored in the link and the set of travel times for each vehicle to cross the link ($\{t_1, ..., t_n\}$). If every individual travel time cannot be measured, we are interested in measuring various travel time quantiles, such as the median, 25%, 75%, 90% quantiles. Sensors currently being used perform poorly when measuring any of these variables.

A simple statistical model can explain the difficulty. Suppose we place a fixed speed sensor in the middle of a road section of length L. We are interested in measuring the travel time t_i of the vehicle i. Let $\bar{v}_i = L/t_i$ be the average speed with which the vehicle crosses the section. If we can measure $\bar{v}_i$ directly, it is equivalent to measuring t_i. Due to the nature of traffic, the vehicle does not cross the section with constant speed. Therefore, the fixed speed sensor measures the speed of vehicle i as $v_i = \bar{v}_i + \eta_i$, where η_i is an independent, identically distributed (i.i.d) random variable, with zero mean and variance σ^2, but whose distribution is unknown. In a highway, η_i has very small variance but this cannot be said of a vehicle crossing a signalized road. If we were interested in the average speed of the various vehicles we can compute $\hat{v}$:

$$
\begin{aligned}
\hat{v} &= \frac{1}{n}\sum_{i=1}^{n} v_i = \frac{1}{n}\sum_{i=1}^{n} \bar{v}_i + \frac{1}{n}\sum_{i=1}^{n} \eta_i, \\
&= \tilde{v}_n + \bar{\eta}_n.
\end{aligned}
$$

Notice we are interested in measuring $\tilde{v}_n$ in this case, the true empirical mean of the measurements. When η_i has bounded moments, the strong law of large numbers guarantees that $\bar{\eta}_n$ is a random variable converging to 0 as $n \to \infty$, and the weak law of large number states that the variance of $\bar{\eta}_n$ is of order $O(\sigma^2/n)$. For a moderately large n we will have accurate estimates of $\tilde{v}_n$. Instead suppose we would like to measure the median speed of the set $\{\bar{v}_1, ..., \bar{v}_n\}$ from the noisy measurements $\{v_1, ..., v_n\}$. Assume n odd and the set to be in increasing order without loss of generality. Also let $|\eta_i| < B_n$, where $B_n = \min\{\bar{v}_m - \bar{v}_{m-1}, \bar{v}_{m+1} - \bar{v}_m\}$ and $m = (n-1)/2 + 1$. This implies that the empirical median is v_m and the error is exactly η_m. The mean square error of the estimated median is then σ^2, regardless of n. In fact, if we construct a sequence $\{\bar{v}_1, ..., \bar{v}_n\}$ such that $B_n > \alpha$ for all n, then as $n \to \infty$, we are able to estimate the average perfectly from the sequence values, but the median estimate will always have error at least α. We have shown with this construction the difficulty of estimating quantiles from point measurements.

We can generalize the above example to cases where $B_n \to 0$. Assume that $\bar{v}_i$ is i.i.d. and has a cumulative distribution function F with density f and η_i is i.i.d. with distribution

G and density g. For simplicity, assume f is continuously differentiable to all orders. Both distributions forms are unknown. Let the empirical distribution of the set of samples $\bar{v}_i$, $i = 1, ..., n$ be denoted by F_n. When $n \to \infty$, $F_n \Rightarrow F$. The empirical distribution of v converges to a distribution H with cumulative distribution:

$$H(x) = \int_{-\infty}^{+\infty} g(\tau) F(x - \tau) d\tau. \tag{2.2.1}$$

If $g(x) = \delta(x)$, then we are in the noise free case, and $H(x) = F(x)$ using the properties of the Dirac function. Using Taylor expansion:

$$F(v_{med} + \delta - \tau) = F(v_{med}) + \sum_{k=1}^{\infty} \frac{(\delta - \tau)^k}{k!} f^{(k)}(v_{med}),$$

and we can then expand, noting that integrals on τ are equivalent to expectations on random variable η and assuming that η has a small support, and δ is small:

$$\begin{aligned} H(v_{med} + \delta) &= F(v_{med}) + \sum_{k=1}^{\infty} \frac{f^{(k)}(v_{med})}{k!} \mathbb{E}[(\delta - \eta)^k] \\ &\approx \frac{1}{2} + f(v_{med})\, \delta + \frac{f'(v_{med})}{2}\, \sigma_\eta^2. \end{aligned}$$

The median solves $H(v_{med} + \delta) = 1/2$, so solving for δ yields $\delta = -(f'(v_{med})\sigma_\eta^2)/(2\, f(v_{med}))$. In essence, the median has a deviation of order $O(\sigma_\eta^2)$, although we were allowed to observe infinitely many samples.

To continue the simple example, consider the alternative approach where we are allowed to observe a fraction β of the average section speeds $\bar{v}_i$. The strategy corresponds to capturing βn of the vehicles and obtaining a noise free observation. For example, if a fraction of users reporting their travel times or real-time locations can be used for this purpose. In this case, standard statistical analysis shows that the median can be estimated with an MSE that scales as $O(\frac{1}{f(v_{med})\, n})$ where n is the number of observations. Clearly, tracking a subset of individual vehicles travel times allows for better estimates than noisy point observations of all vehicles.

In Section 2.3 we discuss how to measure the state of highways and roads using various sensors.

2.2.2 Roads

A road is composed of several layers of material, such as asphalt, sand or cement, on top of soil. It is usually modeled as a beam lying on several layers of elastic foundation. The dynamics of this system is characterized by the displacement response of each point along the beam, in response to an external excitation force. For example, in a common one-dimensional model of pavement, the response of the beam is given by the displacement $y(x, t)$ at location x meters, at time t. The excitation force is described by a function $F(x, t)$.

Road damage can have various causes, such as the dynamics of the pavement experienced during normal usage, or weather pattern variations.Most frequently the damage is related to the magnitudes of the displacement experienced by the pavement. Large sudden variations on the displacement cause rupture in the layers of material.

Most damage to the road infrastructure is caused by truck traffic [Sousa et al., 1988]. As trucks move over stretches of pavement, small irregularities on the ground cause the suspension system to react, and a dynamic force acts on the surface. The magnitude of this dynamic force is directly related to damage to the pavement and to other supporting infrastructure, such as embedded traffic sensors [Cebon, 1999].

In fact, when a truck is moving on top of a pavement beam there are two forces acting: a static force and a dynamic force. The static force corresponds to the force due to gravity generated by the weight of the truck. Typically, this force plays a minor component with regards to road damage. The dynamic force is the force exerted by the truck when it moves on the road. This force depends on the weight of the vehicle, but also depends on the number of axles and other factors such as the air pressure of tires and suspension system characteristics. The displacement in response to this is $y(x,t) = y_0 + y_d(x,t)$, where y_0 corresponds to a constant displacement to due to static effects from the excitation force, and y_d corresponds to the dynamic component. y_0 has very small effect on road damage. Furthermore, it is possible to infer the weight of the truck, as well as the dynamic force from the measurement of y_d alone.

Measuring displacement directly is difficult. Typically it is done using strain gauges. Existing sensors have a very high installation cost and are not durable. Furthermore, their accuracy is questionable [Cebon, 1999]. On the other hand, accelerometers are reliable and accurate. Vibration is given by the acceleration experienced by the ground:

$$s(x,t) = \frac{\partial^2 y(x,t)}{\partial t^2}. \tag{2.2.2}$$

Notice that we can recover the dynamic component of the displacement from vibration, but not the static component. In the next Section we show how to measure vibrations accurately.

2.3 How to Measure?

In this Section we review various measurement technologies for traffic sensing and road infrastructure monitoring. Sensors can be divided into two main groups: intrusive and non-intrusive. Intrusive sensors are embedded in the pavement, and therefore have higher installation costs. An example of an intrusive sensor is a standard inductive loop used for traffic monitoring. Non-intrusive sensors require less disruptive installation, but usually have less accuracy than intrusive ones. An example of a non-intrusive sensor is a digital camera used in traffic monitoring applications.

2.3.1 Traffic

We review the five common sensing technologies. In Section 2.2 we observed that the variables that need to be measured are different for highways and for signalized roads. In highways, we are interested in measuring point values of speed, flow and density at a given location. Given the nature of traffic, one sample every 30 seconds is sufficient for capturing the important phenomena. Signalized roads, on the other hand, require measuring individual vehicle travel times and the number of vehicles for each road section. Since the sensing requirements are different, the sensors themselves will have different characteristics. The differences extend from the chosen transducer to the platform that is used to collect measurements.

Loop Detectors

Loop detectors are the most common intrusive sensors in highways. They consist of an inductive loop that is embedded in the pavement (Figure 2.2). One sensor is placed per lane. A current is sent through the loop. While a vehicle crosses a loop, it causes a change in the inductance of the loop, in turn changing the frequency of the excitation signal. If the change is above a threshold the vehicle is detected. Speed can be measured using the delay between the detection by two consecutive loops separated by a small known distance [Bickel et al., 2007; Ki and Baik, 2006].

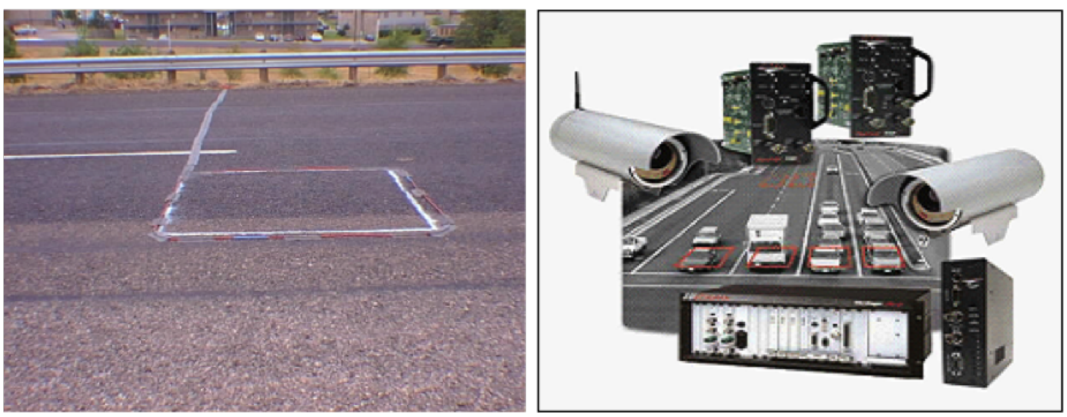


Figure 2.2: Inductive loop installed in the road and a typical camera setup.

Well-tuned inductive loops have high accuracy, but are not very reliable. Their failure rate is high due to the environment they operate in. From temperature to truck traffic, various elements can either break the loop or affect its normal operation. Data quality is a very severe problem in loop detector based systems. In Chapters 3 and 4 we will develop algorithms and statistical methods to evaluate loop failure rates and also detect failed loops. Another important concern is that the installation of loop detectors is very disruptive and expensive. This is a serious drawback to installation at new sites or replacing damaged loops in existing roads.

Systems Based on Imaging

Imaging systems are non-intrusive traffic sensors. The most typical imaging systems are video camera based (Figure 2.2) or infrared (IR) camera based systems [Mimbela and Klein, 2000]. They are installed overhead from the road and require processing the signals appropriately for identifying vehicles.

IR systems detect electromagnetic radiation, that is invisible to the human eye, and can be either active or passive. Active systems emit radiation using appropriate lighting, and the reflected radiation is detected. Presence or absence of vehicles is inferred from the travel times of the signal. Speed is obtained by emitting two signals and measuring the phase difference of the reflected signal. Passive systems rely on measuring gray body emission, the natural radiation emitted from objects. An infrared camera is able to measure this emitted radiation, creating a two dimensional image of the area of interest and changes in this image indicate the presence of a vehicle. Using more sophisticated algorithms, speed can be measured.

A camera based system is composed of one or several cameras, and a system for processing the images to extract traffic information. Variations in successive image frames can be used to detect the presence of a vehicle and tracking the vehicle over successive frames together with data association techniques, such as a Kalman filter, can be used to obtain point estimates of speed [MacCarley et al., 1992; Mimbela and Klein, 2000]. Images can also be used to classify vehicles into several types, such as trucks and cars. Using a multiple camera system deployed along a road, a vehicle can be tracked and individual travel times can be obtained [Klein, 2001]. Under ideal conditions and with a correctly calibrated system, speed estimate accuracies are close to 95% [Michalopoulos et al., 1993]. Unfortunately, system performance is affected by occlusion and environmental factors such as lighting and snow. Furthermore, shadows and camera vibration caused by wind results in a large number of false detections, and poor speed estimates. The installation and maintenance costs are high for such systems, and it is only justified when several detection areas can be capture by a single camera. Single camera systems are useful for measuring highway state parameters. Multiple camera systems can be used in signalized roads, to measure the state parameters.

Radar and Ultrasonic Systems

Radar systems use radio waves to detect direction, distance and speed of target objects. In traffic applications the most common type of radar is microwave radar [Weber, 1999]. There are two types of radar systems: continuous wave (CW) [Duzdar and Kompa, 2001], based on transmitting a single frequency signal in the GHz range, and frequency modulated continuous wave systems (FMCW), based on transmitting a signal with continuously varying frequency. CW systems rely on the Doppler Effect to estimate speed from the frequency shift using a single detector, and as they are unable to detect motionless objects. FMCW can detect motionless objects, but require two detectors to measure speed. Microwave radar systems are not sensitive to weather. The accuracy of speed estimates is 8% for CW systems and 1% FMCW systems.

Ultrasonic systems are similar to microwave radar but use signals at a different frequency spectrum. Multiple detectors are required for speed detection. They are sensitive to temperature and air pressure changes but have higher accuracy than radar systems.

Vehicle Based Probes

Vehicle based probes can provide extensive monitoring if enough vehicles provide information. Typically, they have a smaller upfront cost to the transportation authority, since costs are spread over the various users of the system. On the other hand, a large number of vehicles are required for such a system to provide real-time information. Vehicle based probes provide location information at periodic schedules. Road section travel times can be estimated based on these sequences of locations. These probes do not provide fixed point statistics such as average flow and density, which can be very important for some applications.

There are three major classes of vehicle based probes: GPS based systems, mobile phone systems and RFID based systems. GPS based systems periodically report the GPS location of a vehicle. The time interval between samples depends on the available communication system and in the device used. It can very from 30 seconds to 2 minutes. Measurements of road section travel times can be obtained based on processing the GPS samples by assigning them to a road map, and properly accounting the time intervals between samples to each road section. There are three important and challenging requirements for a successful monitoring system based on GPS samples: collecting the samples at a central location, preserving the privacy of users of the system and having enough adopters to enable road section travel time estimates. Various studies have shown [Zou et al., 2005] that around 8 to 10% of the vehicles need to report their locations continuously for a successful service. Such penetration rates have not been achieved in any existing system yet.

Mobile phone based systems [Zhao, 2000] can rely either on the GPS samples or in using various forms of triangulation to provide location. The challenges of a mobile phone based system are similar to those of a GPS system, with an added concern: mobile phones have limited battery lifetimes and thus cannot rely on GPS only monitoring or complex computations for localization.

RFID based systems use roadside antennas to read RFID tags attached to vehicles. These systems also known as automatic vehicle identification systems, and are used in electronic toll tags collection. The main challenge for adoption of this technology is the need to deploy readers at each location of interest.

Magnetic Wireless Sensor Networks

Magnetic Wireless Sensor Networks (MWSN) are a recent technology developed for traffic applications [Haoui et al., 2008a]. The system consists of two components (Figure 2.3): a pavement embedded intelligent sensor node and an access point (AP) that is associated with a group of nodes.

The sensor node consists of a triaxial magnetometer transducer, a microprocessor and a wireless communication radio. The microprocessor is able to read the analog signals from the magnetometer, process them digitally, and send them over the radio to an AP. The whole system is housed in a 3 inch cube enclosure, resistant to the severe conditions of operation in the highway. It is powered by batteries, and is designed to last 10 years.

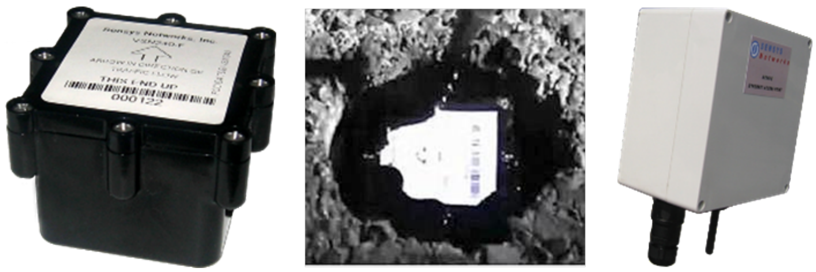


Figure 2.3: Sensys Magnetic Wireless Sensor System. Sensor node, installed sensor node and access point (AP).

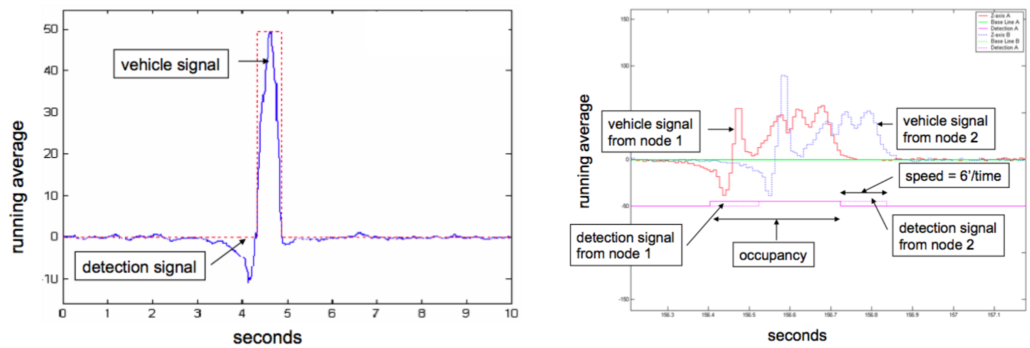


Figure 2.4: Magnetic signature from a sensor node, and corresponding speed computation.

The main limitation on lifetime is the amount of data communication, and usage of radio consumes about 90% of the energy [Ergen and Varaiya, 2006]. The sensor is designed to report 3 to 6 samples every 30 seconds to the AP. The sensor is deployed embedded in the pavement using an epoxy filling (Figure 2.3), and is installed in less than 10 minutes.

The device is designed to sense the change in the earth's magnetic field due to the presence of a vehicle. Whenever a vehicle crosses the sensor, it leaves a magnetic signature for each axis of the magnetometer. The signature is sampled at 128Hz. The length of the signature depends on the speed of the vehicle. Figure 2.4 shows a typical signature. The signature can be thresholded to indicate the presence or absence of a vehicle, and the detection delay between two consecutive sensor nodes can be used for point speed measurements. Figure 2.4 also shows a speed calculation. Each sensor node also reports a time stamp of when the events were recorded.

The AP is typically connected to a group of sensors that are in a fixed location, but at different lanes, in the road (see Figure 2.5). In alternative setups, such as intersections, sensors before and after the intersection can be connected to an AP. This is useful in that data from different sensors can be processed and compared in the AP.

The currently existing sensor network solution is capable of making point measurements of flow, density and speed (if a pair of sensors is used in each lane), therefore serving as a solution for measuring the state of highways. In Chapter 6 we show how we can combine the readings and time stamps of different sensors to create a solution that is capable of measuring

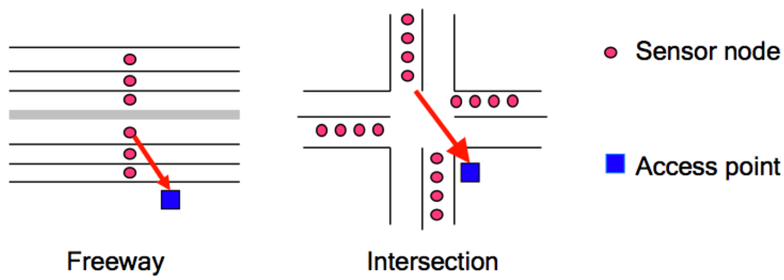


Figure 2.5: Sensor nodes connected to an Access Point.

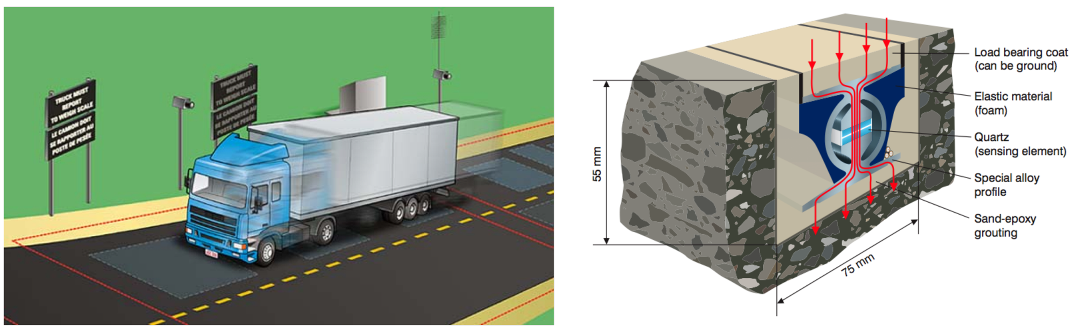


Figure 2.6: Typical weigh-in-motion station configuration and Quartz piezoelectric sensor (Lineas) for measuring displacements in a roadway

individual vehicle travel times, with a high penetration rate. The solution provides the first scalable option for measuring the state of roads in real-time in a privacy preserving way.

2.3.2 Road Infrastructure

Currently, the trucks are charged by weight, measured at weigh-in stations. Most weigh-in stations measure the gross weight of the truck and the axle loads when it is at rest. In some stations, a weigh-in-motion (WIM) system is used [Moses, 1979; Stergioulas et al., 2000], and one can measure the weight of the vehicle as it moves over the sensors. Beyond the total weight, it is very important to count the number of axles in the vehicle, as well as obtain an estimate of individual axle loads. We discuss existing sensors and introduce a novel accelerometer based sensor for this application.

Existing sensors

The most common sensors are piezoelectric, bending plate, load cell, capacitive mat and fiber optic. Piezoelectric sensors are embedded in the pavement (Figure 2.6), and are capable of measuring the speed of the pavement (first derivative of displacement) when subjected to an external excitation signal. Due to their measurement characteristics, piezoelectric sensors can only measure the dynamic component of the force caused by a truck. The installation of the sensor is quite complex, and they require extensive calibration. The response of a piezoelectric sensor is sensitive to the suspension system of a truck, its speed and also to external temperature. Not including these factors into measurements, as done in various deployed systems, results in errors of up to 10% in the estimates of the dynamic component of the force. Piezoelectric sensors also fail often due to weather and traffic conditions.

Bending plate sensors are based on strain gauges attached to a plate. The sensor produces an output proportional to the displacement of the pavement, and therefore can be used to measure both the static load and the dynamic load of a truck. Durability of bending plates is an important issue [Cebon, 1999].

Load cells are based on measuring pressure changes in a fluid contained inside the sensor. Such systems are usually difficult to maintain and very expensive [Mimbela and Klein, 2000]. Capacitive mats are based on creating a capacitor from two metal plates. As vehicles move on top of the capacitor, the distance between the plates changes, and the frequency of an oscillator based on this capacitor changes its frequency. Measurement of this change gives the weight of each axel of the truck. The main issues with these sensors are durability and installation complexity. Fiber optic transducers for WIM station are based on measuring the bending or change in the refractive index of the fiber when a vehicle drives on top of it. These perturbations are measured by intrinsic or extrinsic sensing devices [Safaai-Jazi et al., 1990; Martin et al., 2003].

Accelerometer Wireless Sensor Networks

We have developed an accelerometer wireless sensor network system for road infrastructure monitoring as an alternative to the existing sensors. We use an equivalent platform to the Magnetic Wireless Sensor Network, with the magnetometer replaced by an accelerometer. The sensor node is similar to the one shown in Figure 2.3. It consists of a Silicon Design 2125 MEMS accelerometer, a low pass antialiasing filter with arbitrary gain, an analog to digital converter, a microprocessor and a wireless radio. The SD 2125 has a range of $-5g$ to $5g$,where $g = 9.8$ m$/s^2$ is the gravity constant, with an expected resolution of $50\,\mu$g at 50 Hz bandwidth. The sensor node is battery operated and is housed in an enclosure that is resistant to very heavy loads. The sensor is placed in the pavement inside a 4" diameter hole, that is subsequently filled with epoxy (as in Figure 2.3). The dynamic force generated by typical light trucks causes vibrations on the order of 1mg.

We measure noise as the square root of the mean squared (RMS) of the measurements of the accelerometer when there is no signal present. Under lab conditions the sensor noise level is 120 μg (RMS noise) and when deployed in the field, the noise level is $164\,\mu$g. The measured noise level when a truck has its engine revved up to very high RPM while idling on top of the sensor is $231\,\mu$g, and when the engine is running at regular RPM it is $165\,\mu$g. When truck horn is used, it is $167\,\mu$g. Based on these observations we ascertain the noise level of the sensor as 167 μg.

We also measured the frequency response of the node and compared it to a reference accelerometer in laboratory conditions. Figure 2.7 shows the setup and the measured response. The maximum deviation was 0.1 dB and the average deviation was 0.04 dB. The sensitivity of the accelerometer can be measured using a calibration plate setup, as shown in the diagram in Figure 2.8. A calibration plate consists of a plate of precisely known

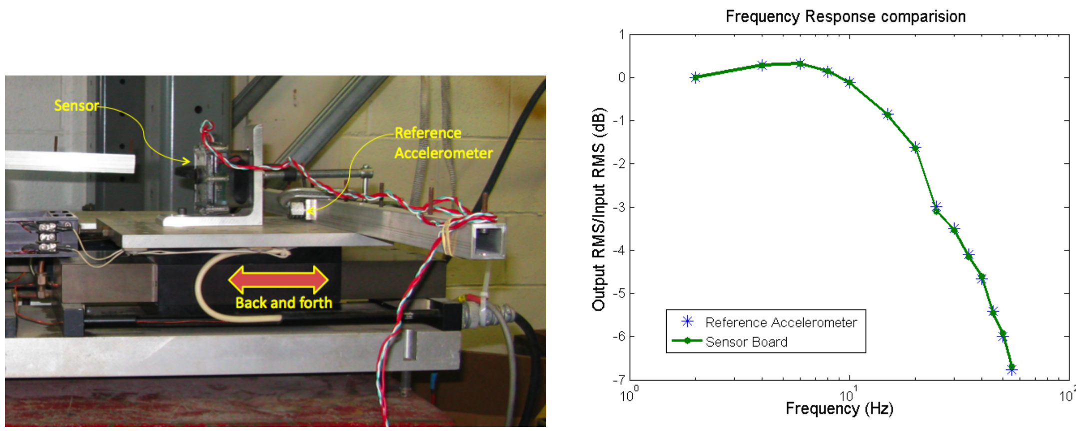


Figure 2.7: Measurement setup and Frequency response of the accelerometer sensor node and of reference accelerometer, corrected for the anti-aliasing filter.

length L and gage blocks, with known and calibrated heights. By stacking a series of gage blocks, we create a series of different heights h_j. By following the geometry suggested by the diagram, at each height j we measure the voltage output V of the accelerometer. We repeat this operation I times. The statistical model can be written as

$$
\begin{aligned}
V_{ij} &= A\sqrt{1-x_j^2} + B\,x_j + C + \epsilon_{ij},\\
x_j &= \frac{h_j}{L},\\
\alpha &= \sqrt{A^2+B^2},\\
A &= \alpha\, g\, \cos(\theta),\\
B &= \alpha\, g\, \sin(\theta),\\
\theta &= \theta_3 - \theta_1,
\end{aligned}
$$

where ϵ_{ij} is an independent identically distribute gaussian random variable, with zero mean. α is the desired sensitivity parameter. We performed six different experiments, and estimated the average sensitivity, using linear regression, as $\alpha = 2.00347\,V/g$ with a standard deviation of $0.0299\,V/g$. When using the sensor in the field we to infer C, since it is a voltage offset at the sensor, and it changes as we turn on and off the sensor system. The offset can be measured as a 2 second average of the signal when there is no input. We measure and subtract this offset. The field θ is assumed to be 0.

In Chapter 8 we use this sensor node to measure experimental responses of a roadway to a truck, and show how the data can be used to estimate the truck weight in an ideal scenario.

2.3.3 Data Aggregation and Processing

Urban traffic systems and road infrastructures are systems spread over a very large area. When constructing a sensing solution it is essential to consider how the data will

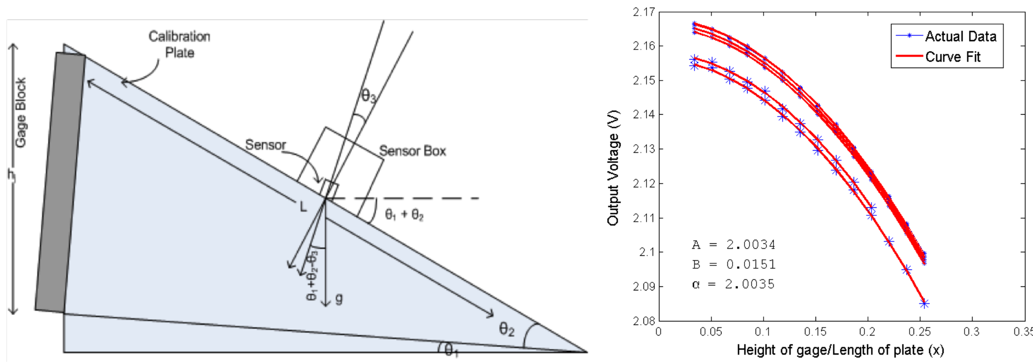


Figure 2.8: Measurement setup and estimated sensitivity model fit for various experiments.

be transmitted to a central location where analysis can be done. As an example, consider highway monitoring for a large state such as California. Each highway sensor produces one sample of speed, density and flow every 30 seconds, and there are 25,000 sensors. All these samples, or at least some pre-processed values, have to arrive in real-time to a central location. Since the sensors are spread out in an area as large as the state of California, it is unrealistic to expect that every sensing station communicates their samples to the central location using identical infrastructure choices.

In fact, data aggregation in large sensing systems happens through a heterogeneous communication medium. The most common medium are long range wireless, such as cellular networks, or a dedicated modem network. Typically, there can be several layers of aggregation forming a hierarchical model, where each level of the hierarchy aggregates data from the lower levels and communicates to the higher level.

Another important issue is that the sensing system itself may or may not have local memory and the ability to correct communication errors. It turns out that most sensors do not perform such error correction. The most common type of error is a dropped data packet corresponding to a single sampling interval for a single sensor. Any system for processing the received data at the central location should be able to handle such errors. If such errors can be handled by having a communication protocol that supports retransmissions and error correction, the performance of the data collection system is substantially improved. In Chapter 3 we present a study of the sensor network that monitors the highways in California. The empirical study identifies key variables to understand the performance of the network, both in terms of failures of the transducers, as well as communication failures. One of the observations is that a network of sensors with retransmissions receives a fraction of more than 90% of the samples generated, whereas in a sensor network without retransmissions this fraction is near 70%.

In Chapter 6 we also show another important consequence of imperfect communication: estimation algorithms are significantly slower to converge to the true estimates. Still with proper processing, imperfect communications can be taken into account.

Chapter 3

Measuring Reliability of a Large Sensor Network

3.1 Introduction

In this chapter we develop a methodology for empirically evaluating the reliability of a large sensor network. As a case study, we use the loop sensor network that monitors traffic for the state of California. The freeway *sensor network* of the California Department of Transportation (Caltrans) has two components: a sensor system and a communication network. The statewide sensor network is divided into twelve parts, each built, operated and maintained by one Caltrans District.

The statewide *sensor system* consists of 25,000 sensors located on the mainline and ramps, and grouped into 8,000 vehicle detector stations (VDS). Over 90 percent of the sensors use inductive loops, most of the remaining use radar detectors. The sensor system produces 30-second averages of vehicle occupancy and volume measured by each sensor.

The *communication network* transports data packets from each VDS to its District Traffic Management Center (TMC). Based on these data the District Advanced Traffic Management System (ATMS) makes decisions about traffic operations. A copy of each data packet is also sent to the freeway Performance Measurement System (PeMS), which archives the data and processes them in different ways to generate a variety of freeway system performance measures. The communication network is built out of communication links that employ different technologies. For example, wireless GPRS links predominate in District 4, whereas telephone land lines are widely used in Districts 7 and 11.

The sensor system in the largest district (District 7) covering Los Angeles and Ventura counties has 8,700 loop sensors; the nine-county San Francisco Bay Area District 4 has 4,600 loop sensors; and District 11, covering San Diego and Imperial counties, has 3,100 loop sensors. We study the sensor networks in three Districts, at first using data from PeMS.

PeMS expects to receive from each sensor one sample (packet) every 30 seconds. Based on the number and quality of the samples that it actually receives from a sensor on a given day, PeMS designates that sensor as 'good' or 'failed' for that day.

Figure 3.1 plots the percentage of *failed* sensors for each day from 10/10/2005 to 12/31/

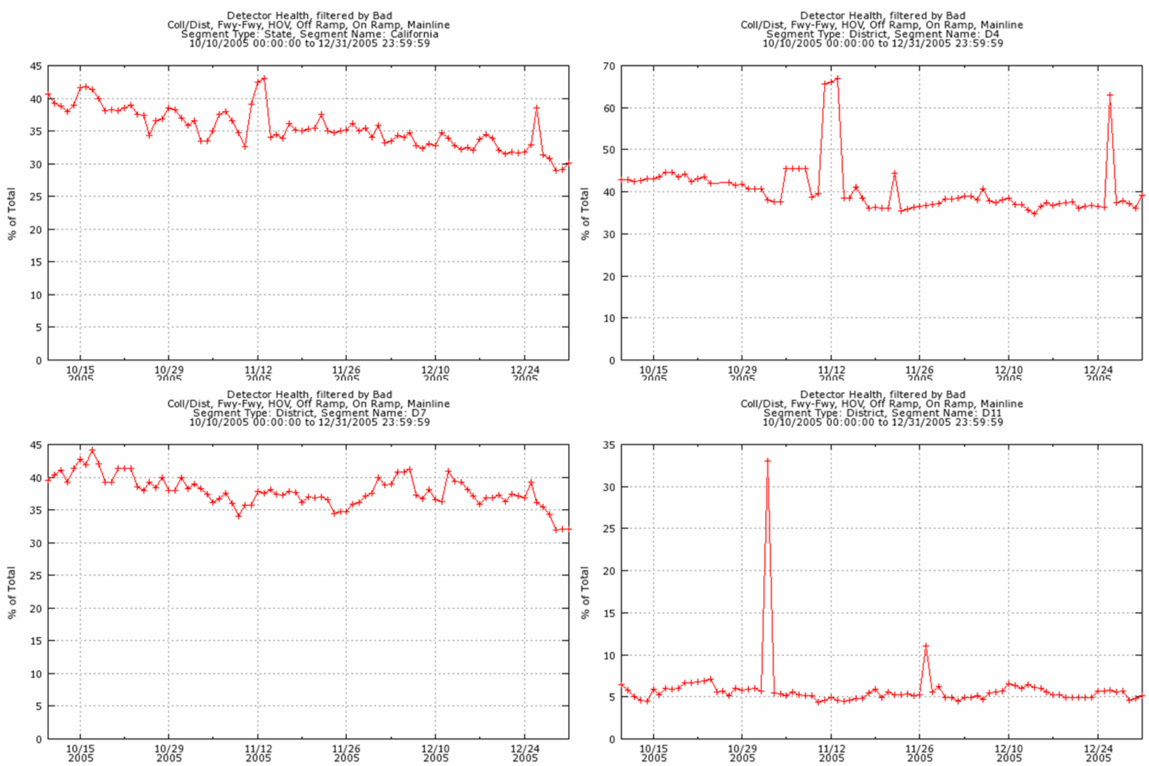


Figure 3.1: Daily fraction of failed sensors for the statewide system, and Districts 4, 7 and 11 from 10/10/2005 to 12/31/2005.

2005 for the whole state and for Districts 4, 7 and 11. The sensor network has very poor reliability, with 35 percent of sensors statewide considered failed on any given day. The reliability varies widely from district to district: District 4 had 40 percent, District 11 had 5 percent, and District 7 had 35 percent failed sensors.

The poor reliability of the sensor network, and the pressing need to use sensor data for better freeway operations decisions, led Caltrans to launch the Detector Fitness Program (DFP) beginning December 2005. The goal of the DFP was to significantly raise the reliability of the statewide sensor network. Over the next 12 months, crews made 9310 visits to sensors in the field in order to diagnose why they had failed and, if possible, to fix the failures.

Most sensors behave like light bulbs: once they fail, they stop functioning for ever. The freeway sensors are quite different: they repeatedly fail and then 'spontaneously' recover from failure, as is evident from the oscillations in Figure 3.1. Thus, metrics designed to measure the reliability of systems with light bulb-like failures cannot be used for the freeway sensor system. The chapter proposes four different ways of measuring the reliability of the freeway sensor system: productivity, stability, and lifetime and fixing time.

Productivity is the distribution of the fraction of days that sensors provide reliable measurements. *Stability* is the distribution of the frequency with which sensors switch from providing reliable measurements to becoming unreliable. *Lifetime* is the distribution of the number of successive days that sensors continue working before they fail, and *fixing time* is the distribution of the number of successive days that sensors remain in a failed state before being fixed. The chapter uses these metrics to compare the reliability of the sensor networks in Districts 4, 7 and 11, as well as to evaluate the effectiveness of the Detector Fitness Program.

The remainder of this chapter is organized as follows. Section 3.2 describes the sensor network in a way that shows the kinds of hardware and software faults that can lead PeMS to declare a sensor failure. Section 3.3 summarizes the two data sets that are used. Section 3.4 considers sensors that are always in a failed state. Section 3.5 introduces the scope chart, which gives a visual summary of the state of the sensor network. Section 3.6 defines productivity and computes the productivity of the three Districts. Section 3.7 defines and evaluates stability. Section 3.8 calculates the lifetime and fixing time distributions. Section 3.9 analyzes the Detector Fitness Program and evaluates its effectiveness. Section 3.10 collects some conclusions.

3.2 Sensor fault description and PeMS failure states

Figure 3.2 is a schematic of the sensor network in District 7. At a particular VDS location, there is a sensor in each lane of the freeway. In more than 90 percent of the locations the sensor is an *inductive loop*, represented by the little circles in the figure. Sensors from the different lanes are connected through a *pull box* to a *controller cabinet* on the side of the road.

The cabinet includes a *170 controller* and a *modem*. The controller *detector cards* process each sensor's measurements to produce 30-second averages of vehicle occupancy and volume, and format these data into a packet, which includes fields indicating the VDS and sensor IDs (identifiers). The cabinet receives power from a local power line.

The TMC receives the data packets from the controller over a digital *communication network*. The network has two parts, one of which is Caltrans-operated and the other is Telco-operated.

A Caltrans-operated *field line* connects the controller cabinet and modem to the Telco demarc box; optionally a *field bridge* connects multiple controllers to the Telco demarc box. A *Telco bridge* connects multiple demarc boxes to a *TMC Line* inside the Telco network. The TMC Line connects to the front-end processor (FEPT) of the District TMC. Up to 20 controllers may share the same Telco line, the different controllers being distinguished by a *Drop ID*.

The FEPT receives data by polling the controller modems. The received packets are forwarded to the District ATMS; a copy is also forwarded to PeMS.

Caltrans deploys several variations of the sensor network. A small fraction of the sensors use radar to detect the presence of a vehicle. However, the radar-based systems also produce data packets with the same format. There is a greater difference in the communication network. Some controller cabinets in District 7 are connected to the TMC over Caltrans-owned optical fiber links. More significant is the use of wireless links rather than land lines as in the figure. For example, District 4 uses the GPRS data service.

Thus the overall sensor network combines several hardware and software subsystems.

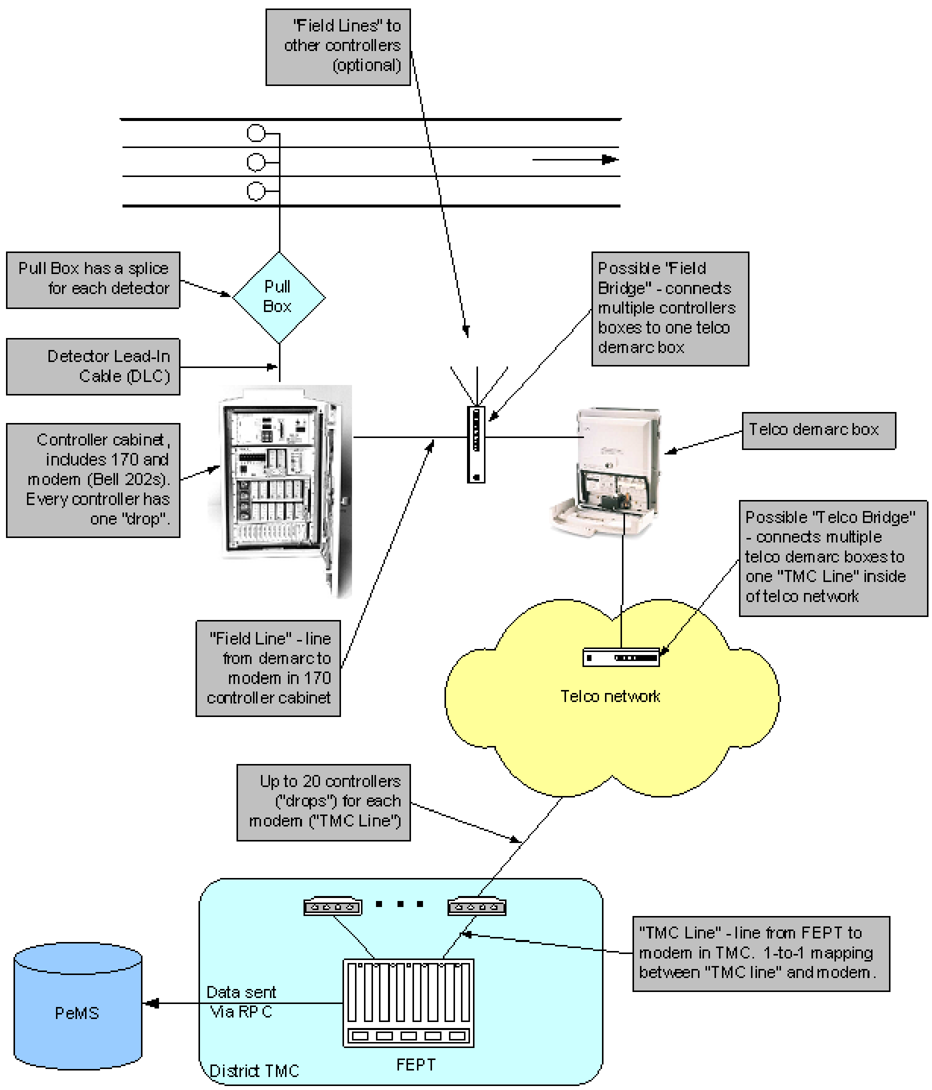


Figure 3.2: The configuration of the sensor network in District 7

Each subsystem is a potential source of failure. The main subsystems are the inductive loop; the detector card for each sensor; the controller; the Caltrans-operated communication sub-network; and the Telco-operated communication sub-network.

When PeMS receives a data packet, it consults a *configuration table* to interpret the data packet. The table contains meta information that helps determine whether the VDS and sensor IDs in the packet are valid and where the sensors are located (mainline, ramps). If a packet contains an ID that is not recognized by the table, the packet is discarded. Conversely, if there is no packet corresponding to an ID in the table, it is assumed that there is a failure in the system corresponding to that sensor.

Every midnight, PeMS examines the sequence of (data) samples received from each detector; subjects the sequence to a set of statistical tests; and classifies the detector 'health' for that day into one of 10 *diagnostic states* displayed in the first column of Table 3.1. A correctly functioning detector should daily provide 2,880 30-sec samples with reasonable values. The statistical tests involve the number of received samples and their values. The second column of the table indicates the nature of the tests, and a detailed description is available in Chen et al. [2002a].

Diagnostic State	**Description**	**Detector Types**
Line Down	No detector on the same communication line as the selected detector is reporting data. If information about communication lines is not available this state is omitted.	ML, Ramps
Controller Down	No detector attached to the same controller as the selected detector is reporting data. This may indicate no power at this location or the communication link is broken.	ML, Ramps
No Data	The individual detector is not reporting any data, but others on the same controller are sending samples. This may indicate a software configuration error or bad wiring.	ML, Ramps
Insufficient Data	Insufficient number of samples are received to perform PeMS diagnostic tests, while other detectors reported more samples.	ML, Ramps
Card Off	Too many samples with an occupancy (for ML and HOV detectors) or flow (for ramps) of zero. The detector card (in the case of loop detectors) is probably *off*.	ML, Ramps
High Val	Too many samples with either occupancy above 70% (for ML and HOV detectors) or flow above 20 veh/30-sec (for ramps). The detector is probably stuck *on*.	ML, Ramps
Intermittent	Too many samples with zero flow and non-zero occupancy. This could be caused by the detector hanging on.	ML
Constant	Detector is stuck at some value. (PeMS counts the number successive occurrences of the same non-zero occupancy value.)	ML
Feed Unstable	The data feed itself died and there were insufficiently many samples during the day to run the tests. On days where this occurs we mark the detectors that were previously good as good and the ones that were previously bad as Feed Unstable.	ML, Ramps
Good	Detector passed all tests	ML, Ramps

Table 3.1: Diagnostic states

The first nine diagnostic states indicate failure; the tenth state, 'good', indicates a functioning detector. The plots in Figure 3.1 refer to the daily fraction of failed sensors.

Comparing the configuration of the sensor network in Figure 3.2 with Table 3.1 we see that knowledge of the sample sequence received by PeMS from a particular sensor is not enough to uniquely relate a failure diagnostic state with an actual hardware or software fault. For instance, if a sensor delivers insufficiently many samples or no samples at all, this may be due to the controller being down or to a failure of a communication link or an error in the configuration table. Therefore we will aggregate the 10 diagnostic states provided by PeMS into two or three 'macro' states: *good*, *sensor system failure*, and *communication network failure*.

3.3 Data Used

We describe the data used in the chapter and then the pre-processing steps taken to convert the data into a standard format used in the subsequent analysis. Two data sets are used.

The first data set consists of the sequence or time series of daily sensor diagnostic states in PeMS for each loop in Districts 4, 7 and 11 as described in Table 3.1.[1] The loops considered in the study are those that were listed in the PeMS configuration table on March 31, 2007. For each loop the sequence of days spans 27 months from January 1, 2005 to March 31, 2007; for loops that were installed on a later date, the sequence begins later. There were 5782 sensors in District 4, 8707 sensors in District 7 and 3264 sensors in District 11.

The second data set comprises records from the Detector Fitness Program (DFP) for Districts 4 and 7 [Rajagopal and Varaiya, 2007]. These records were created by crews following a field visit to a loop. The records are textual and their format is not standardized; hence they require some interpretation on our part, as explained next.

For District 4, the records summarize a total of 4,578 visits between December 12, 2005 and December 30th, 2006. 3244 individual detectors were visited. For District 7, the visits occurred in clusters between December 12, 2005 and August 2, 2006. The records corresponds to a total of 4,732 visits. 3192 individual detectors were visited, implying that several detectors were visited more than once. For different clusters, the data are recorded in different ways on a spreadsheet, possibly because there were different crews. Each row of the spreadsheet corresponds to a visit to an individual loop. Typically, the records contain the following fields:

- Location: comprising the Detector Station (VDS) to which the loop is connected, lane number, highway name, highway direction, and postmile.
- Visit date: typically the date the loop was visited, but sometimes the date the record was entered in the system. It is safe to say that the sensor was visited before this date.
- Problem type: typically a textual description of either the diagnostic state reported by PeMS or the type of problem encountered.
- Related cause: typically the failure cause as surmised by the crew, frequently the name of a broken or missing hardware part.
- Solution: typically the steps taken towards the solution of the problem, and whether the problem was successfully resolved.
- Status: the PeMS diagnostic state after visit.

[1] We use 'sensor', 'loop' and 'detector' interchangeably.

3.3.1 PeMS data pre-processing

Let the time series $s_i(n)$ denote the diagnostic state as determined by PeMS for sensor i on day n; $s_i(n)$ takes one of the 10 values listed in the first column of Table 3.1. We merge several diagnostic states together to obtain a new time series s_i^*:

$$s_i^*(n) = \begin{cases} 1 & \text{if } s_i(n) \in \{\text{Good}\} \\ 0 & \text{if } s_i(n) \notin \{\text{No Data,Controller Down, Good}\} \\ -1 & \text{if } s_i(n) \in \{\text{No Data,Controller Down}\} \end{cases} . \tag{3.3.1}$$

In some cases (as will be made clear) we modify the conditions above to

$$s_i^*(n) = \begin{cases} 1 & \text{if } s_i(n) \in \{\text{Good}\} \\ 0 & \text{if } s_i(n) \notin \{\text{No Data, Insufficient Data, Controller Down, Good}\} \\ -1 & \text{if } s_i(n) \in \{\text{No Data, Insufficient Data, Controller Down}\} \end{cases} . \tag{3.3.2}$$

The aim of the coding scheme is this. $s_i^*(n) = 1$ means that both the sensor system and the communication network associated with loop i were functioning on day n. $s_i^*(n) = 0$ means that the communication network associated with loop i on day n was functional, since some packets were received ($s_i(n) \neq$ No Data), but there was a failure in some part of the sensor system. Lastly, $s_i^*(n) = -1$ corresponds to a communication network failure as no data were received. Sometimes, following (3.3.2), 'Insufficient Data' is treated as a communication failure.

The aim of the analysis is to understand the persistence of the 'good' state and the occurrence of the two failure states. Note that by using (3.3.1) more failures are attributed to the sensor system, whereas by using (3.3.2) more failures are attributed to the communication network.

If there is a 'communication network failure' ($s_i^*(n) = -1$), we cannot say whether the sensor system has also failed, because a communication failure masks or *censors* the corresponding sensor system observation. How can we estimate the sensor system state when there is a communication failure?

A simple approach, in keeping with the non-parametric approach we have adopted here, is to fill in the sensor system failure value with its last known value. Thus if $s_i^*(n) = -1$, we set $s_i^*(n) = s_i^*(n-1)$. If $s_i^*(n) = -1$ on the first day of the series for loop i we ignore the series until the very first day for which $s_i^*(n) \neq -1$. We call the new resulting sequence a *filled* sequence. A *filled* sequence is one in which {No Data, Controller Down} are filled. A *ND-filled* sequence is one in which states {No Data, Controller Down, Insufficient Data} have been filled. The difference lies in the interpretation of the communication failure 'insufficiently many samples received' which can be interpreted as either a sensor system failure, or a communication network fault.

3.3.2 Detector Fitness Program data pre-processing

The DFP maintenance records do not follow a uniform format. The recording frequently was not very careful. For example in District 4, 25% of the records report no underlying cause of failure. In District 7, only 4% of the records suffer from this deficiency. Such recording procedures pose an additional challenge to the data analysis.

For each visit to a sensor an entry in the maintenance record is created and the observed failure and actions taken are recorded. No systematic way of recording the observed failures was followed. The cause for the observed failure and the actions taken are also encoded in a single sentence, such as 'open loops SB lane 3. disabled channels'. Therefore we used a simple parsing scheme to encode the textual records. We created 9 non-mutually exclusive classes: *Upgrade Firmware*, *Under Construction*, *Open/Bad Loop*, *Connection Issues*, *Modem/Card Issues*, *Reset Equipment*, *No Power*, *Other Issues* and *No reported cause*. For each class we seek specific keywords or combination of keywords in the text. If the keywords are observed a '1' is entered for that class for that record; otherwise a '0' is entered. We manually checked many of the assignments made in this way and they worked reasonably well, because failure descriptions use a much smaller vocabulary than freeform text.

Thus for each visit we end up with 13 variables: the sensor visited, the visit date, whether the sensor was fixed or not, and an indicator of 9 possible non-exclusive failure causes.

3.4 Always-failed and Always-working Sensors

Examination of the PeMS diagnostic sequences reveals a significant fraction of sensors that are always failed (i.e., never worked) or always working. We begin by analyzing the statistics of such sensors. Sensor i is called **always-0** if the sensor is assigned a failed state for the entire period T, i.e., $s_i^*(n) = 0, n \in T$. It is called **always-1** if a failed state is never observed for the entire period T, i.e., $s_i^*(n) = 1, n \in T$. T is taken to be one quarter or one year. (See (3.3.1) for definition of s_i^*.)

We use filled sequences to count **always-0** and **always-1** sensors, unless explicitly indicated otherwise. This means that communication failures are *not* considered to be faults for this analysis. The type of filled sequence does not affect the **always-0** status, but it does affect **always-1** status, because if 'Insufficient Samples or Data' is regarded as a communication failure, some sequences that include 0 values can become **always-1** .

Table 3.2 shows the number and percent of **always-0** and **always-1** sensors in the three different Districts. There is a large number of **always-0** sensors in District 4 and District 7 compared to District 11. This discrepancy by itself accounts for a considerable portion of the performance difference between Districts.

Furthermore, District 4 and District 7, in contrast with District 11, have almost no **always-1** sensors. The increase of **always-1** sensors from 2005 to 2006 in District 7 is due in part to sensor misconfigurations that were corrected and in part to the DFP. The increase in **always-0** sensors in District 11 is mainly due to the inclusion of ramp sensors in PeMS starting in 2006.

If we use an ND-filled sequence, which classifies 'Insufficient Data' as a communication failure, then the number of **always-1** sensors increases while the number of **always-0** sensors remains the same, as expected (Table 3.2). The increase in the number of **always-1** sensors assumes that the sensor system reliability is unchanged during a communication failure. This indicates that at least part of the performance difference among Districts can be attributed to communication network failures.

District (Year)	Total	**always-1**	**always-0**
District 4 (2005)	5140	9 (0%)	1171 (23%)
District 4 (2006)	5271	0 (0%)	1327 (25%)
District 7 (2005)	6478	21 (0%)	1090 (17%)
District 7 (2006)	8613	319 (4%)	1399 (16%)
District 11 (2005)	1750	604 (35%)	38 (2%)
District 11 (2006)	3223	1402 (44%)	116 (4%)
District 4 ND (2006)	5271	2106 (40%)	1327 (25%)
District 7 ND (2006)	8613	2590 (30%)	1399 (16%)
District 11 ND (2006)	3223	2544 (79%)	116 (4%)

Table 3.2: Failure summary for always failed and always working sensors filled and ND-filled sequences (ND)

Rajagopal and Varaiya [2007] develops a further breakdown of **always-0** and **always-1** sensors according to type of sensor, highways, lanes and VDS controllers.

3.5 System View

In this section we introduce a view—the *scope chart*—of the sensor system state, based on visualizing the fault sequence over time and across highways. This visualization technique provides a global view of the system or of parts of the system.We first describe how such plot is constructed.

For each sensor i, compute the state sequence $s_i^*(n)$, which assumes values 1 (sensor is good on day n), -1 (communication network failure on day n) and 0 (sensor system failure on day n) (see (3.3.1)). The plot is a two-dimensional 'heat' map (1 = red, -1 = blue, 0 = green). The horizontal axis is time in days. (The sequences cover 27 months or 810 days.) The vertical axis corresponds to some ordering of all sensors. In Figure 3.3 for District 4 all sensors on the same highway are grouped together and within each highway group they are ordered by postmile and lanes.

In the chart we can clearly see horizontal red lines representing sensors that worked for long periods. A blue streak in the horizontal direction indicates a sensor that did not report data for a long period. Blue streaks in the vertical direction correspond to days when many sensors sent no samples. This could be caused by a communication network failure in which several TMC lines failed or the FEPT was unable to poll many modems (see Figure 3.2). Such streaks explain the oscillations observed in the total number of failed sensors in Figure 3.1. The scope chart also allows us to compare the reliability of different highways. The charts suggest that in general District 11 has a much more reliable sensor system. In particular, there are fewer communication failure streaks in District 11 than in District 4 (the blue streaks at the leftmost side of the chart usually corresponds to dates before the sensor was installed into the system). This reinforces the importance of the communication network between the controller modems and the FEPT.

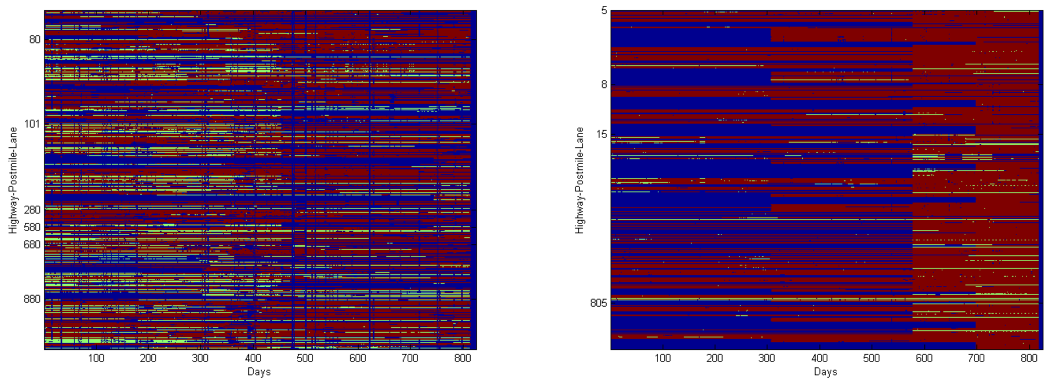


Figure 3.3: Scope chart ordered by highway, postmile and lane for Districts 4 (left) and District 11 (right), 2005-2007. Red streaks corresponding to Good state, green to Bad and blue to Communication network failure.

3.6 System Productivity

In this section we propose a measure of productivity of a District's sensor network. The measure is computed as follows. Consider a time interval T and a sensor set $\mathcal{M}$ of size M. For each sensor $m \in \mathcal{M}$ we calculate the percent of days d_m that the sensor is working as $w_m = 100[d_m/T]$. The *productivity* of $\mathcal{M}$, $P_{\mathcal{M}}(x),\ x \in [0, 100]$ is the cumulative frequency distribution of w_m:

$$P_{\mathcal{M}}(x) = \frac{1}{M}\sum_{m=1}^{M} I\left(w_m \leq x\right). \tag{3.6.1}$$

$P_{\mathcal{M}}(x)$ is the fraction of the sensors that worked for at most x% of days. Evidently, sensor set $\mathcal{M}_a$ has *strictly better productivity* than $\mathcal{M}_b$ if $P_{\mathcal{M}_a}(x) < P_{\mathcal{M}_b}(x)$ for all x. A single number to compare two sensor sets is the *total productivity* (TP) defined as the area above the productivity function,

$$\mathrm{TP}_{\mathcal{M}} = 1 - \int_0^{100} P_{\mathcal{M}}(x)dx, \tag{3.6.2}$$

which of course is the empirical average of w_m,

$$\mathrm{TP}_{\mathcal{M}} = \frac{1}{M}\sum_{m=1}^{M} w_m. \tag{3.6.3}$$

If we model the sensor state as a two-state ('good' and 'failed') stationary Markov chain, TP is the steady-state probability of the chain being in the 'good' state.

We compute the productivity of the sensor networks in Districts 4, 7 and 11, using the raw (non-filled) data sequence (3.3.1). We omit all sensors that are **always-0** for the chosen time horizon since these sensors have already been accounted for.

Figure 3.4 displays the results. For any point on the curve take the y-ordinate (say 20%), determine the corresponding x-ordinate (say 40 days), and interpret the point to mean 20%

of the sensors worked for less than 40% of the time. Alternatively, 80 % (100%-20%) of the sensors worked for more than 40% of the time. The total productivity of the sensor network is the area above the productivity curve.

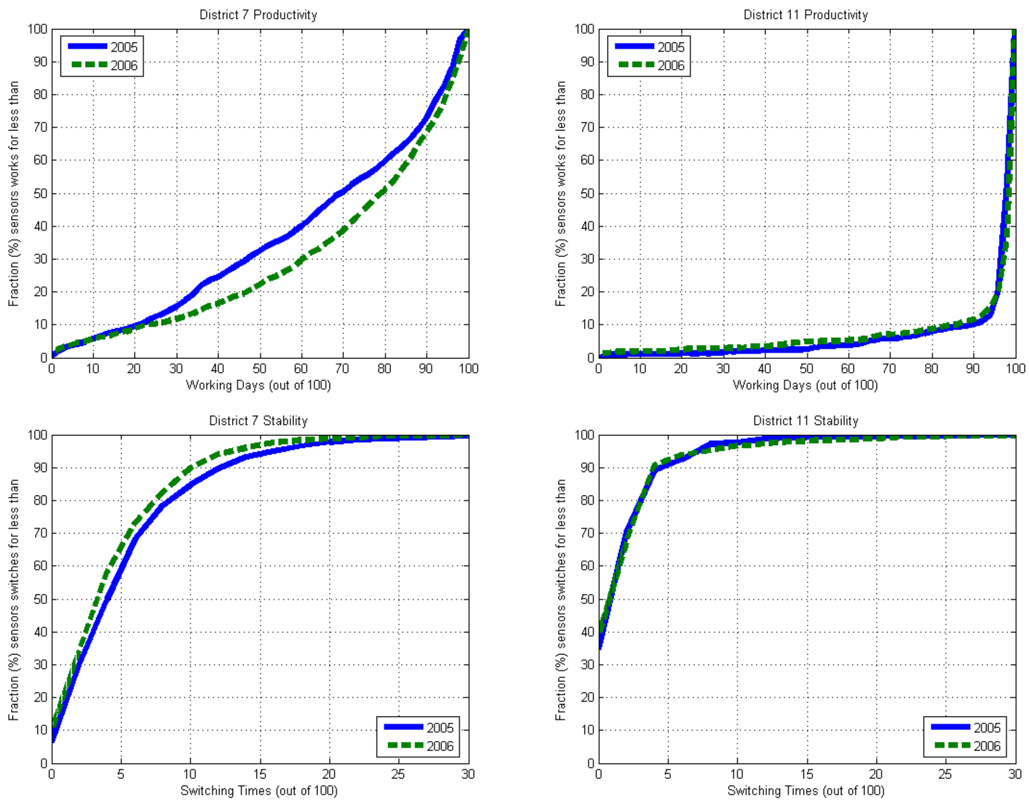


Figure 3.4: Productivity of District 7 (top left) and District 11 (top right). Stability of Districts 7 (bottom left) and 11 (bottom right), 2005 and 2006

For District 7, productivity in 2006 is strictly better than in 2005, presumably a result of the Detector Fitness Program (DFP). For District 11 productivity remained unchanged, and is strictly better than the productivity of both Districts 7 and 4 (not shown). For District 4, the median productivity for both years was unchanged. The effect of the DFP for District 4 is mixed.

Productivity for the districts aggregated by highways and lanes does not show extreme variability [Rajagopal and Varaiya, 2007].

3.7 System Stability

From Table 3.2 we know that the majority of sensors switch between good and failed states one or more times. (These are the sensors that are not **always-0** or **always-1** .) Sensors with the same productivity may switch different number of times. We propose a simple system metric that captures this difference.

For a sensor set $\mathcal{M}$ of size M and time interval T, we compute the normalized number of state changes $s_m = (r_{10,m} + r_{01,m})/T$, where $r_{10,m}$ is the number of times sensor m switches from the good state to a failed state during T and $r_{01,m}$ is the number of switches from a failed to the good state. The *stability* of $\mathcal{M}$, $S_{\mathcal{M}}(x),\ x \in [0, 100]$ is the cumulative distribution of s_m:

$$S_{\mathcal{M}}(x) = \frac{1}{M}\sum_{m=1}^{M} I\left(s_m \leq \frac{x}{100}\right). \tag{3.7.1}$$

$S_{\mathcal{M}}(x)$ is the fraction of sensors that switched states on at most $x\%$ of the days. The *total stability* TS is the area below $S_{\mathcal{M}}(x)$:

$$\mathrm{TS}_{\mathcal{M}} = \int_0^{100} S_{\mathcal{M}}(x)dx. \tag{3.7.2}$$

Sensor set $\mathcal{M}_a$ is *strictly more stable* than a set $\mathcal{M}_b$ if $S_{\mathcal{M}_a}(x) > S_{\mathcal{M}_b}(x)$ for all x. $\mathcal{M}_a$ is on average more stable than $\mathcal{M}_b$ if $\mathrm{TS}_{\mathcal{M}_a} > \mathrm{TS}_{\mathcal{M}_b}$. If we model the sensor state as a stationary two-state Markov chain, its two transitional probabilities are determined by its total productivity and total stability. Stability is a measure of how frequently individual sensors switch between working and failed states.

Using the raw data, we estimate the stability of different Districts. We discard sensors that are **always-0** , as they were considered separately. The average stability of the system is just the area below the stability distribution curve.

Figure 3.4 compares the stability of District 7 and District 11 for 2005 and 2006. For a point on the plot, suppose its y-ordinate is 50% and the corresponding x-ordinate is 5. This means that 50% of the sensors switched 5 or fewer times during a 100 day period. In the figure, District 7 is more stable in 2006 than in 2005, as the stability curve in 2005 strictly dominates 2006. The median number of switches decreased from 4 in 2005 to 3 in 2006. For District 11 the median remains at 1 for both years, with a large number of sensors with 0 switches (**always-1** sensors). District 4 has a median of 7 switches for 2006 and 5 for 2005. Comparison across highways and lanes did not reveal any extreme differences for all three districts [Rajagopal and Varaiya, 2007].

3.8 Lifetime Estimates

Estimation of *lifetime* or *survival* curves is the standard approach in statistics for characterizing system failures [Nikulin, 2004; Klein and Moeschberger, 2003]. In this approach a number of individuals are observed starting at varying initial times and their failure times are recorded. Records of individuals that did not experience failures during the observation period will be right-censored as we don't know when they would have failed. The survival curve is the complement of the cumulative distribution of time to failure. The standard non-parametric estimators of the survival curve are the Nelson-Aalen and Kaplan-Meyer estimators [Nikulin, 2004; Klein and Moeschberger, 2003]. These estimators are appropriate only for individuals experiencing a permanent failure rather than recurring failures.

In the California sensor network, many failed sensors 'spontaneously' start working again, which is different from the standard survival analysis setting. In the sensor network literature as well recurring failures are usually ignored, but it is an important phenomenon that should be understood [Koushanfar et al., 2003; Zhou and Guo, 1998]. Spontaneous failure and recovery processes could indicate that the loss of performance is not a result of failures in the underlying hardware (which are likely to be permanent), but is rooted in the design choices for the communication network and sensor unit.

We use simple estimates of survival curves, which account for spontaneous recovery. Choose a time period T. The data comprise filled or ND-filled sequences. For each sensor i, compute the runs of 0's and 1's. A *0-run* is the count of the number of successive days a sensor is failed ($s_i(n) = 0$), i.e. *time to fix*; a *1-run* is the count of the number of successive days $s_i(n) = 1$, i.e. a sensor remains in a working state (*lifetime*). Each sensor's 0-runs and 1-runs alternate. Denote the set of 0-runs and 1-runs for sensor i by $\mathcal{R}_i^0$ and $\mathcal{R}_i^1$ respectively. We normalize all run lengths by the total number of days the sensor is in the system during T.

The first estimate is the *lifetime distribution*, which is the empirical cumulative distribution function for $\mathcal{R}^1 = \bigcup_{i \in \mathcal{A}} \mathcal{R}_i^1$, while the *mean lifetime* of sensor i is

$$\mu_1(i) = \frac{\sum_{r_i \in \mathcal{R}_i^1} r_i}{\left|\mathcal{R}_i^1\right|}. \tag{3.8.1}$$

The second estimate is the *fixing time distribution*, which is the empirical cumulative distribution function for $\mathcal{R}^0 = \bigcup_{i \in \mathcal{A}} \mathcal{R}_i^0$, while the *mean fixing time* of sensor i is

$$\mu_0(i) = \frac{\sum_{r_i \in \mathcal{R}_i^0} r_i}{\left|\mathcal{R}_i^0\right|}. \tag{3.8.2}$$

$\mu_1(i)$ is the average time sensor i is working before it fails and $\mu_0(i)$ is the average time it takes to become fixed after it has failed.

We can compute the empirical distributions of the mean lifetime $\mu_1(i)$ and the mean fixing time $\mu_0(i)$. In these distributions, each sensor contributes a single number. The difference between the 1-run distribution and the mean lifetime distribution, is that the former represents a system property (for example, sensors that are **always-1** contribute less to the distribution, as they have a smaller number of runs), whereas the latter is a distribution of the lifetime property of individual sensors.

3.8.1 Runs distributions

The 1-run distribution for District 11 is strictly better than that for District 4 (Figure 3.5), implying that sensors in District 11 keep working much longer before they fail. There is no significant change year over year for all districts. The 0-run distributions in Figure 3.5 show there is a large number of 1 day long failures, 61% in District 11, 48% in District 4 and 49% in District 7 (not shown). The run distribution computed according to ND filled sequences reveals that 42% of the 0-runs are one day long in 2006 in District 4, 50% for District 7 and 33% District 11, thus the one day long spontaneous failures are not limited to insufficient number of samples received. But, the 1-runs distribution also shows that District 4 and 7 behaves much closer to that of District 11, when such faults are excluded (see Rajagopal and Varaiya [2007]).

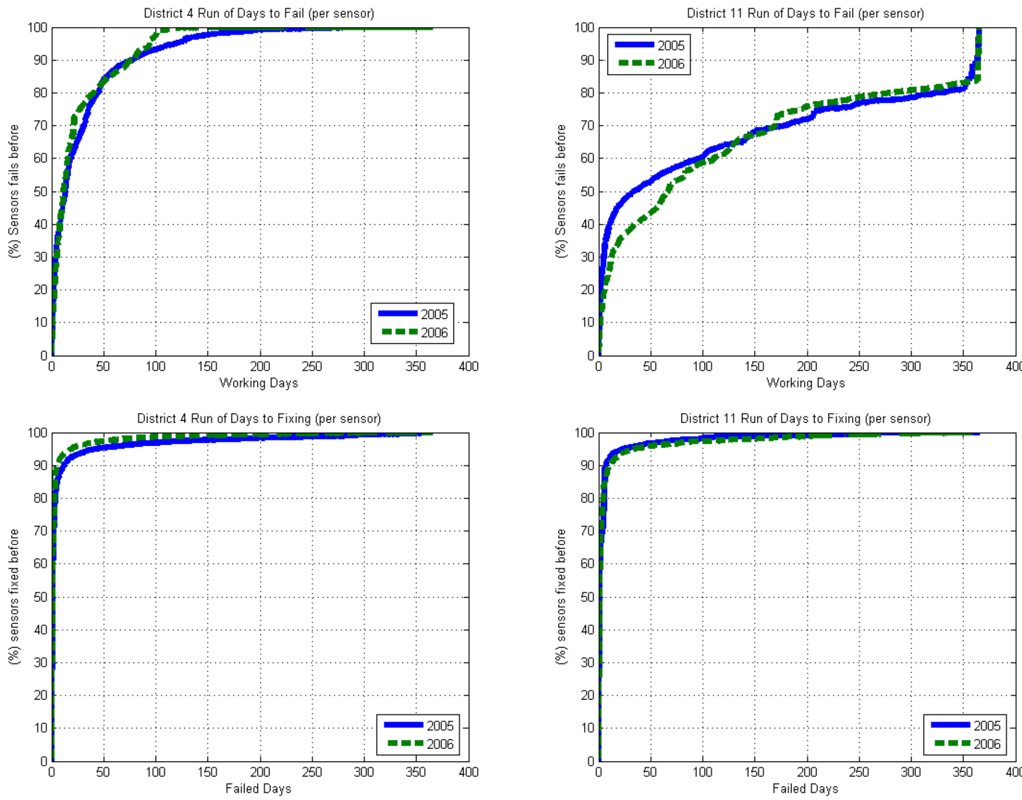


Figure 3.5: 1-runs distribution of District 4 (top left) and 11 (top right); 0-runs distribution of District 4 (bottom left) and 11 (bottom right) (2005-2006)

3.8.2 Lifetime and Fixing time

Figure 3.6 shows the mean lifetime distributions for sensors in Districts 4 and 11. District 4 shows some improvement in 2006, but District 11 is more productive due to the **always-1** sensors. The fixing time curves for District 11 is slightly better than for District 4, although both district have a large number of short failed bursts. This corroborates the results in previous sections.

Rajagopal and Varaiya [2007] shows that when ND-filled sequences are considered, Districts 4 and 11 perform similarly with 60% and 70% of the sensors, respectively, having **always-1** runs. This suggests that communication failures play a major part in the failure states of sensors.

3.9 Detector Fitness Program

The Detector Fitness Program for Districts 4 and 7 is an attempt to improve the reliability of their sensor networks. The Program sent crews to fix sensors that were suspected on

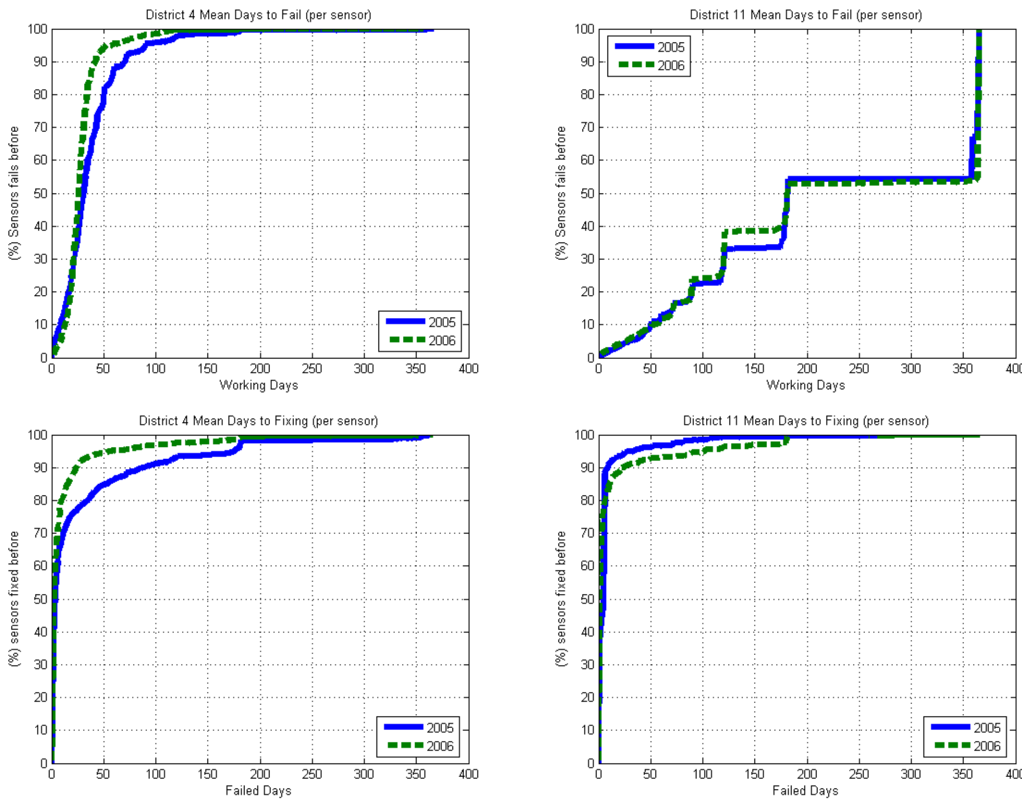


Figure 3.6: Sensor lifetime (top) and fixing time (bottom) distribution of District 4 and 11 (2005-2006, filled)

the basis of their PeMS diagnostic state *for a single day*. We have seen above that the sensor network in these Districts is very unstable. Hence it is a poor idea to determine the suspect list on the basis of a single day, especially if the failed state is due to a communication failure.

Only 1083 out of 3244 visited sensors (33%) in District 4 and 1651 out of 3192 (52%) in District 7 were *claimed* fixed. The number for District 7 is higher because the crew could replace the loop in some locations. Thus the DFP records claim a 'success' rate between 30 and 50%. We will see below that this claim is illusory.

An analysis reveals that for District 4 (District 7) 12.5% (21.6%) faults are 'Bad/Open Loops', 15.1% (17.3%) are 'Missing Parts', 26.2% (40.1%) are 'Modem/Card issues', 5.6% (7.9%) are locations under construction and 4.0% (8.0%) are 'No Power'. These are not mutually exclusive classes. Notice the significant number of non-operational loops which may report samples depending on district wide software configuration decisions. Careful maintainenance of the configuration should improve sensor network reliability.

3.9.1 always-0

About 30% of the visited sensors for both districts were **always-0** sensors, 867 sensors in District 4 and 958 in District 7. The total corresponded to 65% and 68% of existing **always-0** sensors and thus the analysis might apply to the entire population of **always-0** sensors. 22% of these sensors in District 4 and 26% in District 7 were sensors with bad/open loops. 43% of the sensors in District 4 and 15% in District 7 were reported to have bad equipment. Non existent lanes corresponded to 8.5% and 14.3% of the sensors. Only 9% in District 4 and 9% in District 7 were claimed fixed by the DFP. After the fixing date, 40% of the claimed fixed sensors (29 of 73) sensors in District 4 returned to an **always-0** state, and 24% (20 of 84) of those in District 7 returned to an **always-0** state. The real effectiveness of the program among **always-0** sensors is only about 5%.

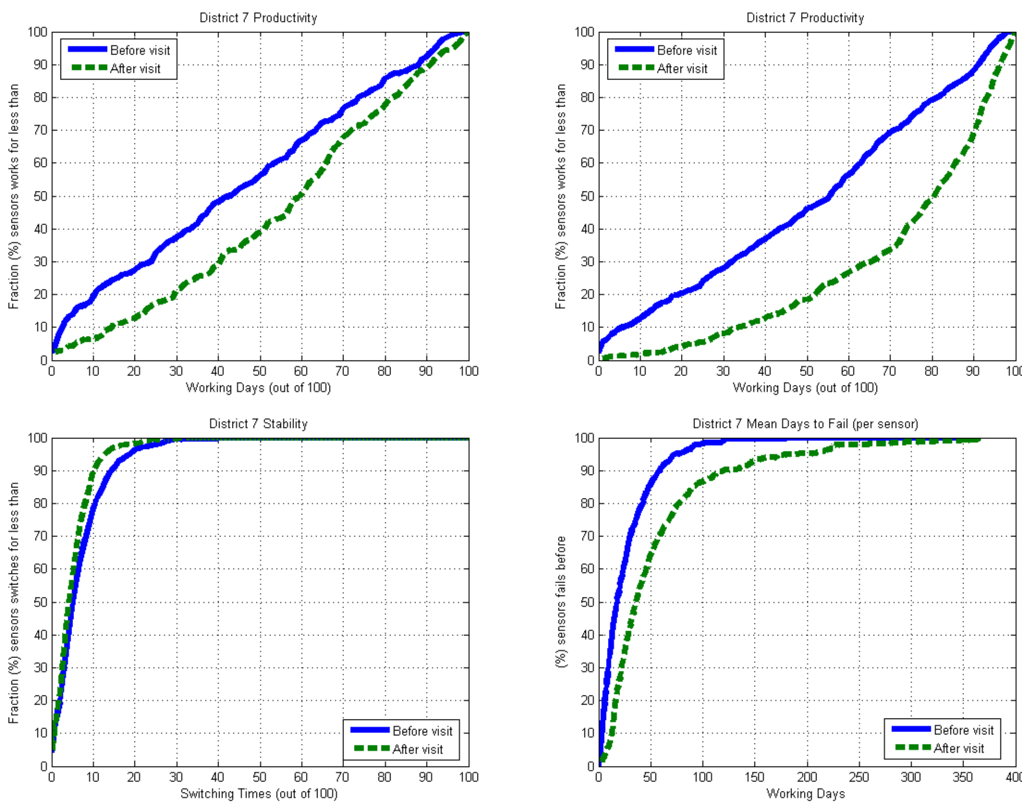


Figure 3.7: Productivity of visited but **not** fixed (top left) and visited and fixed (top right) sensors. Stability (bottom left) and lifetime (bottom right) of visited and fixed sensors in District 7 before and after visit (2005-2007)

3.9.2 Productivity and stability

Figure 3.7 compares the productivity of visited sensors claimed fixed and those visited but not fixed in District 7. The productivity was calculated for the time period before the visit and after the visit. A clear improvement can be observed due to fixing. The non fixed sensors experience a slight improvement that could be caused by misreporting, since in some cases the fixed status was checked by looking at a single day of PeMS status information. The Figure also shows the stability of the sensors claimed fixed, and the improvement is insignificant. This confirms the conclusion that that communication network failure is an "independent" failure, which the DFP does not effectively address.

3.9.3 Lifetime and Fixing Time

To conclude the section, we compute the sensor lifetime curve for visited and fixed sensors in District 7 (Figure 3.7). Fixed sensors improved their average length of 1-runs. Still the improvement is not as remarkable as the improvements observed in productivity, due to the inherent instability of the communication network. In Rajagopal and Varaiya [2007], the 0-run length distribution is also computed, and it confirms that the 0-runs become shorter, indicating that they are oscillations due to the communication network. Typically, long 0-runs correspond to actual failures, and short 0-runs to communication faults.

3.10 Discussion

In this Chapter we performed a systematic analysis of failures and the actions taken against them in two districts in California. Our analysis did not rely in any specific parametric models, avoiding any particular assumptions about the sensor behavior. Instead we devised simple metrics that can be easily computed for very large systems. We use a whole day (or block) of samples to attribute a sensor state.

We observed that four independent properties characterize well a sensor system: productivity, stability and lifetime/fixing time. Productivity is a measure of the performance of the sensor system, whereas stability captures the performance of the communication system. Lifetime and fixing time provide a peek at effective performance of the system. Simple approaches for censored inference generated good results.

Some observations about the districts in this study are that permanently failed sensors are the biggest influence in district wide performance. Furthermore, District 11 has a better communication system performance than Districts 4 and 7.

The Detector Fitness Program addressed the productivity of the detector system, but had limited success. Sensors for fixing should be chosen using the proposed metrics. The program should focus on sporadically working sensors and those with modem issues.

Chapter 4

Simultaneous Sequential Fault Detection for Multiple Sensors

4.1 Introduction

A randomly time-varying environment is monitored by a group of sensors. Each sensor has a fixed location where it periodically collects a noisy sample of the environment. A sensor may fail at any time, after which it reports incorrect measurements. Based on the sensor reports we wish to identify which sensors have failed and when the faults occurred.

If a failed sensor reports measurements with implausible values the fault can be correctly and quickly identified; but if it continues to report plausible values, fault detection is more difficult. We propose fault detection algorithms for this difficult case. The intuitive idea underlying the algorithms is for each sensor to detect a change in the correlation of time series of its own measurements with those of its neighbors' measurements. We call this *change point detection*.

In order for the idea to work, we make two assumptions. First, the measurements of functioning neighboring sensors must be correlated, while the measurements of a faulty sensor and a neighboring functioning sensor are not correlated. Second, since the environment being monitored is time-varying and the measurements are noisy, we require the average time between successive faults to be longer than the *event time scale*–the time between significant changes in the environment. The first assumption helps identification of a faulty sensor by comparing its measurements with its neighbors. Since the identification is made through statistical correlations, the probability of an incorrect fault identification (probability of false alarm) will be positive. The second assumption implies that a change in the environment can be distinguished from a change in the status of sensors, and also that there is sufficient time to reduce the false alarm probability at the cost of a delay in identifying when the fault occurred.

As a concrete example consider the California freeway performance measurement system or PeMS, comprising a collection of 25,000 sensors, one per lane at 9,700 locations [PeMS, 2009]. Every 30 seconds, a sensor reports the number of vehicles that crossed the sensor and the average occupancy or density (vehicles per meter) in the preceding five minutes. If no sensor has failed, these reports are directly used to generate a real-time traffic map on the PeMS website. On any day, however, upwards of 30 percent of the sensors have failed. PeMS uses statistical algorithms to identify the failed sensors and generate the traffic map without their measurements [Chen et al., 2003]. These algorithms rely on correlating each sensor's measurements with those of its neighbors but, unlike the approach here, they do not use temporal correlation. Also, PeMS algorithms are centralized, whereas ours are distributed as measurements are only communicated among neighbors.

We summarize our contribution. Section 4.2 reviews work related to our contribution. Section 4.3 proposes a change point distributed fault model for multiple faults, together with performance metrics to evaluate any sensor fault detection method.

Section 4.4 presents a distributed, online algorithm for simultaneously detecting multiple faults. The detection procedure relies on online message passing of detection results only among neighboring sensors. Section 4.4 also gives performance guarantees of the proposed algorithm in terms of the probability of false alarm (PFA) and the detection delay between the instant a fault occurs and the time when the algorithm detects the failure.

Sections 4.5 and 4.6 consider the selection of event time scales and propose efficient implementation schemes that minimize the amount of data transfer. Section 4.7 analyzes node density and fault detection tradeoffs.

4.2 Related Work

4.2.1 Fault detection in sensor networks

There is a sizable literature on fault detection in the context of sensor networks [Chong and Kumar, 2003]. Fault detection of multiple sensors has received some attention [Koushanfar et al., 2004]. An algorithm to increase the reliability of a 'virtual' sensor by averaging values of many physical sensors in a fault tolerant manner is presented in Marzullo [1990]. The analysis assumes that each sensor measures the same physical variable with a certain uncertainty and fault specification. In Ould-Ahmed-Vall et al. [2007], the authors develop a fault tolerant event detection procedure based on the assumption that time-varying failure probabilities of each node are known and a threshold test is used for detection. They also use geographical information to enhance spatial event detection. Decisions are made using only the current time observations, without accounting for trends in the data. Luo et al. [2006] proposes a similar model. Elnahrawy and Nath [2004] describes a method for outlier detection based on Bayesian learning. The procedure learns a distribution for interval ranges of the measurements conditional on the neighbor's interval ranges and last observed range. Neighbor's information and past information are assumed conditionally independent when the current range is observed. The idea of detecting malfunctioning sensors based on correlation-type reliability scores among the neighboring sensors is considered in Kwon et al. [2003]. The model leads to a detection rule based on the posterior probability of the sensor failure given the observed scores at a certain time instance without looking at the time series of measurements. A model-based outlier detection method is developed in Tulone and Madden [2006]. The method relies on estimating a regression model for each individual sensor, and estimating deviations from the predictions of the model. Jefferey et al. [2006] proposes a systematic database approach for data cleansing. A time window primitive for outlier detection based on model estimation is proposed.

4.2.2 Sequential detection

Sequential change point problems have been extensively analyzed for the case when there is a single change point (e.g. see [Lai, 2001]). Some popular procedures are the CUSUM procedure [Page, 1954] and the Shiryaev-Roberts-Pollak procedure [Roberts, 1966; Shiryayev, 1963], which rely on likelihood ratios. The asymptotic performance of these procedures when pre-change and post-change distributions are known have been analyzed in a series of papers, starting with Lorden [1971], under different performance criteria, and minimax and Bayesian settings [Pollak, 1987; Lai, 1998; Borovkov, 1999; Tartakovsky and Veeravalli, 2005].

Tartakovsky and Veeravalli [2005] show the asymptotic delay optimality of the Shiryaev-Polak-Roberts (SRP) rule under diminishing false alarm rates for the Bayesian setting. Borovkov [1999] compute the asymptotic delay of the CUMSUM rule under diminishing worst case false alarm probability, over all possible change times.

Single change point models have been proposed as fault detection procedures for a single system [Benveniste et al., 1990]. Such a model has not been extended for application in the sensor network type setting, where each element of a system can experience a change point, but changes have specific correlation structures.

In this Chapter we introduce a multiple change point model for fault detection that attempts to bring together some of the preferred empirical approaches, such as correlation tracking [Papadimitriou et al., 2006], with guaranteed performance bounds possible through change point analysis. The generalization of the a posteriori rule, a natural generalization of the SRP procedure, is shown to be far from optimal. Near optimality is shown for a specific algorithm, which is a specially constructed extension of the SRP rule for the fault detection problem.

In sensor network applications, it is desirable to have procedures that only use information of geographically close sensors to limit communication costs and improve lifetime for the network. Such a constraint leads to a distributed processing constraint in a change point problem. Various authors [Mei, 2008; Veeravalli, 2001; Veeravalli et al., 1993] analyze distributed versions of single change point problems, and derive an optimal rule for some cases, under the constraint that the rules belong to predetermined classes.

4.3 Problem statement

4.3.1 Set-up and underlying assumptions

There are m sensors, labeled u_1 to u_m. Sensor u's measurements form the time series $\{X_t(u)\}$. We are interested in developing an online and distributed procedure for detecting faulty sensors based on the data $\{X_t(u_i) \mid i = 1, \ldots, m\}$. Our method relies on the following assumptions, elaborated further below.

- Neighboring functioning sensors have correlated sensor measurements, but a failed sensor's measurements are not correlated with its functioning neighbors. The neighborhood relationship is specified by the known *fault graph* $\mathcal{G}(V, E)$: V is the set of sensors or nodes and E is the set of undirected edges (Figure 4.1). The graph normally includes self loops. In practice, the neighborhood relationship is that of geographic

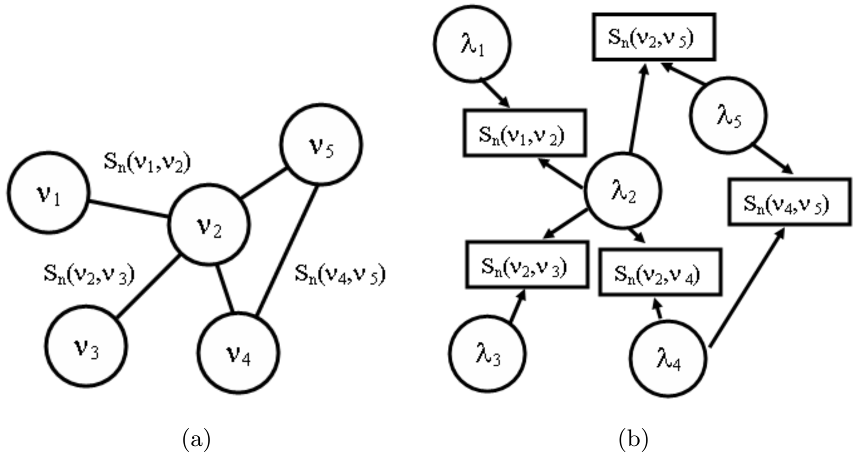


Figure 4.1. (a) Neighborhood graph of a sensor network and (b) corresponding statistical dependency graph.

proximity. In PeMS, for example, sensors at the same and adjacent locations on the freeway are considered neighbors.

- Each sensor makes a periodic noisy p-dimensional measurement of its environment. $X_t(u)$ is the measurement at time t. The sensors need not be synchronized.

- Instead of making a decision at each sampling time t, we choose to make decisions after a block of T samples has been observed. The time scale T is selected to be longer than that of an event. For instance, in PeMS, T corresponds to the number of samples for a day. We index blocks by k and n.

- Sensors fail independently and the time of failure λ_u has a prior distribution π_u. The time of failure is defined in the block time domain n.

- Each pair of sensors (u, u') at each time instant n computes a score $S_n(u, u')$ which reflects how correlated are the measurements at time n. The variables $S_n(u, u')$ are independent conditional on the failure times λ_u and $\lambda_{u'},$. Naturally, the scores are symmetric: $S_n(u, u') = S_n(u', u)$.

- The score $S_n(u, u')$ experiences a change when either sensor u or u' fails:

$$\begin{aligned}\mathbf{S}_n(u, u') &\overset{i.i.d}{\sim} f_0(\cdot|u, u'), \quad n < \min(\lambda_u, \lambda_{u'}), \\ &\overset{i.i.d.}{\sim} f_1(\cdot|u, u'), \quad n \geq \min(\lambda_u, \lambda_{u'}).\end{aligned}$$

  Once a change happens, the score cannot experience any further change.

- At time n, each sensor can use the information it observed until the current time.

The setup captures the notion that after a sensor has failed, any scores related to it cannot be used anymore since its own measurements become uncorrelated with those of its

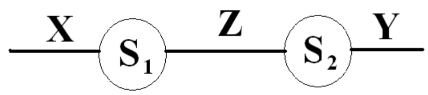


Figure 4.2: Transformation of the data of two sensors.

.

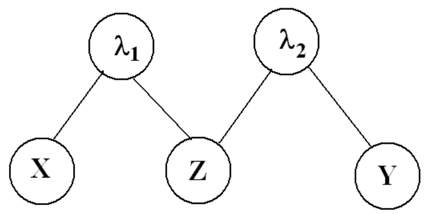


Figure 4.3. Graphical representation of the dependency structure between random variables $X, Y, Z, \lambda_1, \lambda_2$.

neighbors, i.e. neighbors cannot use this particular shared score to infer their own state. In this problem there are m change points that have to be detected. Each sensor u runs its own sequential test represented by a stopping time ν_u. Based on the information available at time n, the rule sets $\nu_u = n$ if it decides that u has failed at time n. The stopping time can only depend on the the scores $S(u, u')$ for u' in neighborhood of u in the fault graph. We enforce this constraint to represent the limited computation and communication setting available in typical distributed sensor networks.

In this Chapter, we propose a strategy to determine the stopping time ν_u for a general matching graph. The interactions between the various stopping times are complex. To obtain analytic performance insights we restrict to the case where $m = 2$. The reduced problem captures most of the important features of the complete problem and is amenable to theoretical analysis. The reduced setup is described next.

4.3.2 Reduced setup and notation

We consider the sequential detection problem shown in Figure 4.2. We have reduced the full sensor network analysis to the analysis of two sensors, 1 and 2. The random variable relating to the common link between sensors is denominated Z. The random variable relating to the non common link of sensor 1 is denoted as X. Similarly the random variable relating to the non common link of sensor 2 is denominated Y. The sequence of random variables in X is denoted as $\mathbf{X}_n^k$, and similarly for other random variables. The change time (or failure time) variables for sensors 1 and 2 are denoted by λ_1 and λ_2. We assume change times for different sensors are independent and identically distributed: The joint prior distribution of the change times is denoted $\mathbb{P}(\lambda_1 = k_1, \lambda_2 = k_2) = \pi(k_1)\pi(k_2)$. The independence assumption is realistic for the applications we consider. Define the cumulative quantities $\Pi_n^1 = \mathbb{P}(\lambda_1 > n)$ and $\Pi_n^2 = \mathbb{P}(\lambda_2 > n)$.

Figure 4.3 shows the probability dependency structure. Conditional on the change times λ_1 and λ_2, the random variables X, Y and Z are all independent. Furthermore, the density function prior to change is given by $f_0(\cdot)$. The density function after change is given by

$f_1(\cdot)$. In an effort to simplify notation, the density functions are different for each random variable, but this is implied with the use of the argument. The product density of random variables $X_r, ...X_n$ with density $f_0(X_i)$ for $i = r, ..., k-1$, and density $f_1(X_i)$ for $i = k, ..., n$ will be denoted as $\mathrm{L}_k(\mathbf{X}_n^r)$ We make similar definitions for random variables Z and Y.

The σ-field generated by a sequence such as $\mathbf{X}_n^1$ is denoted by $\mathcal{F}_X^n$. For the fields of joint variables, we use notation such as $\mathcal{F}_{X,Y}^n$. The probability measure in the joint space of random variables when the change happens at $\lambda_1 = k_1$ and $\lambda_2 = k_2$ is defined as:

$$
\begin{aligned}
\mathbb{P}_{k1,k2} &= \mathbb{P}_{k_1}(X)\mathbb{P}_{k_1\wedge k_2}(Z)\mathbb{P}_{k_2}(Y) \\
&\sim \prod_{i=1}^{k_1-1} f_0(X_i) \prod_{i=k_1}^{\infty} f_1(X_i) \prod_{i=1}^{k_1\wedge k_2-1} f_0(Z_i) \prod_{i=k_1\wedge k_2}^{\infty} f_1(Z_i) \prod_{i=1}^{k_2-1} f_0(Y_i) \prod_{i=k_2}^{\infty} f_1(Y_i) \\
&= \mathrm{L}_{k_1}(\mathbf{X}_\infty^1)\mathrm{L}_{k_1\wedge k_2}(\mathbf{Z}_\infty^1)\mathrm{L}_{k_2}(\mathbf{Y}_\infty^1).
\end{aligned}
$$

The measure has the property that the random variable Z changes as soon as a change happens in either sensors. This models the expected behavior in a fault detection problem: the correlation corresponding to the common link becomes zero as soon as either sensor 1 or 2 fails. This feature of the measure makes fault detection a challenging problem, since it is a multiple change point, multiple hypothesis test. From the definitions it can also see that in the special case $\lambda_2 = \infty$ we have $\mathbb{P}_{k_1,\infty} = \mathbb{P}_{k_1}(X)\mathbb{P}_{k_1}(Z)\mathbb{P}_\infty(Y)$. The appropriate marginalized measures are also defined, such as:

$$
\mathbb{P}_{\lambda_1,\lambda_2} = \sum_{k_1=1}^{\infty}\sum_{k_2=1}^{\infty} \pi(k_1)\pi(k_2)\mathbb{P}_{k1,k2}.
$$

The restriction that sensor 1 can only use random variables X and Z for its decision, whereas sensor 2 can only use random variables Y and Z for its decision localized, can be given in terms of the preceding definitions.

Definition 4.3.1 (Localized stopping time)**.** A localized stopping time *for sensor 1 is a stopping time* $\nu_1 \in \mathcal{F}_{X,Z}^n$. *Similarly, a localized stopping time for sensor 2 is a stopping time* $\nu_2 \in \mathcal{F}_{Y,Z}^n$.

For expectations our notation is that $\mathbb{E}_{k_1,k_2}$ refers to expectations with respect to the measure $\mathbb{P}_{k_1,k_2}$. It will be useful to define the log-likelihood ratio of sample i for random variable X_i and the accumulated log-likelihood:

$$
r_i(X) = \log\left(\frac{f_1(X_i)}{f_0(X_i)}\right) \;\; ; \;\; R_n^k(X) = \sum_{i=k}^{n} r_i(X). \tag{4.3.1}
$$

Similar definitions hold for all random variables. We make assumptions about the expectations of the log-likelihoods under pre-change and post-change distributions. In particular, assume they are all finite ($*$ denotes *don't care*):

$$\begin{aligned}\mathbb{E}_{1,*}[r_i(X)] &= \int f_1(x)\log\frac{f_1(x)}{f_0(x)}\mu(dx)\\ &= D(f_1(X)||f_0(X))\\ &= q_1(X),\end{aligned} \tag{4.3.2}$$

where μ is the Lebesgue measure. Similarly,

$$\begin{aligned}\mathbb{E}_{\infty,*}[r_i(X)] &= \int f_0(x)\log\frac{f_0(x)}{f_1(x)}\mu(dx)\\ &= -D(f_0(X)||f_1(X))\\ &= -q_0(X).\end{aligned} \tag{4.3.3}$$

For Y a similar assumption holds, only noting that expectations will be with respect to $\mathbb{E}_{*,0}$ and $\mathbb{E}_{*,\infty}$. For Z, again the definitions hold, but expectations should be with respect to $\mathbb{E}_{0,0}$ and $\mathbb{E}_{\infty,\infty}$. The assumption is that $q_0(X), q_1(X), q_0(Z), q_1(Z), q_0(Y)$ and $q_1(Y)$ are all positive and finite.

Further detailed technical assumptions are stated in Section 4.10.

4.3.3 Performance metrics

A fault detection rule for sensor u is denoted ν_u, and this is a stopping time [Durrett, 1995]. In the change point literature, such a stopping time is evaluated according to two metrics: probability of false alarm and detection delay, see e.g., [Tartakovsky and Veeravalli, 2005].

Definition 4.3.2 (Probability of false alarm)**.** *Given a stopping time ν_u and the change time λ_u the false alarm probability at $\lambda_u = k_u$ is defined as*

$$PFA^{(k_1,k_2)}(\nu_u) = \mathbb{P}_{k_1,k_2}(\nu_u \le k_u). \tag{4.3.4}$$

The false alarm probability for procedure ν_u is given by

$$\begin{aligned}PFA^{\pi_1,\pi_2}(\nu_u) &= \mathbb{P}_\lambda(\nu_u \le \lambda_u)\\ &= \sum_{k_1=1}^{\infty}\sum_{k_2=1}^{\infty}\pi_1(k_1)\pi_2(k_2)\mathbb{P}_{k_1,k_2}(\nu_u \le k_u).\end{aligned}$$

The marginal false alarm probabilities for procedures ν_1 and ν_2 are

$$\begin{aligned}MPFA^{(\lambda_1,k_2)}(\nu_1) &= \mathbb{P}_{\lambda_1,k_2}(\nu_1 \le \lambda_1),\\ MPFA^{(k_1,\lambda_2)}(\nu_2) &= \mathbb{P}_{k_1,\lambda_2}(\nu_2 \le \lambda_2).\end{aligned} \tag{4.3.5}$$

Definition 4.3.3 (Detection delay)**.** *The m-th moment of the delay of a sequential procedure*

ν_u for change time $\lambda_u = k_u$ is defined as

$$D_m^{(k_1,k_2)}(\nu_u) = \mathbb{E}_{k_1,k_2}\left[(\nu_u - k_u)^m \,|\nu_u \geq k_u\right]. \tag{4.3.6}$$

The m-th moment of the detection delay is

$$\begin{aligned} D_m^{\pi_1,\pi_2}(\nu_u) &= \mathbb{E}_{\lambda_1,\lambda_2}\left[(\nu_u - \lambda_u)^m \,|\nu_u \geq \lambda_u\right] \\ &= \sum_{k_1=1}^{\infty}\sum_{k_2=1}^{\infty} \pi_1(k_1)\pi_2(k_2) D_m^{(k_1,k_2)}(\nu_u). \end{aligned} \tag{4.3.7}$$

A good procedure achieves small (even minimum) delay $D_m^{\pi}(\nu_u)$, while maintaining $\mathrm{PFA}^{\pi}(\nu_u) \leq \alpha$, for a pre-specified PFA α.

An optimal detection procedure for sensor u is a procedure ν_u for which the delay $D_1^{\pi_1,\pi_2}(\nu_u)$ is minimized while keeping the false alarm below a chosen probability α so that $\mathrm{PFA}^{\pi_1,\pi_2}(\nu_u) < \alpha$. Such a rule is called an *optimal sequential procedure*. Notice that the *optimal sequential procedure* does not necessarily satisfy $\mathrm{MPFA}(\nu_u) < \alpha$, since only the average false alarm is guaranteed to be below α.

4.3.4 Data Preprocessing and Fault Behavior Model

Denote by $\mathbf{X}_n(u)$ the nth observed sample block by sensor u, which has size $T \times p$. Let $\mathcal{H}_{u,n}$ denote data available up to block $n-1$. Each sensor computes a vector score at time n, determined by a transformation F:

$$\begin{aligned} &\mathbf{S}_n(u,u) = F(\mathbf{X}_n(u), \mathcal{H}_{u,n}), \\ &\mathbf{S}_n(u,u') = F(\mathbf{X}_n(u), \mathbf{X}_n(u'), \mathcal{H}_{u,n}, \mathcal{H}_{u',n}), u' \in \mathcal{N}_u. \end{aligned} \tag{4.3.8}$$

$\mathcal{N}_u$ is the set of neighbors u' of u. We call $\mathbf{S}_n(u,u')$ the *link score* of the link $(u',u) \in E$. The transformation is symmetric, so $\mathbf{S}_n(u,u') = \mathbf{S}_n(u',u)$. The statistic F captures a notion of distance between two block samples. We focus on correlation statistics, defined in the next subsection. In time block units the random change time is λ_u, which when required we assume to be a geometric random variable with parameter $d_u T$.

Intuitively, our fault detection model posits that the score $\mathbf{S}_n(u,u')$ undergoes a change in distribution whenever either u or u' fails, i.e., at time $\min(\lambda_u, \lambda'_u)$. This model captures the notion that in a networked setting, failed sensor data cannot be used to detect faults in other sensors. Thus our model departs from the traditional single change point detection models [Lai, 2001], in that we are dealing with multiple *dependent* change points based on measurements from a collection of sensors. The standard theory for a single change point can no longer be applied in a straightforward manner.

We formally specify our change point model. Given a score function $\mathbf{S}_n(u,u')$ for each pair of neighbors (u,u'), it is assumed that $\mathbf{S}_n$ for different pairs of sensors are independent. Also given are distributions $f_0(\cdot|u,u')$ and $f_1(\cdot|u,u')$.

4.3.5 Correlation scores

Our choice of correlation score function is motivated by the observation that in many applications when a sensor fails the the correlation experiences an abrupt change (e.g. [Kwon et al., 2003]). The choice of correlation statistics is also attractive because it can be used in non-stationary environments if the time scale is appropriately chosen. Without losing generality assume $p = 1$ so that $X_t(u)$ is a scalar and $\mathbf{X}_n(u)$ is a vector of size T.

The score is defined as

$$\begin{aligned} \mu_n(u) &= \frac{1}{T} \sum_{t \in T_n} X_t(u), \qquad (4.3.9) \\ s_n(u, u') &= \frac{1}{T} \sum_{t \in T_n} (X_t(u) - \mu_n(u))(X_t(u') - \mu_n(u')), \\ \mathbf{S}_n(u, u') &= \phi \left(\frac{s_n(u, u')}{\sqrt{s_n(u, u) s_n(u', u')}} \right), \\ T_n &= [(n-1)T + 1, nT]. \end{aligned}$$

The actual score is a transformation of the empirical correlation estimate. The trivial choice is $\phi(x) = x$. To obtain desired statistical behavior, it is sometimes better to choose a combined Fisher and Box-Cox type transformation,

$$\phi_f(x, \gamma) = \frac{1}{2} \log \left(\frac{1 + x^\gamma}{1 - x^\gamma} \right). \qquad (4.3.10)$$

We assume that the scores scaled by $\sqrt{T}$ converge to a normal distribution, and are independent conditional on the change times λ_u and λ'_u . This assumption is not required and more complex covariance structures inferred from the data could be used. But our choice works well in practice, and simplifies exposition. Thus

$$\begin{aligned} \mathbf{S}_n(u, u') &\sim N(\mu(u, u'), T^{-1}\ \sigma^2_{u,u'}), \quad n < \frac{1}{T} \min(\lambda_u, \lambda'_u), \\ &\sim N(0, T^{-1}\ \sigma^2), \quad n \geq \frac{1}{T} \min(\lambda_u, \lambda'_u). \qquad (4.3.11) \end{aligned}$$

Before the change time, each computed score (in our case covariances) is approximately normal. The mean and variance parameters depend on the pairs of sensors. The variance scales as $1/T$ with respect to the window size T. Above we assumed mean and variance are time invariant, but this is not necessary. The assumption can be justified with a simple model. Suppose the blocks $\mathbf{X}_n(u)$ and $\mathbf{X}_n(u')$ are jointly Gaussian random variables, and the Fisher-Box transformation (Equation 4.3.10) with $\gamma = 1$ is used; it can then be shown [Lehmann, 1999] that asymptotic normality holds and

$$\sigma^2_{u,u'} = \begin{cases} \frac{(1-\mu_{u,u'})^2}{T}, & \text{for } \phi(x) = x \\ \frac{1}{T}, & \text{for } \phi(x) = \phi_f(x, 1) \end{cases} . \qquad (4.3.12)$$

The link information measure for (u, u') is [Lehmann, 1999]:

$$q_1(u,u') = D(f_1 \| f_0)$$
$$= T\,\frac{\mu(u,u')^2}{2\sigma^2_{u,u'}} + \frac{1}{2}\left[\frac{\sigma^2}{\sigma^2_{u,u'}} + \log\left(\frac{\sigma^2_{u,u'}}{\sigma^2}\right) - 1\right]. \tag{4.3.13}$$

The link information measure is minimized when $\sigma^2_{u,u'} = \sigma^2$.

4.4 Multiple Sensor Online Detection

In this section we investigate specific stopping time rules to solve the multiple sensor online fault detection problem proposed in Section 4.3. We start the section discussing the classical posterior rule for detecting single change points, and the relevant performance bounds. Then, two new methods are proposed: a direct extension of the Bayes posterior rule, which is a natural extension of rules for single change points and LFDIE, a procedure that uses local information exchange. The methodologies are described and analyzed in the context of the reduced problem setup. Furthermore, some fundamental performance limits for the multiple sensor online fault detection problem are computed. To conclude the section, we extend the procedure LFDIE for the general fault detection problem and discuss its implementation and formulate a conjecture about its performance.

4.4.1 Background

Suppose a single sensor fails at a random time λ, with distribution $\mathbb{P}(\lambda = n) = \pi_1(n)$. The observations X are an identical independently distributed random variable, with distribution f_0 before change and f_1 after change. The fault detection formulation for this single sensor is the standard single change point detection problem.

Shirayev [Shirayev, 1978] showed that an *optimal sequential procedure* is the procedure that tests the hypothesis $H_1 : \lambda \le n$ against $H_0 : \lambda > n$ at each n, using the observations $X_1, ..., X_n$. The Shirayev-Robert-Polak (SRP) sequential procedure is a threshold test on the posterior probability

$$p_n = \mathbb{P}(\lambda \le n \,|\mathcal{F}^n_X\,) \tag{4.4.1}$$

The SRP procedure uses the test quantity:

$$\Lambda_n(X) = \frac{p_n}{1-p_n} = \frac{\displaystyle\sum_{k=0}^{n} \pi_1(k) \prod_{r=1}^{k} f_0(X_r) \prod_{r=k+1}^{n} f_1(X_r)}{\displaystyle\sum_{k=n+1}^{\infty} \pi(k) \prod_{r=1}^{n} f_0(X_r)},$$
$$= \Lambda_0 + \Pi_n^{-1} \sum_{k=1}^{n} \pi_1(k) e^{R^k_n(X)}. \tag{4.4.2}$$

Write

$$B_\alpha = \frac{1-\alpha}{\alpha}, \tag{4.4.3}$$

the Shirayev-Roberts-Polak (SRP) optimal stopping time is given by

$$\nu_S(X) \quad = \quad \inf\left\{n : \Lambda_n(X) \geq B_\alpha\right\}. \tag{4.4.4}$$

Tartakovsky and Veeravalli [2005] showed the SRP procedure achieves the *optimal asymptotic delay* for the problem of minimizing the expected delay constrained to a false alarm probability α (i.e. $\text{PFA}^\pi(\nu_S) < \alpha$). Furthermore, the asymptotic moments of the delay as $\alpha \to 0$ are bounded as

$$\begin{aligned} \lim_{\alpha\to 0} D_m^{(k)}(\nu_S(X)) \quad &\doteq \quad \left(\frac{|\log\alpha|}{q_1(X)+d}\right)^m \text{ for } k \geq 1, \\ \lim_{\alpha\to 0} D_m^{\pi}(\nu_S(X)) \quad &\doteq \quad \left(\frac{|\log\alpha|}{q_1(X)+d}\right)^m, \end{aligned} \tag{4.4.5}$$

which *matches the lower bound for delays for any procedure with false alarm* α.

The constant $d > 0$ depends only on the prior distribution and is defined as

$$\lim_{k\to\infty} \frac{\Pi_{k+1}}{k} = -d \tag{4.4.6}$$

Distributions with exponential tail, such as the geometric distribution have $d > 0$ and finite, thus helping to reduce the delay. Distributions with heavy tail have $d = 0$, and knowledge of the prior does not reduce the delay in the asymptotic analysis.

The single change point problem is considerably simpler than the multiple change problem, since once a change is detected, it is attributed to a unique fault, and there is no chance of *confusion* with other potentially failed sensors.

Remark 1. *Whenever required, the generalization of the test for a SRP procedure using random variables X and Z is*

$$\Lambda_n(X,Z) = \Lambda_0 + \Pi_n^{-1} \sum_{k=1}^{n} \pi_1(k) e^{R_n^k(X)+R_n^k(Z)}. \tag{4.4.7}$$

The corresponding stopping time is $\nu_s(X,Z)$ and uses the threshold in Equation (4.4.3). Similarly we can define $\Lambda_n(Y,Z)$ and $\Lambda_n(Y)$, using π_2, Y and Z. The corresponding stopping times are $\nu_s(Y,Z)$ and $\nu_s(Y)$.

Remark 2. *We assume without loss of generality that $\Lambda_0 = 0$ for the SRP procedure.*

4.4.2 Localized stopping time without information exchange

In this Section we study the first natural approach to solving the multiple sensor failure sequential test problem. We focus on sensor 1. Heuristically, a threshold test in the posterior probability seems a reasonable choice for stopping time. For the single change point case this is an optimal choice. In the modified framework, such a choice may not be optimal, but it is certainly an attractive and simple test. Intuitively, this is the first test one would consider. The posterior probability test can be written as:

$$\begin{aligned} \nu_1(X,Z) &= \inf\left\{n : p_n(X,Z) \geq 1-\alpha\right\}, \\ p_n(X,Z) &= \mathbb{P}_{\lambda_1,\lambda_2}(\lambda_1 \leq n \mid \mathcal{F}^n_{X,Z}). \end{aligned} \tag{4.4.8}$$

To put into standard form, notice that

$$\frac{p_n(X,Z)}{1-p_n(X,Z)} \geq \frac{1-\alpha}{\alpha}, \tag{4.4.9}$$

is an equivalent test to the original. Then the test statistic is

$$\Lambda^{\text{noex}}_n(X,Z) = \frac{\mathbb{P}_{\lambda_1,\lambda_2}(\lambda_1 \leq n \mid \mathcal{F}^n_{X,Z})}{\mathbb{P}_{\lambda_1,\lambda_2}(\lambda_1 > n \mid \mathcal{F}^n_{X,Z})}. \tag{4.4.10}$$

From the problem definition, we can compute the probabilities involved in the statistics

$$\begin{aligned} \Lambda^{\text{noex}}_n(X,Z) &= \frac{a_n}{b_n}, \\ a_n &= \sum_{k_1=1}^{n}\sum_{k_2=1}^{\infty} \pi_1(k_1)\pi_2(k_2)\mathrm{L}_{k_1}(\mathbf{X}^1_n)\mathrm{L}_{k_1\wedge k_2}(\mathbf{Z}^1_n), \\ b_n &= \Pi_{1,n}\,\mathrm{L}_{n+1}(\mathbf{X}^1_n)\left\{\Pi_{2,n}\,\mathrm{L}_{n+1}(\mathbf{Z}^1_n) + \sum_{k_2=1}^{n}\pi_2(k_2)\mathrm{L}_{k_2}(\mathbf{Z}^1_n)\right\}. \end{aligned} \tag{4.4.11}$$

Similarly,we can define a stopping time based on $\Lambda^{\text{noex}}_n(Y,Z)$ for sensor 2. The first important observation is that computing the test quantity is hard in general. Moreover, no delay reduction benefit is obtained from using the shard link.

Theorem 4.4.1. *Assume $q_0(Z) \geq q_1(Z)$ or $\pi_2(k_2) > 0$ for $k_2 \geq K_2$. For the posterior threshold test $\nu_1(X,Z)$ without information exchange given by Equation (4.4.8), the delay satisfies*

$$\begin{aligned} \lim_{\alpha\to 0} D^{k_1,k_2}_1(\nu_1(X,Z)) &\geq \lim_{\alpha\to 0} D^{k_1,k_2}_1(\nu_1(X)), \\ \lim_{\alpha\to 0} D^{\pi_1,\pi_2}_1(\nu_1(X,Z)) &\geq \lim_{\alpha\to 0} D^{\pi_1,\pi_2}_1(\nu_1(X)). \end{aligned} \tag{4.4.12}$$

For the threshold test $\nu_2(Y,Z)$, the delay satisfies

$$\begin{aligned} \lim_{\alpha\to 0} D^{k_1,k_2}_2(\nu_2(Y,Z)) &\geq \lim_{\alpha\to 0} D^{k_1,k_2}_2(\nu_2(Y)), \\ \lim_{\alpha\to 0} D^{\pi_1,\pi_2}_2(\nu_2(Y,Z)) &\geq \lim_{\alpha\to 0} D^{\pi_1,\pi_2}_2(\nu_2(Y)). \end{aligned} \tag{4.4.13}$$

The result shows that the performance of the rule does not depend on the statistics of the shared link Z. This is a negative result, since we expect an improvement in performance of the order of the KL divergence for the pre and post-change distributions of Z.

The intuition behind this result lies in the fact that without some information from the private link of sensor 2, sensor 1 cannot be sure that the shared link failed because of a fault in sensor 1 or in sensor 2. Therefore, in the hypothesis test, the null hypothesis as well as the alternative hypothesis incorporate the information that a change in the shared link could have happened because the other sensor failed.

Thus in this extension the common link information is not useful in determining which sensor has failed. The reason is that the information in either link pair (X, Z) or (Y, Z) by itself is not helpful in determining whether the change in Z is induced by a failure in sensor 1 or in sensor 2.

4.4.3 LFDIE: A localized stopping time with information exchange

In this section we propose LFDIE (Localized Fault Detection with Information Exchange), an interacting stopping time method that attempts to overcome the limitations discussed in the previous section and benefit from common link information.

We now proceed to define the networked procedure we implement in the sensor network in Figure 4.3. First, denote by ν_1 the stopping time rule for sensor 1, given by using the information from random variables X and Z (see Equation (4.4.7)):

$$\begin{aligned} \nu_1 &= \min\{k : \Lambda_k(X, Z) \geq B_\alpha\} \\ &= \nu_S(X, Z). \end{aligned} \tag{4.4.14}$$

Next, denote by $\tilde{\nu}_1$ the stopping time rule for sensor 1 obtained by using just the random variable X:

$$\begin{aligned} \tilde{\nu}_1 &= \min\{k : \Lambda_k(X) \geq B_\alpha\}, \\ &= \nu_S(X). \end{aligned} \tag{4.4.15}$$

Define stopping times for sensor 2 analogously: $\nu_2 = \nu_S(Y, Z)$ and $\tilde{\nu}_2 = \nu_S(Y)$. We can now define LFDIE in terms of the above stopping times. The stopping time choice for sensor 1 is

$$\bar{\nu}_1 = \nu_1 \mathbb{I}(\nu_1 \leq \nu_2) + \max(\tilde{\nu}_1, \nu_2)\, \mathbb{I}(\nu_1 > \nu_2). \tag{4.4.16}$$

Similarly define for sensor 2:

$$\bar{\nu}_2 = \nu_2 \mathbb{I}(\nu_2 \leq \nu_1) + \max(\tilde{\nu}_2, \nu_1)\, \mathbb{I}(\nu_2 > \nu_1) \tag{4.4.17}$$

In LFDIE there is an information exchange between sensors, but it is constrained to a single bit that informs when a sensor's statistic has crossed its threshold. Then the other sensor stops using the common link (that is, it recomputes its own statistics without using the information in the shared link). This is a new feature of the model investigated in this chapter. Previous literature in distributed hypothesis testing focused in the case where all sensors observed the same hypothesis. Here we have a problem where sensors observe hypothesis that interact with each other. The maximum operator is required because when the hypothesis test is recalculated not using the common information, one has already waited at least ν_2 for sensor 1 or ν_1 for sensor 2.

The procedure works in an intuitive manner: Each sensor computes posteriors as if the other sensor is always working, until the time one of them declares itself as failed. Notice that both sensors at this point are using the information in the shared link. When one sensor is thought to have failed (e.g. $\nu_1 > \nu_2$) the other sensor stops using the shared link information, and recomputes the change point test using only the information of its own 'private' link. The max operator reflects the situation that information for one's own private link also dictates that its sensor hsa failed (e.g., $\tilde{\nu}_1 < \nu_2$), in which case one should stop immediately at the present time (ν_2).

Implementation of the procedure requires an extra single bit of information that is issued to neighbors when a sensor declares itself as failed. If this bit is received the neighboring sensors stop using the shared link with the failed sensor, and use a rule based on the remaining links.

LFDIE requires very little memory for the two sensor case, as each sensor doesn't need to store the observed values of the random variables X, Y, Z. One can just compute two recursions for each sensor using Equation (4.4.7). For the multiple sensor case, each sensor has to keep track of all the link variables since when the shared link is dropped the sensor has to recompute the score using only the remaining links, and there are multiple such combinations. In Section 4.5 we propose efficient solutions for this problem.

4.4.4 Performance Analysis: False Alarm

In the remainder of the section we compute the false alarm probability and the detection delay for LFDIE. The detection with information exchange algorithm is interesting if we are able to show that for a given false alarm rate $O(\alpha)$, it achieves expected delays smaller than if the common link information is not used. Detailed technical assumptions are stated in Section 4.10.

Intuitively, analyzing Equations (4.4.16) and (4.4.17) we notice that sensor 1 can raise two kinds of false alarm at some time n: one caused without any change ($\lambda_1 > n$ and $\lambda_2 > n$), and another caused when the shared link experiences a change due to a fault in sensor 2 ($\lambda_2 < n$). Based on this observation we define the confusion probability:

Definition 4.4.1. *The confusion probabilities of stopping times in a set of procedures* $(\bar{\nu}_1, \bar{\nu}_2)$ *are defined as*

$$\begin{aligned} \xi^{\alpha}_{\lambda_1,\lambda_2}(\bar{\nu}_1) &= \mathbb{P}_{\lambda_1,\lambda_2}(\bar{\nu}_1 \le \bar{\nu}_2, \lambda_2 \le \bar{\nu}_1 \le \lambda_1), \\ \xi^{\alpha}_{\lambda_1,\lambda_2}(\bar{\nu}_2) &= \mathbb{P}_{\lambda_1,\lambda_2}(\bar{\nu}_2 \le \bar{\nu}_1, \lambda_1 \le \bar{\nu}_2 \le \lambda_2). \end{aligned} \tag{4.4.18}$$

A ***regular*** *fault detection procedure is a set of procedures for which the following conditions hold:*

$$\begin{aligned} \lim_{\alpha \to 0} \xi^{\alpha}_{\lambda_1,\lambda_2}(\bar{\nu}_1) &= 0, \\ \lim_{\alpha \to 0} \xi^{\alpha}_{\lambda_1,\lambda_2}(\bar{\nu}_2) &= 0. \end{aligned} \tag{4.4.19}$$

*A **strong** fault detection procedure is a set of procedures that has*

$$\begin{aligned}\xi^{\alpha}_{\lambda_1,\lambda_2}(\bar{\nu}_1) &= O(\alpha),\\ \xi^{\alpha}_{\lambda_1,\lambda_2}(\bar{\nu}_2) &= O(\alpha).\end{aligned} \tag{4.4.20}$$

The definition of false alarm for sensor 1 is

$$\mathrm{PFA}^{\pi_1,\pi_2}(\bar{\nu}_1) = \mathbb{P}_{\lambda_1,\lambda_2}(\bar{\nu}_1 < \lambda_1). \tag{4.4.21}$$

By the choice of the threshold for ν_1 and $\tilde{\nu}_1$, the false alarm when there is no change observed in sensor 2 ($\lambda_2 = \infty$) is bounded:

$$\begin{aligned}\mathrm{PFA}^{\pi_1,\infty}(\nu_1) &= \mathbb{P}_{\lambda_1,\infty}(\nu_1 < \lambda_1) < \alpha,\\ \mathrm{PFA}^{\pi_1,\infty}(\tilde{\nu}_1) &= \mathbb{P}_{\lambda_1,\infty}(\tilde{\nu}_1 < \lambda_1) < \alpha.\end{aligned} \tag{4.4.22}$$

Based on this observation, together with the definition of confusion probability we can bound the false alarm probability of LFDIE as shown in Theorem 4.4.2. The proof for the theorem is shown in Section 4.11.

Theorem 4.4.2 (False alarm of LFDIE)**.**

(a) The probability of false alarm of sensors 1 and 2 for the joint procedure with information exchange is bounded by:

$$\begin{aligned}PFA^{\pi_1,\pi_2}(\bar{\nu}_1) &< 2\,\alpha + \xi^{\alpha}_{\lambda_1,\lambda_2}(\bar{\nu}_1),\\ PFA^{\pi_1,\pi_2}(\bar{\nu}_2) &< 2\,\alpha + \xi^{\alpha}_{\lambda_1,\lambda_2}(\bar{\nu}_2)\end{aligned} \tag{4.4.23}$$

(b) The marginal probability of false alarm of sensors 1 and 2 for LFDIE *is bounded by:*

$$\begin{aligned}MPFA^{\pi_1,k_2}(\bar{\nu}_1) &< 2\,\alpha + \xi^{\alpha}_{\lambda_1,k_2}(\bar{\nu}_1),\\ MPFA^{k_1,\pi_2}(\bar{\nu}_2) &< 2\,\alpha + \xi^{\alpha}_{k_1,\lambda_2}(\bar{\nu}_2)\end{aligned} \tag{4.4.24}$$

To complete the understanding about the false alarm, the confusion probability needs to be analyzed. The results up to now show that this probability is a key quantity to model the interaction between stopping times in the fault detection problem. In general, we believe that quantities of this type will emerge whenever a multiple change point problem is solved in a distributed manner.

Theorem 4.4.3 (Confusion probability regularity)**.** *The theorem is stated for sensor 1. For sensor 2 it suffices to exchange the role of X and Y.*

(a) *The procedure* LFDIE *is a regular fault detection procedure.*

(b) *Let assumptions 4.10.2 and 4.10.5 hold. Define $b_1 = q_0(X) - q_1(Z) + d_1$ and the rate*

$$r^*_a = \frac{1}{w^*}\frac{[\min\{q_0(X), q_1(Z)\} + q_1(Y) + d_1 - d_2]^2}{\max\{\sigma^2_0(X), \sigma^2_1(Z)\} + \sigma^2_1(Y)}, \tag{4.4.25}$$

where

$$w^* = \sqrt{\frac{\sigma_1^2(X,Z)}{\max\{\sigma_0^2(X), \sigma_1^2(Z)\} + \sigma_1^2(Y)}}[\min\{q_0(X), q_1(Z)\} + q_1(Y) + d_1 - d_2] - b_1.$$

Then

$$\lim_{\alpha \to 0} \frac{\log \xi^{\alpha}_{\lambda_1,\lambda_2}(\bar{\nu}_1)}{\log \alpha} \leq r^*,$$

where

(a) If $b_1 \leq 0$ *then* $r^* = r_a^*$*;*

(b) If $b_1 > 0$ *then* $r^* = \max(r_a^*, r_b^*)$*, where*

$$r_b^* = 4\,\frac{b_1}{\sigma_1^2(X,Z)}.$$

Therefore, if $r^* > 1$*,* LFDIE *is a strong fault detection procedure.*

Proof.

$$\begin{aligned}
\xi^{\alpha}_{\lambda_1,\lambda_2}(\bar{\nu}_1) &= \sum_{k_1,k_2} \pi(k_1)\pi(k_2)\mathbb{P}_{k_1,k_2}(\bar{\nu}_1 \leq \bar{\nu}_2, k_2 \leq \bar{\nu}_1 < k_1) \\
&= \sum_{k_1,k_2} \pi(k_1)\pi(k_2)\mathbb{P}_{k_1,k_2}(\nu_1 \leq \nu_2, k_2 \leq \nu_1 < k_1) \\
&= \sum_{k_1,k_2} \pi(k_1)\pi(k_2)\mathbb{P}_{\infty,k_2}(\nu_1 \leq \nu_2, k_2 \leq \nu_1 \leq k_1) \\
&\leq \sum_{k_1,k_2} \pi(k_1)\pi(k_2)\mathbb{P}_{\infty,k_2}(k_2 \leq \nu_1 \leq \nu_2) \\
&= \sum_{k_2} \pi(k_2)\mathbb{P}_{\infty,k_2}(k_2 \leq \nu_1 \leq \nu_2).
\end{aligned}$$

We continue the proof using

Lemma 4.4.1. *Let Assumption 4.10.2. For any* $k_2 > 0$*, the following bound holds:*

$$\lim_{\alpha \to 0} \frac{\log \mathbb{P}_{\infty,k_2}(k_2 \leq \nu_1 \leq \nu_2)}{\log \alpha} \leq r^*,$$

Let Assumption 4.10.2 and 4.10.5. Then:

$$\lim_{\alpha \to 0} \frac{\log \mathbb{P}_{\infty,\lambda_2}(k_2 \leq \nu_1 \leq \nu_2)}{\log \alpha} \leq r^*,$$

Given this lemma, it is immediate by the dominated convergence theorem that as $\alpha \to 0$:

$$\xi^{\alpha}_{\lambda_1,\lambda_2}(\bar{\nu}_1) \to 0.$$

showing that the procedure is regular proving **(a)** without Assumption 4.10.5 . Including Assumption 4.10.5, **(b)** follows since $\mathbb{P}_{\infty,\lambda_2}(k_2 \leq \nu_1 \leq \nu_2) = \sum_{k_2} \pi(k_2)\mathbb{P}_{\infty,k_2}(k_2 \leq \nu_1 \leq \nu_2)$. A similar proof can be shown for sensor 2. $\square$

For sensor 1, the amount of information of link Y only comes into play if the information available on its own private link is not strong enough compared to the shared link, meaning $q_0(X) > q_1(Z)$. Otherwise, for the procedure to be regular, it is required that $q_0(X) \approx q_1(Z)$ and sensor 2 needs to have high information on its own private link $q_1(Y)$, so the conditions are satisfied.

4.4.5 Performance Analysis: Detection Delay

For the analysis of the delay asymptotics we follow the approach proposed in Tartakovsky and Veeravalli [2005] with careful modifications to account for the differences in structure in our problem. We also simplify the exposition somewhat.

Definition 4.4.2 (Detection Delays)**.** *Define the following detection delay constants:*

$$\begin{aligned} L_1^\alpha &= \frac{|\log\alpha|}{q_1(X)+q_1(Z)+d_1}, L_2^\alpha = \frac{|\log\alpha|}{q_1(Y)+q_1(Z)+d_2}, \\ \tilde{L}_1^\alpha &= \frac{|\log\alpha|}{q_1(X)+d_1}, \tilde{L}_2^\alpha = \frac{|\log\alpha|}{q_1(Y)+d_2}, \end{aligned}$$

where d_1 is the rate for prior π_1 and d_2 is the rate for prior π_2.

We start with a fundamental lemma, which is basically a change of measure argument. For the lemma we need an assumption on the log likelihood ratio (LLR) partial sums defined in Equation (4.3.1). The assumption 4.10.3 just implies a standard weak uniform convergence. We also need a careful definition of a class of procedures.

Define the conditional false alarm,

$$\mathrm{MPFA}^{\pi_1,\pi_2}(\bar{\nu}_1|\lambda_2<\lambda_1) = \sum_{k_1=1}^{\infty}\pi_1(k_1)\sum_{k_2=1}^{k_1-1}\pi_2(k_2)\mathbb{P}_{\infty,k_2}(\bar{\nu}_1<k_1). \tag{4.4.26}$$

Definition 4.4.3 (False alarm classes)**.** *For stopping times $\nu_1(X,Z)$ dependent only on X and Z define the classes:*

(i) $\Delta_1(\alpha)$ *such that* $PFA^{\pi_1,\infty}(\nu_1) \leq \alpha$*,*

(ii) $\tilde{\Delta}_1(\alpha,k_2)$ *such that* $MPFA^{\pi_1,k_2}(\nu_1) \leq \alpha$*,*

(iii) $\tilde{\Delta}_1(\alpha)$ *such that* $MPFA^{\pi_1,\pi_2}(\nu_1|\lambda_2<\lambda_1) \leq \alpha$*.*

Also, define similar classes for stopping times $\nu_2(Y,Z)$ dependent on Y and Z.

Notice that the procedures $\bar{\nu}_1$ and $\bar{\nu}_2$ does not belong to $\Delta_1(\alpha)$ or to $\Delta_2(\alpha)$, as they have a false alarm rate that is greater than α and more importantly they depend on all three X, Z and Y by definition. But ν_1 and $\tilde{\nu}_1$ do belong to $\Delta_1(\alpha)$ and ν_2 and $\tilde{\nu}_2$ belong to $\Delta_2(\alpha)$.

We can now proceed to prove the fundamental Lemma. The arguments are basically a change of measure, followed by using our concentration assumption for the log-likelyhood ratio and exponential tail assumption for the prior.

Lemma 4.4.2. *Define for all* $0 < \epsilon < 1$*:*

$$\begin{aligned}
\gamma_{\epsilon,\alpha}^{(k_1,k_2)}(\nu_i) &= \mathbb{P}_{k_1,k_2}(k_i \leq \nu_i \leq k_i + (1-\epsilon)L_i^{\alpha}),\\
\gamma_{\epsilon,\alpha}(\nu_i) &= \mathbb{P}_{\lambda_1,\lambda_2}(\lambda_i \leq \nu_i \leq \lambda_i + (1-\epsilon)L_i^{\alpha}),\\
\gamma_{\epsilon,\alpha}(\nu_i, \lambda_j < \lambda_i) &= \mathbb{P}_{\lambda_1,\lambda_2}(\lambda_i \leq \nu_i \leq \lambda_i + (1-\epsilon)L_i^{\alpha}, \lambda_j < \lambda_i),\\
\tilde{\gamma}_{\epsilon,\alpha}^{(k_1,k_2)}(\nu_i) &= \mathbb{P}_{k_1,k_2}(k_i \leq \nu_i \leq k_i + (1-\epsilon)\tilde{L}_i^{\alpha}),\\
\tilde{\gamma}_{\epsilon,\alpha}(\nu_i) &= \mathbb{P}_{\lambda_1,\lambda_2}(\lambda_i \leq \nu_i \leq \lambda_i + (1-\epsilon)\tilde{L}_i^{\alpha}, \lambda_j < \lambda_i).
\end{aligned}$$

where d_1 *depends only on the prior* π_1 *and* $j = 2$ *if* $i = 1$ *and* $j = 1$ *if* $i = 2$*. Then for all* $k_1, k_2 \geq 1$ *and* $0 < \epsilon < 1$*:*

$$\begin{array}{ll}
(i) & \lim_{\alpha\to 0}\sup_{\nu_1\in\Delta_1(\alpha)} \gamma_{\epsilon,\alpha}^{(k_1,k_2)}(\nu_1) = 0,\\
& \lim_{\alpha\to 0}\sup_{\nu_1\in\Delta_1(\alpha)} \gamma_{\epsilon,\alpha}(\nu_1) = 0,\\
& \lim_{\alpha\to 0}\sup_{\nu_1\in\Delta_1(\alpha)} \gamma_{\epsilon,\alpha}(\nu_1, \lambda_1 < \lambda_2) = 0,\\
(ii) & \lim_{\alpha\to 0}\sup_{\nu_1\in\tilde{\Delta}_1(\alpha,k_2)} \tilde{\gamma}_{\epsilon,\alpha}^{(k_1,k_2)}(\nu_1) = 0 \text{ for } k_1 > k_2,\\
& \lim_{\alpha\to 0}\sup_{\nu_1\in\tilde{\Delta}_1(\alpha,k_2)} \gamma_{\epsilon,\alpha}^{(k_1,k_2)}(\nu_1) = 0 \text{ for } k_1 \leq k_2,\\
(iii) & \lim_{\alpha\to 0}\sup_{\nu_1\in\tilde{\Delta}_1(\alpha)} \tilde{\gamma}_{\epsilon,\alpha}(\nu_1) = 0.
\end{array}$$

An equivalent result holds for ν_2 *belonging to classes* $\Delta_2(\alpha)$*,* $\tilde{\Delta}_2(\alpha, k_2)$ *and* $\tilde{\Delta}_2(\alpha)$*.*

Based on Lemma 4.4.2 we can prove a performance lower bound for the two sensor case among certain classes of procedures. The lower bound essentially guarantees that no procedure that belongs in the given class can have delay smaller than that stated in the bound. It gives us a certificate against which to check the optimality of a given procedure.

Theorem 4.4.4 (Delay lower bound)**.** *Let Assumption 4.10.3 and denote* $c_d = (1 + o(1))$*. Then:*

$$\lim_{\alpha\to 0}\inf_{\nu_1\in\tilde{\Delta}_1(\alpha,k_2)} \mathbb{E}_{k_1,k_2}[(\nu_1 - k_1)^m|\nu_1 \geq k_1] \geq \left[(L_1^{\alpha})^m\mathbb{I}(k_1 \leq k_2) + (\tilde{L}_1^{\alpha})^m\mathbb{I}(k_1 > k_2)\right] c_d,$$

$$\lim_{\alpha\to 0}\inf_{\nu_1\in\Delta_1(\alpha)\cap\tilde{\Delta}_1(\alpha)} \mathbb{E}_{\lambda_1,\lambda_2}[(\nu_1 - \lambda_1)^m|\nu_1 \geq \lambda_1] \geq \left[(L_1^{\alpha})^m\mathbb{P}(\lambda_1 \leq \lambda_2) + (\tilde{L}_1^{\alpha})^m\mathbb{P}(\lambda_1 > \lambda_2)\right] c_d,$$

$$\lim_{\alpha\to 0}\inf_{\nu_2\in\tilde{\Delta}_2(\alpha,k_1)} \mathbb{E}_{k_1,k_2}[(\nu_2 - k_2)^m|\nu_1 \geq k_2] \geq \left[(\tilde{L}_2^{\alpha})^m\mathbb{I}(k_1 \leq k_2) + (L_2^{\alpha})^m\mathbb{I}(k_1 > k_2)\right] c_d,$$

$$\lim_{\alpha\to 0}\inf_{\nu_2\in\Delta_2(\alpha)\cap\tilde{\Delta}_2(\alpha)} \mathbb{E}_{\lambda_1,\lambda_2}[(\nu_2 - \lambda_2)^m|\nu_2 \geq \lambda_2] \geq \left[(\tilde{L}_2^{\alpha})^m\mathbb{P}(\lambda_1 \leq \lambda_2) + (L_2^{\alpha})^m\mathbb{P}(\lambda_1 > \lambda_2)\right] c_d.$$

Proof. We prove the first two assertions. First notice that from the definitions in Lemma 4.4.3, $\tilde{\Delta}_1(\alpha, k_2) \subseteq \Delta_1(\alpha)$. Also notice that $\tilde{\Delta}_1(\alpha, k_2) \subseteq \tilde{\Delta}_1(\alpha)$, so that $\tilde{\Delta}_1(\alpha, k_2) \subseteq \Delta_1(\alpha)\cap$

$\tilde{\Delta}_1(\alpha)$. Let $\nu_1 \in \tilde{\Delta}_1(\alpha, k_2)$, if $k_1 \le k_2$:

$$\begin{aligned}\mathbb{E}_{k_1,k_2}[(\nu_1 - k_1)^m | \nu_1 \ge k_1] &= \frac{\mathbb{E}_{k_1,k_2}[(\nu_1 - k_1)_+^m]}{\mathbb{P}_{k_1,k_2}(\nu_1 \ge k_1)}, \\ &\ge \frac{((1-\epsilon)L_\alpha^1)^m}{\mathbb{P}_{k_1,k_2}(\nu_1 \ge k_1)}(\mathbb{P}_{k_1,k_2}(\nu_1 \ge k_1) - \gamma_{k_1,k_2}(\nu_1)),\end{aligned}$$

But $\mathbb{P}_{k_1,k_2}(\nu_1 \ge k_1) = 1 - \mathbb{P}_{\infty,\infty}(\nu_1 < k_1) \ge 1 - \alpha/\Pi_{k_1}^1$ for $k_1 \le k_2$ using Lemma 4.11.3, and Lemma 4.4.2 shows that $\gamma_{k_1,k_2}(\nu_1)) \to 0$ uniformly over ν_1, so

$$\inf_{\nu_1 \in \tilde{\Delta}_1(\alpha,k_2)} \mathbb{E}_{k_1,k_2}[(\nu_1 - k_1)^m | \nu_1 \ge k_1] \ge ((1-\epsilon)L_\alpha^1)^m(1 + o(1)) \quad \text{as } \alpha \to 0.$$

A similar bound works for $k_2 < k_1$, except $\mathbb{P}_{k_1,k_2}(\nu_1 \ge k_1) = \mathbb{P}_{\infty,k_2}(\nu_1 \ge k_1) \ge 1 - \alpha/\Pi_n^1$ for $\nu_1 \in \tilde{\Delta}_1(\alpha, k_2)$.

For the second statement, we note that:

$$\begin{aligned}\inf_{\nu_1 \in \tilde{\Delta}_1(\alpha)} \mathbb{E}_{\lambda_1,\lambda_2}[(\nu_1 - \lambda_1)_+^m] \ge &\inf_{\nu_1 \in \tilde{\Delta}_1(\alpha)} \mathbb{E}_{\lambda_1,\lambda_2}[(\nu_1 - \lambda_1)_+^m \mathbb{I}(\lambda_1 \le \lambda_2)] + \\ &+ \inf_{\nu_1 \in \tilde{\Delta}_1(\alpha)} \mathbb{E}_{\lambda_1,\lambda_2}[(\nu_1 - \lambda_1)_+^m \mathbb{I}(\lambda_1 > \lambda_2)]\end{aligned}$$

We can use Lemma 4.4.2 (i) and (iii) to bound such quantities in the same manner as in the first case. Lemma 4.11.3 (i) and (iii) can be used to bound the appropriate probabilities as before. □

We conclude the section computing the asymptotic delay of the procedure LFDIE. The asymptotic performance differs from the lower bound only on the factor δ_α.

Theorem 4.4.5 (Performance of LFDIE)**.** *Let Assumption 4.10.4.The delay of* LFDIE *represented as the set of stopping times* $(\bar{\nu}_1, \bar{\nu}_2)$ *has:*

(a) *For* $\alpha \to 0$,

$$\begin{aligned}D_1^{\pi_1,\pi_2}(\bar{\nu}_1) &\le D_1^{\pi_1,\infty}(\nu_1)\frac{1 - \delta_\alpha + o(1)}{1 - o(1)} + D_1^{\pi_1,\infty}(\tilde{\nu}_1)\frac{\delta_\alpha + o(1)}{1 - o(1)}, \\ D_1^{\pi_1,\pi_2}(\bar{\nu}_2) &\le D_1^{\pi_1,\infty}(\nu_2)\frac{\delta_\alpha + o(1)}{1 - o(1)} + D_1^{\pi_1,\infty}(\tilde{\nu}_2)\frac{1 - \delta_\alpha + o(1)}{1 - o(1)}\end{aligned} \tag{4.4.27}$$

(b) *For* $\alpha \to 0$,

$$\begin{aligned}D_1^{\pi_1,\pi_2}(\bar{\nu}_1) &\ge D_1^{\pi_1,\infty}(\nu_1)\frac{1 - \delta_\alpha + o(1)}{1 - o(1)} + D_1^{\pi_1,\infty}(\tilde{\nu}_1)\frac{\delta_\alpha + o(1)}{1 - o(1)}, \\ D_1^{\pi_1,\pi_2}(\bar{\nu}_2) &\ge D_1^{\pi_1,\infty}(\nu_2)\frac{\delta_\alpha + o(1)}{1 - o(1)} + D_1^{\pi_1,\infty}(\tilde{\nu}_2)\frac{1 - \delta_\alpha + o(1)}{1 - o(1)}.\end{aligned} \tag{4.4.28}$$

Here:

$$D_1^{\pi_1,\infty}(\nu_1) = \frac{|\log\alpha|}{q_1(X)+q_1(Z)+d_1},\ D_1^{\pi_1,\infty}(\nu_2) = \frac{|\log\alpha|}{q_1(Y)+q_1(Z)+d_2},$$
$$D_1^{\pi_1,\infty}(\tilde{\nu}_1) = \frac{|\log\alpha|}{q_1(X)+d_1},\ D_1^{\pi_1,\infty}(\tilde{\nu}_2) = \frac{|\log\alpha|}{q_1(Y)+d_1},$$
$$\delta_\alpha = \mathbb{P}_{\lambda_1,\lambda_2}(\nu_1 > \nu_2). \tag{4.4.29}$$

The results are also valid for λ_1 and λ_2 replaced by k_1 and k_2.

4.4.6 General Networks

The shared information algorithm for the two-sensor network can be suitably modified for a general network. Table 4.1 shows the proposed procedure, following the same principle as the two-sensor case. In this algorithm, whenever a sensor declares itself failed, all its neighbors recompute their test statistic excluding links with the failed sensor. Section 4.5 discusses implementation details, including finite storage, and transmission efficient computation.

The analysis in Sections 4.4.4 and 4.4.5 applies to the general network if the probability of sensors failing simultaneously is small, which will be the case if the fault rates are very small compared to the number of neighboring sensors. The analysis even with this simplification is quite involved, but a key quantity emerges—the confusion probability. If the confusion probability is small, the probability of false alarm is small.

The asymptotic delays depend crucially on the parameter δ_α. In this subsection we explore this further, for the case of independent identically distributed link distributions in a fully connected network.

In the two-sensor case, if X and Y have the same probability density, it is clear from symmetry that $\delta_\alpha = 1/2$. Focusing on sensor 1, we see that the delay in this case is

$$D_m^\pi(\bar{\nu}_1) \doteq \frac{1}{2}\, D_m^\pi(\nu_1) + \frac{1}{2}\, D_m^\pi(\tilde{\nu}_1).$$

Furthermore, it is known that if we have $\lambda_2 = \infty$ fixed (sensor 2 never fails), then any detection procedure ν has a delay that satisfies [Tartakovsky and Veeravalli, 2005]

$$D_m^\pi(\nu) \geq \left[\frac{|\log\alpha|}{q_1(X)+q_1(Z)+d}\right]^m = D_m^\pi(\nu_1).$$

In the case when $\lambda_2 = 0$ fixed (sensor 2 is always failed), link Z gives no information about the status of sensor 1, so any procedure for detecting a fault in sensor 1 satisfies

$$D_m^\pi(\nu) \geq \left[\frac{|\log\alpha|}{q_1(X)+d}\right]^m = D_m^\pi(\tilde{\nu}_1).$$

Networked Sensor Fault Detection: Each sensor $u \in V$ initializes its current neighbors set with all neighboring sensors in the fault graph (including self loops), so $\mathcal{N}_W(u) = \mathcal{N}(u)$. Then each sensor updates its current estimate of its own change point test statistic at time n:

(a) *Data Dissemination:* Each sensor broadcasts its current block of T samples $\mathbf{X}_n(u)$ to sensors u' that are active neighbors in the fault graph (i.e. $u' \in \mathcal{N}_W(u)$). Transmitted block might be transformed or compressed (see Section 4.5).

(b) *Score Computation:* After collecting all data blocks, the sensor computes the current score for shared links according to some transformation F, for example the correlation (Equation (4.3.9)):

$$S_n(u,u') = F(\mathbf{X}_n(u), \mathbf{X}_n(u')), \qquad u' \in \mathcal{N}_W(u). \tag{4.4.30}$$

(c) *Update Test Statistic:* Recursive update of test statistic using active links(Section 4.5):

$$\begin{aligned} O_n(u) &= \sum_{u' \in \mathcal{N}_W(u)} \left\{ \frac{(S_n(u,u'))^2}{2\sigma^2} + \right. \\ &\qquad \left. - \frac{(S_n(u,u') - \mu_{uu'})^2}{2\sigma^2_{uu'}} + \log\left(\frac{\sigma^2}{\sigma^2_{uu'}}\right) \right\} \\ \log(\Lambda_n(u)) &= \log\left(\frac{\Lambda_{n-1}(u)}{1-\rho} + \frac{\rho}{1-\rho}\right) + O_n(u), \\ \Lambda_0(u) &= \pi_0/(1-\pi_0), \;\; \rho = 1 - e^{-dT} \end{aligned} \tag{4.4.31}$$

(d) *Fault check and inform:* If

$$\Lambda_n(u) \geq \frac{1-\alpha}{\alpha}, \tag{4.4.32}$$

sensor u is declared faulty, and broadcasts failed bit $\delta(u)$ to all sensors $u' \in \mathcal{N}_W(u)$.

(e) *Update Current Links*: For each $u' \in \mathcal{N}_W(u)$, if bit $\delta(u')$ is received:

$$\begin{aligned} &\mathcal{N}_W(u) = \mathcal{N}_W(u) - u', \\ &\text{Recompute } \Lambda_n(u) \text{ with new } \mathcal{N}_W(u), \text{using stored samples.} \end{aligned} \tag{4.4.33}$$

If $\mathcal{N}_W(u)$ is empty (no self loops in fault graph), then stop sensor u.

Table 4.1. Description of the networked fault detection algorithm. In a centralized data collection model, the data dissemination stage has no cost.

For any procedure

$$D_m^\pi(\nu) = D_m^\pi(\nu|\lambda_1 < \lambda_2)\mathbb{P}(\lambda_1 < \lambda_2) + \\ + D_m^\pi(\nu|\lambda_1 \geq \lambda_2)\mathbb{P}(\lambda_1 \geq \lambda_2).$$

Since the priors are identical, $\mathbb{P}(\lambda_1 \geq \lambda_2) = 1/2$. The statistics of ν conditional on $\lambda_1 < \lambda_2$ are the same as when we set $\lambda_2 = \infty$. This result can be understood intuitively since Z indicates the failure of sensor 1 in this case. So $D_m^\pi(\nu|\lambda_1 < \lambda_2) \geq D_m^\pi(\nu_1)$. Intuitively, when $\lambda_1 \geq \lambda_2$ link Z gives no information on the change point for sensor 1, so any procedure should only use link X in the limit of small false alarm probability. Heuristically we reason that $D_m^\pi(\nu|\lambda_1 \geq \lambda_2) \geq D_m^\pi(\tilde{\nu}_1)$. Putting it all together gives

$$D_m^\pi(\nu) \geq \frac{1}{2}\, D_m^\pi(\nu_1) + \frac{1}{2}\, D_m^\pi(\tilde{\nu}_1).$$

Thus, in a sense the proposed procedure achieves optimality, if the confusion probability is of $O(\alpha)$.

Consider now a fully connected network, with all links having i.i.d. link distributions before and after change. Denote the performance metric by q_1. Notice that everything is symmetric in this case. Each sensor has an equal chance of being the $(n-k)$th sensor to fail. If we take small false alarm probability ($\alpha \to 0$) and all pairwise confusion probabilities go to zero with the false alarm probability going to zero, it is clear that no false alarm occurs. In the limit, the kth sensor uses either $(k-1)$ sensors to make its decision (if there are no self loops in the graph) or k (if there are self loops). The delay is

$$D_m^\pi(\nu_k) \doteq \left[\frac{|\log(\alpha)|}{(k-1+\delta_s)q_1 + d}\right]^m, \tag{4.4.34}$$

where $\delta_s = 1$ if the fault graph has self loops. Since each sensor has an equal chance of failing as the k-th sensor, the average delay for each sensor is

$$D_m^\pi(\nu) \doteq \frac{1}{|V|}\sum_{k=1}^{|V|}\left[\frac{|\log(\alpha)|}{(k-1+\delta_s)q_1 + d}\right]^m. \tag{4.4.35}$$

4.5 Algorithm Implementation

We investigate several practical considerations in the implementation of the proposed detection algorithm.

4.5.1 Correlation Computation: Compression and Synchronization

Given blocks $\mathbf{X}_n(u)$ and $\mathbf{X}_n(u')$ from sensors u and u', direct correlation as in Equation 4.3.9 might not be the best choice, either because the clocks of the two sensors may be delayed relative to each other, or more importantly, there could be a propagation delay in the underlying physical environment that reduces the effective correlation score between both sensors. A simple solution to improve performance and overcome these difficulties is to use cross correlation instead of correlation [Oppenheim et al., 1999b]. Denote by $\mathbf{X}_n^k(u)$ the block of samples $X_t(u)$ for $t \in [(n-1)T + k, nT + k]$, that is the samples delayed by k units.

The maximum cross correlation can be used to 'synchronize' the samples:

$$[k^{opt}, l^{opt}] = \arg \max_{k,l \in [0,M], k \leq l} \frac{1}{P} \sum_{n \in [1,P]} F(X_n^k(u), X_n^l(u')).$$

Here M is the maximum allowed shift between the sensor samples, P is the number of blocks to evaluate the shift, and F is the correlation score definition in Equation (4.3.9). The shift is adjusted so that the correlation between samples is maximized either once at initialization or periodically depending on the clock skew between the nodes. Once the shift is adjusted, correlations are computed with respect to the chosen shifts.

If the block size T is large enough, an alternative procedure, which saves energy by reducing the amount of data transfer, is to use a Discrete Cosine Transform (DCT) to evaluate the maximum cross correlation. The method relies on computing the DCT of each block $\mathbf{X}_n(u)$ appropriately zero-padded and using these coefficients to compute the maximal correlations with a simple scalar product. Additional savings can be obtained by using only a few coefficients of the DCT. Details of such a strategy can be found in [Oppenheim et al., 1999b]. If the underlying signal has a few dominant frequencies this method is very efficient. Alternative transforms such as wavelets could be used. In fact, this is the suggested approach even when synchronization is not required.

4.5.2 Quantization

Considerable savings can be obtained if the block vectors $\mathbf{X}_n(u)$ are quantized to some finite precision before the correlation is performed. Since we are working in a stochastic framework, dithered quantization is favored. A stylized version of quantizing a real number x in dithered quantization is to output $y = Q_b(x + \epsilon)$, where ϵ is a uniform random variable and Q_b is a function that outputs a b-bit quantized version of the input.

Denote by $\mathbf{S}_n^b(u, u')$ the correlation score computed from the quantized samples of block $\mathbf{X}_n(u)$. The following lemma gives the asymptotic behavior of the estimates, when the expected value of the score without quantization is $\mu_{u,u'}$.

Lemma 4.5.1. *Let us assume that the quantizer is $B+1$-bit with full scale X_{max} such that the quantization error is uniformly distributed in interval $[-\frac{X_{max}}{2^b}, \frac{X_{max}}{2^b}]$ and statistically independent of the system input. (This assumption is valid for subtractive dither quantization when the dither satisfies certain conditions, e.g. i.i.d uniform dither [Oppenheim et al., 1999b]). As $T \to \infty$,*

$$\sqrt{T}(\mathbf{S}_n^b(u, u') - \mu_{u,u'}) \xrightarrow{d} N(0, T\,\bar{\sigma}_{u,u'}^2)$$
$$\bar{\sigma}_{u,u'}^2 = \frac{1}{T}\left(\sigma_{u',u}^2 + 2\,X_{max}^2\,\sigma_b^2 + \sigma_b^4\right);\ \sigma_b^2 = \frac{1}{12 \cdot 2^{2b}}.$$

Proof. Once we replace x_i^b and y_i^b by $x_i + \epsilon_{x,i}^b$ and $y_i + \epsilon_{y,i}^b$ respectively, where $\epsilon_{x,i}^b$ and $\epsilon_{y,i}^b$ are the quantization errors for x_i^b and y_i^b respectively, x_i, $\epsilon_{x,i}^b$, y_i and $\epsilon_{y,i}^b$ are all independent of each other, and the result follows. $\square$

Quantization increases the variance of a Gaussian distribution by additional terms that are inversely proportional to 2^{2b}, so $b = O(-\log(\sigma_{u',u}/X_{\max}))$ gives a performance that is about the same with or without quantization.

4.5.3 Windowed iteration

Computational efficiency is important in practical applications. The information sharing procedure proposed in Section 4.4.3 relies on computing the Shiryaev statistic for each sensor (Equation (4.4.7)). The statistic can be recursively computed as:

$$\begin{aligned}\log(\Lambda_n) &= \log\left(\frac{\Pi_{n-1}}{\Pi_n}\Lambda_{n-1} + \frac{\pi_n}{\Pi_n}\right) + \log\left(\frac{f_1(S_n)}{f_0(S_n)}\right),\\ &= \log\left(\frac{\Lambda_{n-1}}{1-\rho} + \frac{\rho}{1-\rho}\right) + \log\left(\frac{f_1(X_n)}{f_0(X_n)}\right),\end{aligned} \tag{4.5.1}$$

where $\rho = \frac{1}{dT}$, and for correlation computation

$$\log\left(\frac{f_1(X_n)}{f_0(X_n)}\right) = \log\left(\frac{\sigma^2}{\sigma^2_{uu'}}\right) + \frac{S_n^2}{2\sigma^2} - \frac{(S_n-\mu_{uu'})^2}{2\sigma^2_{uu'}}. \tag{4.5.2}$$

The log function is used for convenience and to increase numerical precision.

The procedure in Section 4.4.3 requires each sensor to keep a history of all observed link score samples, since whenever a sensor detects a failure, others sensors that share links with the failed sensor have to recompute the test statistic without the shared link score. There is a practical implementation of the algorithm that avoids this. Before a failure occurs, the test statistic is ideally expected to be zero. After the failure, the proposed procedure requires about $D_m^\pi(\nu)$ samples to detect a fault, so a procedure that remembers a constant multiple of this number of samples works well. Notice that as sensors fail sequentially we have to increase the number of stored samples. Denoting by $\mathcal{N}_W(u)$ the set of working neighbors at time n, the sample storage size $M_n(u)$ required at time n for u is

$$\begin{aligned}\tilde{q}_{1,n}(u) &= \max_{u'\in\mathcal{N}_W(u)} q_1(u,u'),\\ M_n(u) &= T\,\frac{C\,\log(\alpha)}{\displaystyle\sum_{u'\in\mathcal{N}_W(u)} q_1(u,u') - \tilde{q}_1(u) + d\,T},\end{aligned} \tag{4.5.3}$$

in which C is a constant factor (a good choice is $C = 1.5$) and T is the window size. The memory estimate subtracts the most informative link at each stage since we don't know which sensor might fail requiring recomputation, and we always assume the most useful sensor (in terms of decreasing delay) might. Each time a sensor reports a failure, sensors that share fault links all recompute the Shiryaev statistic using the stored samples.

4.6 Time Scale Selection

We address the choice of time scale or block size T. We first show how performance for different T values can be compared. We then discuss how to choose T. Lastly, we show a practical problem using PeMS data.

4.6.1 Delay scaling

The fault model of Equation (4.3.11) might suggest that we could reduce detection delay arbitrarily, since by increasing T we can make the variance arbitrarily small. But to legitimately compare the mth moment of the delay for different T, we should consider the total number of samples rather than the number of blocks,

$$\begin{aligned} D_m^{\pi,T}(\nu_u) &= T^m \times D_m^{\pi}(\nu_u), \\ &= \left[T\,\frac{\log(\alpha)}{q_1 + d\,T}\right]^m = \left[\frac{\log(\alpha)}{\frac{q_1}{T} + d}\right]^m. \end{aligned}$$

Here q_1 is a sum or average of the individual link quality metric, which by Equation (4.3.13) is given by

$$\frac{q_1(u,u')}{T} = \frac{\mu(u,u')^2}{2\sigma_{u,u'}^2} + \frac{1}{2\,T}\left[\frac{\sigma^2}{\sigma_{u,u'}^2} + \log\left(\frac{\sigma_{u,u'}^2}{\sigma^2}\right) - 1\right].$$

Thus merely by increasing T one cannot reduce the delay arbitrarily: If the variances are equal before and after a fault, the delay (in number of samples) is independent of T; and if the variances are different, there could even be a performance loss as $\frac{q_1(u,u')}{T}$ might decrease with T.

4.6.2 Events and faults time scale comparison

The choice of the time scale parameter must compare the time scale of faults–duration between successive faults–and the time scale of events–time between signification changes in the environment. In most sensing environments, one expect events to have a much smaller time scale than faults. That is, a change in sensor measurements caused by an event is expected to propagate to neighboring sensors at a speed that depends on the physical environment. On the other hand, sensor faults should not propagate to neighboring sensors and these faults are likely to persist longer.

Sensor failures frequently are intermittent: a sensor fails and after some time it spontaneously recovers. (PeMS sensors suffer from intermittent failures.) In such situations, if a large enough density of sensors is available, the detection delay can be made small enough to detect intermittent failures. In fact, once a sensor is detected as failed, the sequential procedure can continue with some modifications to detect when the measurements are reliable again. So the requirement for detection of intermittent failures is that the average length of time a sensor remains failed is of the same order as the detection delay.

Consider a simple model in which once an event occurs at the location of sensor u, its

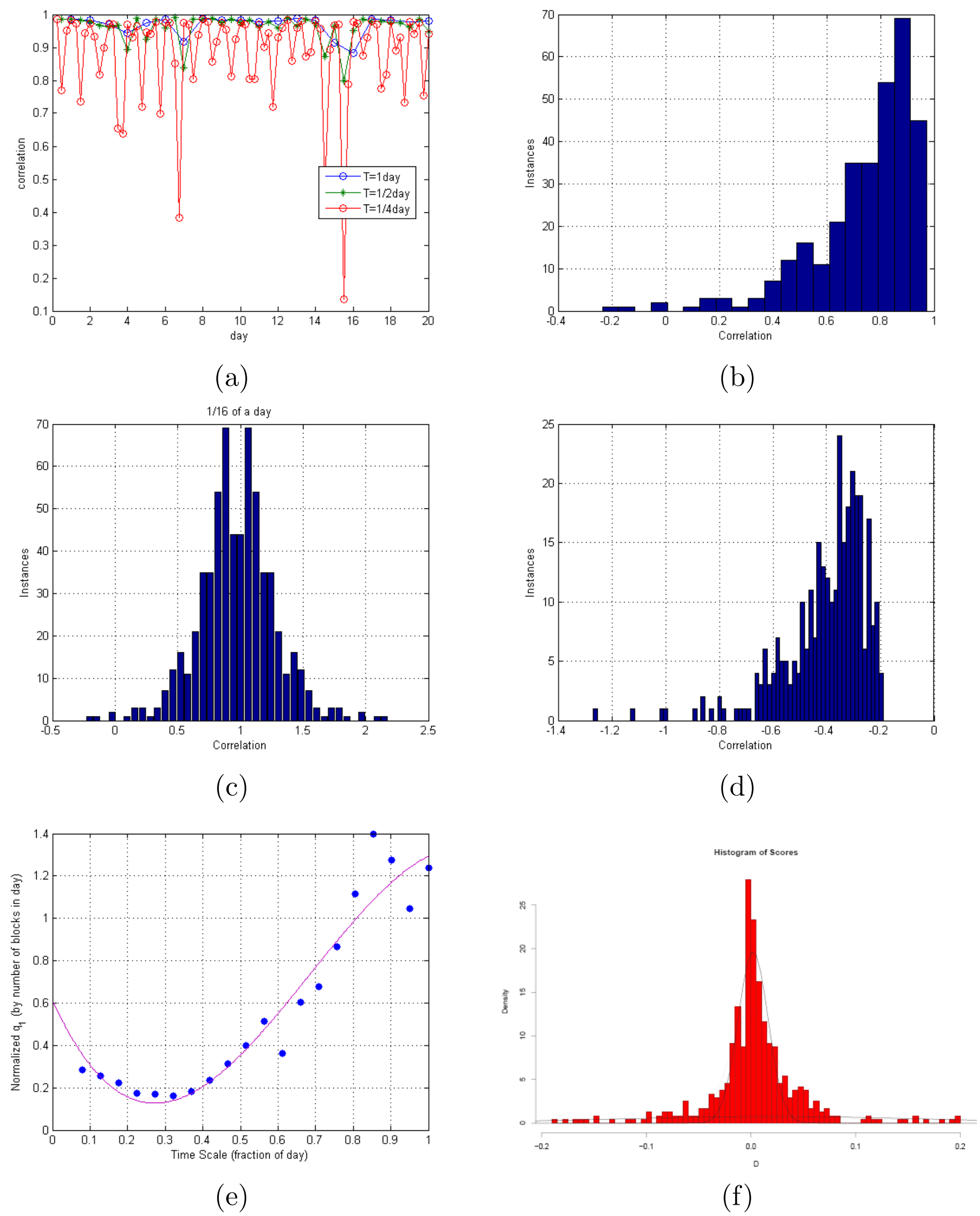


Figure 4.4. (a) Daily correlation values for different time scales, (b) Correlation distribution for 1/16 of total daily samples, (c) Symmetrized version of (b), (d) Fisher transform with $\gamma = 1$, (e) Information parameter q_1 normalized by T and (f) Correlation distribution for broken sensors from Kwon et al. [2003].

measurements become uncorrelated with those of its neighbor u'. Suppose events on average last τ samples. This could be either how long the event lasts, or the time to propagate the change caused by an event to neighboring sensors. During the time window τ, u samples an i.i.d. random variable with variance σ_e^2. At other times, the sensors sample i.i.d. values with a correlation of $\rho_{u,u'}$ and a variance of σ_S^2. If $\tau >> T$, we are unable to distinguish the event at sensor u from the sensor's failure. In fact, a simple computation reveals that the expectation of the empirical correlation with a window of size T (assuming an event at u occurs at the beginning of the time block) is

$$\hat{\rho}_{u,u'}(T,\tau) = \left\{ \frac{(1-r)_+}{\sqrt{(1-r)_+ + r\,\psi_{e,S}}} \right\} \rho_{u,u'},$$
$$r = \frac{\tau}{T},\ \psi_{e,S} = \frac{\sigma_e^2}{\sigma_S^2}$$

As expected, when T is large relative to τ, the effect of the event is reduced (implying a correlation that is close to the case when the event is not present). Furthermore, if event uncertainties are large with respect to usual behavior uncertainties, a larger time scale helps even more. If event uncertainties are small, expected correlations are smaller, but the events do not significantly affect the system.

4.6.3 Example

To show how to select the time scale in a real application, we use 5-minute average density data from PeMS for Interstate 210-West in Los Angeles, which has 45 sensing stations. Events such as accidents and demand-induced traffic congestion cause changes in the measured density, and we wish to distinguish the changes due to these events from changes due to sensor failures. We select two neighboring stations. Figure 4.4(a) shows the correlation over time for different time scales. Notice that for small time scales, we can observe large correlation drops, which correspond to events that have a low propagation speed. The implicit averaging proposed by our algorithm is essential in such situations.

Notice from Figure 4.4(b) that the correlation with the identity transformation function does not have a gaussian characteristic. The main reason for this is that our data set is limited. We propose two different approaches for handling such situations. Both are simple and fit within the methodology proposed here. The first approach uses a padded density estimate by adding a sample $2-r_i$ for each original sample r_i in the set. Figure 4.4(c) shows the padded histogram for our sample set, in which we can clearly see a bell curve. From this curve we are able to estimate the parameters $\mu = 1$ (by definition) and $\sigma^2 = 0.0928$. But we also know that correlation values never exceed 1 (which is also the mean of our estimated distribution). Thus, we should use as a distribution for the score the distribution conditional on the fact that the score is less than the mean, which can be directly computed as

$$\mathbf{S}_n(u,u') \sim 2\,N(1, T^{-1}\ \sigma_{u,u'}^2),\quad n < \frac{1}{T}\min(\lambda_u, \lambda_u').$$

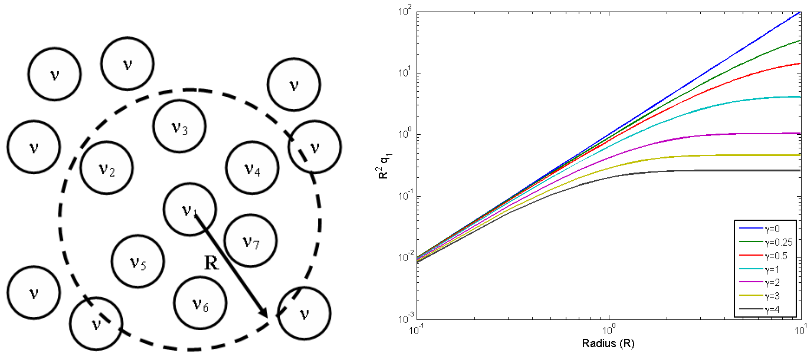


Figure 4.5: Informativeness models with respect to connectivity radius R .

After failure we don't see the cutoff effect [Kwon et al., 2003], so the distribution remains as before (Equation 4.3.11). Notice that the algorithm is identical, except that the constant factor $(-|\mathcal{N}_W(u)|\log 2)$ should be added to the definition of $O_n(u)$ in Table 4.1. The second approach is to use the Fisher type transformation in Equation (4.3.10). Figure 4.4(d) shows the result for the parameter value $\gamma = 1$. The distribution is more gaussian shaped. Figure 4.4(e) computes the scaled information metric $\bar{q}_1/T$ for several choices of the time scale parameter T. Observe that if the time is less than half a day, performance is the same. Some gains are observed as we increase the time scale.

4.7 Energy, delay and density tradeoff

We develop a tradeoff model to evaluate optimal choices of neighborhood size on an energy constrained network. We use delay results from previous sections to evaluate choices faced by a sensor under such constraints in a random placement setting.

4.7.1 Correlation decay

Many sensor networks monitor spatial and temporal changes in the environment. The correlation between measurements at different locations usually decays with distance. For example, in PeMS, the correlation of traffic measurements by adjacent sensors decays with the distance between them, since there are more points (ramps) where vehicles enter and exit. A simple way to capture this effect is an additive model

$$F(k+1) = F(k) + F_{in}(k+1) - F_{out}(k),$$

where k denotes the kth section of the highway, $F(k)$ denotes the flow in the kth section, $F_{in}(k+1)$ denotes the incoming flow to the kth section through an on-ramp, $F_{out}(k)$ denotes the outgoing flow in the previous section. Assume that the incoming flows are i.i.d. random variables with variance σ^2. If the outgoing flows are proportional to the input flows

($F_{out}(k) = -\beta F(k)$, for $0 < \beta < 1$) we have

$$\rho(k, \tilde{k}) = \frac{\sigma^2}{1-\beta^2}\beta^{|k-\tilde{k}|}.$$

The correlation decays with the distance between sensors, but the decay rates are different. The performance of the proposed fault detection algorithms depends crucially on the expected correlations between the sensors, as well as on the variance of this estimate, through the information parameter $q_1(u_i, u_j)$ of the link between sensors u_i and u_j. Under reasonable conditions, the variance of the correlation estimate increases as the correlation itself decreases. Under our normality assumptions, we showed that $q_1 = \rho^2/\sigma_\rho^2$. If we assume a power law decay with distance and $\sigma_\rho^2 = O(1/\rho^{2p})$, we can state that

$$q_1(u_i, u_j) \propto T\, \beta^{\gamma \cdot \operatorname{dist}(u_i, u_j)}, \tag{4.7.1}$$

in which the parameter $\gamma \geq 0$ controls the decay rate of the link informativeness as the distance between the sensors $\operatorname{dist}(u_i, u_j)$ increases.

4.7.2 Energy consumption

Some sensor networks have limited energy. If most energy is consumed in communication, it is important to minimize the data to be transferred. Suppose the energy consumed in transferring data between u_i and u_j is proportional to the square of the distance between them, $e_C(u_i, u_j) \propto \operatorname{dist}(u_i, u_j)^2$. There might then be a maximum radius R of interest to realize fault detection for a single sensor with a limited power budget.

4.7.3 Tradeoff analysis

We adopt the viewpoint of a single sensor u_1, whose neighbors are randomly placed following a Poisson process on a disk with center u_1 and mean (spatial) density η_F sensors per m^2 [Proakis, 2000]. Assume these neighbors never fail. We use a mean field approximation to evaluate the tradeoffs between energy, detection delay and density.

The expected link informativeness in a disk with radius R, normalized by time scale, is

$$\begin{aligned}
\bar{q}_1 &= \mathbb{E}[q_1(u_1, u_j)/T] \\
&= C \int_0^R \beta^{\gamma \operatorname{dist}(u_1, u_j)} d\mu(\operatorname{dist}(u_1, u_j)) \\
&= C \int_0^R \beta^{\gamma x} \frac{2}{R^2} x dx \\
&= \begin{cases} C & \gamma = 0 \\ \frac{2C}{(\log(\beta)\gamma R)^2}\left[1 + \beta^{\gamma R}(\log(\beta)\gamma R - 1)\right] & \gamma > 0 \end{cases}.
\end{aligned}$$

For density η_F, the disk has on average $N = \eta_F \pi R^2$ sensors. Using the mean field approximation (valid for large N), the expected sample delay of the detection procedure is

$$\mathbb{E}[D_m^\pi(\nu_1)] = \mathbb{E}\left[\frac{\log\alpha}{\sum_j q_1(u_1,u_j) + d\,T}\right] \approx \mathbb{E}\left[\frac{\log\alpha}{N\bar{q}_1\,T + d\,T}\right]$$
$$= \frac{\log\alpha}{\eta_F\,\pi R^2 \bar{q}_1\,T + d\,T}.$$

The expected power consumption for each transmission round to each neighbor is

$$\mathbb{E}[e_C(u_1,u_j)] = K\int_0^R \mathrm{dist}(u_1,u_j)^2 \mu(\mathrm{dist}(u_1,u_j)),$$
$$= K\int_0^R x^2 \frac{2}{R^2} x dx = \frac{1}{2} K\,R^2.$$

The average number of rounds of communication is $\bar{\lambda} + D_m^\pi(\nu_1)$, where $\bar{\lambda} = e^{dT}$ is the average failure time. Putting these together, using the mean field approximation to the delay in the first step, we obtain the total power consumed

$$\bar{P} = \mathbb{E}[e_C(u_1,u_j)(\lambda + D_m^\pi(\nu_1))\,N],$$
$$\approx \frac{1}{2} K\,R^2\,[\bar{\lambda} + \mathbb{E}[D_m^\pi(\nu_1)]]\,\eta_F\,\pi R^2.$$

If $\bar{q}_1$ is small compared to d, the expected delay is dominated by $1/d$, which is smaller than $\bar{\lambda}$. If $\bar{q}_1$ is large, the delay is small. Thus essentially the total average power consumed by sensor u_1 is $O(\rho\,R^4)$. The expected sample delay is of order

$$\mathbb{E}[D_m^{\pi,T}(\nu_1)] = T\,\mathbb{E}[D_m^\pi(\nu_1)] = O\left(\frac{1}{\max\{\eta_F\,R^2\,\bar{q}_1, d\}}\right).$$

There are two ways to improve performance: (1) by increasing R for a fixed density, which corresponds to communicating with neighbors further away, and (2) by increasing the density as a function of R, requiring additional sensors. Which choice is better depends on the parameter γ of the underlying environment. For the model in Equation (4.7.1), $R^2\bar{q}_1$ increases with R^2 when $\gamma = 0$, and is order constant when $\gamma > 0$. Thus increasing R for a fixed density does not help reduce the delay arbitrarily when $\gamma > 0$. Figure 4.5 plots $\bar{q}_1$ as a function of R for the different models. In the order constant situations we need to increase the density as a function of $\eta_F(R) = R^p$ for some $p > 0$, which increases energy consumption from $O(R^4)$ to $O(R^{4+p})$. If performance is measured as total average power per unit detection delay, $\bar{P}/\mathbb{E}[D_m^\pi(\nu_1)] = O(R^2/\bar{q}_1)$, increasing density improves performance.

4.8 Examples

We evaluate the performance of our algorithm in simulations, which allows us to precisely specify the moment of failure. We simulate three different situations: the two-node network and the fully connected network proposed in Section 4.4, and a toroidal grid network (see

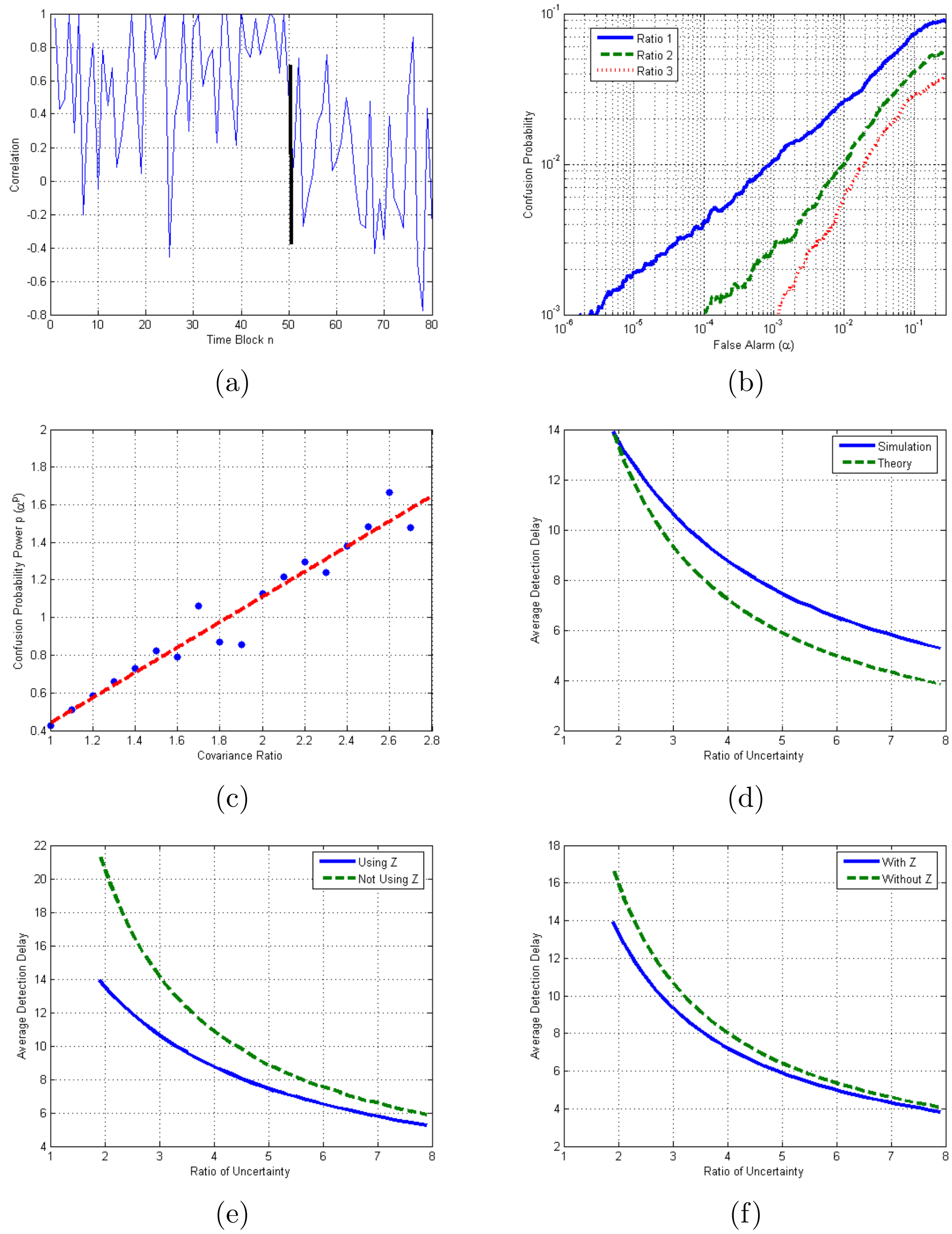


Figure 4.6. Two Sensor Network: (a)Sample path for correlation with change point at $n = 50$, (b) Confusion probability estimates for different variance ratios and (c) Confusion probability exponent estimates. Covariance ratio in these figures refers to the quantity σ_Z^2/σ_S^2.

[Dimakis et al., 2006] for a definition). This is basically a four connected network that wraps around.

As a benchmark, we compute the expected delay of a naive fault detection strategy: direct thresholding of the correlation, assuming that the distributions are known. For a 5-node fully connected network, and a false alarm probability of 0.0001, approximate computations reveal that the expected delay is on the order of 172 blocks. By comparison, our approach yields a delay of 50 blocks for a false alarm probability of 10^{-20} (essentially zero), which it is much more efficient. The main reason is that we perform appropriate implicit averaging.

4.8.1 Two Sensor Network

We consider the example in Figure 4.2., assuming that $f_0(X) \sim \mathcal{N}(\mu(X), \sigma^2(X))$ and $f_1(X) \sim \mathcal{N}(0, \sigma^2(X))$. Similarly, we make definitions for Y and Z. In this case, the information strength for each link are given by

$$q_1(X) = q_0(X) = \frac{\mu(X)^2}{\sigma^2(X)},$$
$$q_1(Y) = q_0(Y) = \frac{\mu(Y)^2}{\sigma^2(Y)},$$
$$q_1(Z) = q_0(Z) = \frac{\mu(Z)^2}{\sigma^2(Z)}.$$

Using the results obtained in Section 4.4.4, we can conclude that LFDIE is a strong fault detection procedure if

$$4\,\frac{q_0(X) - q_1(Z)}{2\sigma^2(X,Z)} > 1,$$
$$4\,\frac{q_0(Y) - q_1(Z)}{2\sigma^2(Y,Z)} > 1,$$

whenever $q_1(X) > q_1(Z)$ and $q_1(Y) > q_1(Z)$. Let us assume $\mu(X) = \mu(Y) = \mu(Z) = 1$, which is the standard type of assumption in correlation fault detection. $\sigma^2(X,Z)$ is the variance of the log-likelihood under after change measure for Z and pre-change measure for X, which can be computed as

$$\sigma^2(X,Z) = \frac{1}{\sigma^2(X)} + \frac{1}{\sigma^2(Z)},$$

obtaining the conditions

$$\sigma^2(X) < \frac{1}{3}\sigma^2(Z),$$
$$\sigma^2(Y) < \frac{1}{3}\sigma^2(Z).$$

The result can be interpreted intuitively if we consider that private links X and Y represent the aggregate connection of sensors 1 and 2 to always working sensors. Furthermore, consider that a connection between pairs of sensors has strength $1/\sigma^2$ (i.e. all variances involved are σ^2). Then, we can write $q_1(Z) = 1/\sigma^2$, $q_1(X) = N_1/\sigma^2$ and $q_1(Y) = N_2/\sigma^2$ where N_1 is the number of always working sensors connected to sensor 1 and N_2 those connected to sensor 2. Then the conditions above state that at least 3 always working sensors are required for LFDIE to be a strong fault detection procedure.

For the numerical simulation, the mean parameters are $\mu_X = \mu_Y = \mu_Z = 1$ before change, and zero after change. Random variables X and Y are i.i.d. with variance σ_S^2. The common link Z has a fixed variance $\sigma_Z^2 = 1$. The prior failure rate is $d = -\log(0.01)$. Figure 4.6(a) shows a typical correlation sample path when $\sigma_S^2 = 0.2$. Notice that without time averaging it is very hard to say exactly when the change (failure) occurred.

In Section 4.4.4 we argued that the confusion probability should go to zero as the false alarm rate $\alpha \to 0$ for the procedure to be consistent, and we see this in Figure 4.6(b). Notice though that the rate depends on the uncertainty in the non-shared links σ_S^2. From Figure 4.6(c), if $\sigma_Z^2/\sigma_S^2 < 1.8$, the confusion probability is $O(\alpha^p)$ with $p < 1$, so the total false alarm rate of the procedure grows slower than α. But for higher ratios, our procedure essentially has false alarm rate α, so it is indeed valuable to have additional sensors in a neighborhood. The theoretical prediction guarantees that the procedure is strong for ratio $\sigma_Z^2/\sigma_S^2 > 3$.

Figure 4.6(d) shows the theoretical and experimental average delays obtained when the threshold is $\alpha = 10^{-7}$. There is disagreement between the curves, although the qualitative behavior is as expected. The disagreement is because our results are for $\alpha \to 0$. This discrepancy is well known in sequential analysis [Tartakovsky and Veeravalli, 2005]. In the next section we show the high accuracy of the approximation for small values of α. Figure 4.6(e) compares the behavior of our procedure using the common link Z and one that does not use it at all. There is a substantial reduction in delay using a shared link. Figure 4.6(f) is the corresponding theoretical prediction. There is a qualitative agreement between theory and simulation experiment.

4.8.2 General Networks

Now consider a fully connected network of sensors. Figure 4.7(a) shows the average detection delay for $\alpha = 0.12$ and Figure 4.7(c) for $\alpha = 10^{-20}$. As α becomes very small, our theoretical predictions agree better with experiment. Furthermore, the reduction in delay diminishes as the number of sensors increases beyond 20. Figure 4.7(b) shows the actual PFA observed for selected false alarm targets. As with the two-sensor case (in which the uncertainty ratio played the role of the number of nodes), beyond 10 sensors the false alarm probability is below the target level. Thus the confusion probability rate becomes large at that point. Figure 4.7(d) shows that with 20 nodes, the observed false alarm is always below the target level.

Lastly, we simulate a toroidal network, in which each sensor has four neighbors. The previous results lead us to believe that the average delay should remain the same independent of the number of sensors in the network, since the connectivity is fixed. Figure 4.7(e) shows this (except for when we move from 4 nodes–which is fully connected). Compare the delay level to the uncertainty ratio of 5 or a fully connected network with 4 sensors. The results are close. We can see also in Figure 4.7(f) that since the connectivity is still low, the false alarm is slightly higher than the target.

4.9 Discussion

In the Chapter we developed and evaluated an algorithm for distributed online detection of faulty sensors. We proposed a set of basic assumptions and a framework based on the notion of a *fault graph* together with fundamental metrics to evaluate the performance of any sequential fault detection procedure. Then we proceeded to analyze an efficient algorithm that achieves a good performance under the proposed metrics, and even an optimal performance under certain scenarios. As far as we know, this is the first derivation of bounds on detection delay subject to false alarm constraints in a multiple fault or multiple change point setting. We validated the assumptions behind our algorithms with real data collected from a freeway monitoring application.

Our algorithm performs an implicit averaging which leverages the short term history of the samples reducing the detection delay for a fixed false alarm. Most of the proposed methods in the literature do not perform this averaging, and therefore are subject to much longer delays. Our algorithm and framework are general enough that even model based methods for computing scores, such as the one proposed in Tulone and Madden [2006] or the primitive in Jefferey et al. [2006], can benefit from the proposed procedure. That score method though might not be very efficient if the observed processes are non stationarity such as in freeway monitoring. Compared to procedures such as in Elnahrawy and Nath [2004] and in Ould-Ahmed-Vall et al. [2007], our method benefits from implicit averaging, whereas those methods make sequential decisions based on only the current observation.

One important feature of the proposed procedure is that weak sources of evidence can be combined to give a reliable detection of failure. As long as the average correlation when a sensor is working is slightly larger then when it has failed detection can be performed reliably. Notice that very large uncertainties are tolerated, although detection delays increase. On the other hand, as more neighboring sensors are added, the shared information can be used to reduce delays. This means that in situations where fault periods are short can still be detected. Some straightforward adaptation of the algorithm also allows for detecting when a malfunctioning sensor might return to give reasonable readings in intermittent detection scenarios.

Although we focused on the case where the distribution of the correlations is approximately Gaussian, in case other score metrics are used, the proposed algorithm can be adapted for different statistical distributions. As avenues for future work we propose to investigate the estimation of the fault graph, currently based on geographic proximity, and generalizations of the methodology to applications such as event detection.

4.10 Technical Assumptions

Some technical assumptions are required in order to obtain performance estimates for the procedures proposed in the Chapter. The first assumption is that priors have tail bounds.

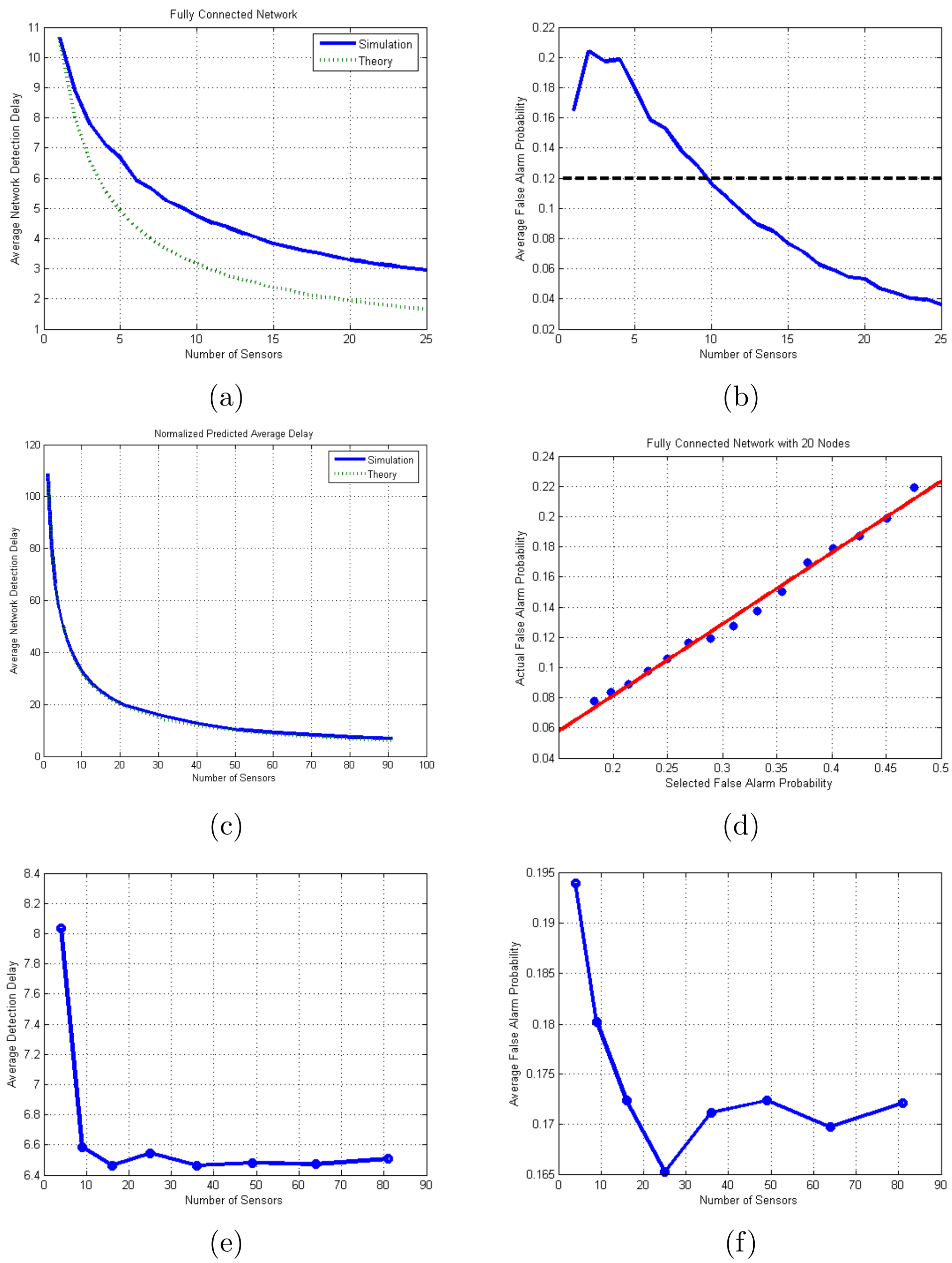


Figure 4.7. Fully Connected Network: (a)Detection Delay as a function of the number of sensors for $\alpha = 0.12$ and (b)Empirical average false alarm. (c)Detection Delay as a function of the number of sensors for $\alpha = 10^{-20}$ and (d) Selected false alarm rate and actual rate for network with 20 nodes. Grid Network: (e)Average Detection Delay as a function of number of sensors and (f) False alarm rate. Chosen false alarm rate $\alpha = 0.12$.

Assumption 4.10.1. *The priors π_1 and π_2 of sensors 1 and 2 satisfy the tail limit:*

$$\lim_{k_1 \to \infty} \frac{\Pi^1_{k_1+1}}{k_1} = -d_1, \tag{4.10.1}$$

$$\lim_{k_2 \to \infty} \frac{\Pi^2_{k+2+1}}{k_2} = -d_2. \tag{4.10.2}$$

The next assumption is on the tails of the log-likelihood random variables.

Assumption 4.10.2. *Assume either of the following:*

(a) Log likelihood ratios are independent and have finite first and second moment. Denote the variance of the likelihood ratio of X under f_0 by $\sigma_0^2(X)$ and under f_1 by $\sigma_1^2(X)$, of Y by $\sigma_0^2(Y)$ and $\sigma_1^2(Y)$ and of Z by $\sigma_0^2(Z)$ and $\sigma_1^2(Z)$. For concreteness, consider the likelihood ratio for X, $R^n_r(X)$. Then we assume the following tail bounds exist for $x > \mu^r_n(X)$,

$$\mathbb{P}_{k_1,k_2}\left(R^r_n(X) > x\right) \le K(X) \exp -\frac{(x - \mu^r_n(X))^2}{\sigma^r_n(X)^2} \tag{4.10.3}$$

where

$$\mu^r_n(X) = (n - k_1)q_1(X) - (k_1 - r)_+ q_0(X), \tag{4.10.4}$$

$$\sigma^r_n(X)^2 = \gamma(X)\{(n - k_1)\sigma_1^2(X) + (k_1 - r)_+\sigma_0^2(X)\}. \tag{4.10.5}$$

Similar bounds hold for Y and Z, with μ and σ appropriately defined. Also, we assume the bounds for sums, such as $R^r_n(X) + R^r_n(Z)$, by again using the appropriate definitions, such as $\mu^r_n(X, Z) = \mu^r_n(X) + \mu^r_n(Z)$ and $\sigma^r_n(X, Z)^2 = \sigma^r_n(X) + \sigma^r_n(Z)$. The constants for the bounds are defined as $K(X, Y)$ and $\gamma(X, Z)$.

(b) All log likelihood ratios are bounded within interval $[-M, M]$. Using Hoeffding's bound [Grimmett and Stirzaker, 1992], we can obtain a bound similar to Equation (4.10.3) for each random variable, except that in this case $\gamma(X) = 2$ and

$$\sigma^r_n(X)^2 = 2\{(n - k_1)\sigma_1^2(X) + (k_1 - r)_+\sigma_0^2(X) + M/3\}. \tag{4.10.6}$$

The tail bound assumption is not overly restrictive. In fact, it only imposes a light tail constraint on the individual likelihood random variables, and then uses independence. Some cases where this happens is when f_0 and f_1 are Gaussian densities, or both densities have bounded domain. In the first case, the tail bounds can be obtained from large deviations, and in the second case from Hoeffding's inequality. The rationale behind these assumptions is that it allows precise computation of the probability of deviations of the likelihood ratio sequence maximum. We then assume different forms of expectation concentration of the log-likelihood.

Assumption 4.10.3. *For all $\epsilon > 0$ and $k_1, k_2 \ge 1$, as $N \to \infty$:*

$$
\begin{aligned}
\mathbb{P}_{k_1,k_2}\left(\frac{1}{N}\max_{1\le n\le N} R^{k_1}_{k_1+n}(X) > (1+\epsilon)q_1(X)\right) &\to 0\\
\mathbb{P}_{k_1,k_2}\left(\frac{1}{N}\max_{1\le n\le N} R^{k_1\wedge k_2}_{k_1\wedge k_2+n}(Z) > (1+\epsilon)q_1(Z)\right) &\to 0\\
\mathbb{P}_{k_1,k_2}\left(\frac{1}{N}\max_{1\le n\le N} R^{k_2}_{k_2+n}(Y) > (1+\epsilon)q_1(Y)\right) &\to 0 \qquad (4.10.7)
\end{aligned}
$$

Assumption 4.10.4 (r-quick convergence of LLR)**.** *The log-likelihood ratios* $R^{k_1}_{k_1+n-1}(X)$, $R^{k_1\wedge k_2}_{k_1\wedge k_2+n-1}(Z)$ *and* $R^{k_2}_{k_2+n-1}(Y)$ *define the stopping times:*

$$
\begin{aligned}
T^{(k_1,k_2)}_{\epsilon}(X) &= \sup\left\{n\ge 1 : \left|\frac{1}{n}R^{k_1}_{k_1+n-1}(X) - q_1(X)\right| > \epsilon\right\}\\
T^{(k_1,k_2)}_{\epsilon}(Y) &= \sup\left\{n\ge 1 : \left|\frac{1}{n}R^{k_1}_{k_1+n-1}(Y) - q_1(Y)\right| > \epsilon\right\}\\
T^{(k_1,k_2)}_{\epsilon}(Z) &= \sup\left\{n\ge 1 : \left|\frac{1}{n}R^{k_1\wedge k_2}_{k_1\wedge k_2+n-1}(Z) - q_1(Z)\right| > \epsilon\right\} \qquad (4.10.8)
\end{aligned}
$$

For all $\epsilon > 0$ *and* $k_1 \ge 1$ *and* $k_2 \ge 1$*, for some* $r \ge 1$*:*

$$
\begin{aligned}
\mathbb{E}_{k_1,k_2}\left[T^{(k_1,k_2)}_{\epsilon}(X)\right]^r &< \infty\\
\mathbb{E}_{k_1,k_2}\left[T^{(k_1,k_2)}_{\epsilon}(Y)\right]^r &< \infty\\
\mathbb{E}_{k_1,k_2}\left[T^{(k_1,k_2)}_{\epsilon}(Z)\right]^r &< \infty\\
\sum_{k_1=1}^{\infty}\sum_{k_2=1}^{\infty}\pi_1(k_1)\pi_2(k_2)\mathbb{E}_{k_1,k_2}\left[T^{(k_1,k_2)}_{\epsilon}(X)\right]^r &< \infty\\
\sum_{k_1=1}^{\infty}\sum_{k_2=1}^{\infty}\pi_1(k_1)\pi_2(k_2)\mathbb{E}_{k_1,k_2}\left[T^{(k_1,k_2)}_{\epsilon}(Y)\right]^r &< \infty\\
\sum_{k_1=1}^{\infty}\sum_{k_2=1}^{\infty}\pi_1(k_1)\pi_2(k_2)\mathbb{E}_{k_1,k_2}\left[T^{(k_1,k_2)}_{\epsilon}(Z)\right]^r &< \infty \qquad (4.10.9)
\end{aligned}
$$

Assumption 4.10.5. *Let*

$$
\begin{aligned}
S^{k_1}_n(X) &:= \log\frac{\pi_1(k_1)}{\Pi_1(n)} + R^{k_1}_n(X) + R^{k_1}_n(Z)\\
S^{k_2}_n(Y) &:= \log\frac{\pi_2(k_2)}{\Pi_2(n)} + R^{k_2}_n(Y) + R^{k_2}_n(Z) \qquad (4.10.10)
\end{aligned}
$$

Let $\eta_1 = \min\{n : S_n^{k_1}(X) \geq \log B_\alpha\}$*, and define for arbitrary* $\epsilon > 0$*,*

$$T_\epsilon^{k_1} = \sup\{n : |(n - k_1 + 1)^{-1} S_n^{k_1}(X) - (q_1(X) + q_1(Z) + d_1)| \geq \epsilon\}. \tag{4.10.11}$$

Assume that $\mathbb{E}_{\infty,k_1} \exp T_\epsilon^{k_1} < \infty$ *for any* $\epsilon > 0$ *and for any* k_1*.*

Similarly, let $\eta_2 = \min\{n : S_n^{k_2}(Y) \geq \log B_\alpha\}$*, and define for arbitrary* $\epsilon > 0$*,*

$$T_\epsilon^{k_2} = \sup\{n : |(n - k_2 + 1)^{-1} S_n^{k_2}(Y) - (q_1(Y) + q_1(Z) + d_2)| \geq \epsilon\}. \tag{4.10.12}$$

Assume that $\mathbb{E}_{\infty,k_2} \exp T_\epsilon^{k_2} < \infty$ *for any* $\epsilon > 0$ *and for any* k_2*.*

4.11 Proofs

4.11.1 Proof of Theorem 4.4.1

Proof. We prove the statement for $\nu_1(X, Z)$. The proof for $\nu_2(Y, Z)$ follows along the same lines. We start the proof by defining an upper bound to the test statistic $\Lambda_n^{\text{noex}}(X, Z)$ that defines the stopping time $\nu_1(X, Z)$. Selecting $\bar{k}_2 = k_1 \wedge k_2$, using the assumption $\pi_2(\bar{k}_2) > 0$, we can lower bound:

$$b_n \geq \Pi_{1,n} \, \mathrm{L}_{n+1}(\mathbf{X}_n^1) \pi_2(\bar{k}_2) \mathrm{L}_{\bar{k}_2}(\mathbf{Z}_n^1),$$

so that simple algebra shows

$$\frac{a_n}{b_n} \leq \Pi_{1,n}^{-1} \pi_2(\bar{k}_2)^{-1} \sum_{k_1=1}^{n} \sum_{k_2=1}^{\infty} \pi_1(k_1) \pi_2(k_2) S_n^{k_1}(X). \tag{4.11.1}$$

Now we can proceed as

$$\begin{aligned}
\log \frac{a_n}{b_n} &\leq -\log \Pi_{1,n} - \log \pi_2(\bar{k}_2) + \log \sum_{k_1=1}^{n} \sum_{k_2=1}^{\infty} \pi_1(k_1) \pi_2(k_2) S_n^{k_1}(X), \\
&\leq -\log \Pi_{1,n} + \log \sum_{k_1=1}^{n} \pi_1(k_1) \, S_n^{k_1}(X) \qquad\qquad := \log \, \Lambda_n(X) \\
&\quad - \log \pi_2(\bar{k}_2) \qquad := r_n,
\end{aligned}$$

where we notice that the test statistic can be upper bounded by the sum of the standard Shyryaev test statistic for X with change point at λ_1 and a quantity r_n which depends only on the shared link Z. Define the stopping time

$$\eta = \inf\left\{n : \log \, \Lambda_n(X) + r_n \geq \log \frac{1 - \alpha}{\alpha}\right\}.$$

It is clear that $\nu_1(X,Z) \geq \eta$. So we have the following chain of inequalities using the definition of the delays:

$$\begin{aligned}
\mathbb{E}_{k_1,k_2}\left[(\nu_1(X,Z)-k_1)^m | \nu_1(X,Z) \geq k_1\right] &= \frac{\mathbb{E}_{k_1,k_2}\left[[(\nu_1(X,Z)-k_1)^+]^m\right]}{\mathbb{P}_{k_1,k_2}\left(\nu_1(X,Z) \geq k_1\right)}, \\
&\geq \mathbb{E}_{k_1,k_2}\left[[(\nu_1(X,Z)-k_1)^+]^m\right], \\
&\geq \mathbb{E}_{k_1,k_2}\left[[(\eta-k_1)^+]^m\right], \\
&\geq (L_{\alpha,\epsilon})^m \, \mathbb{P}_{k_1,k_2}(\eta \geq k_1 + L_{\alpha,\epsilon}),
\end{aligned}$$

where $L_{\alpha,\epsilon} = (1-\epsilon)\frac{-\log\alpha}{q_1(X)+d}$ and in the last line we used the Markov inequality. Now we want to show $\mathbb{P}_{k_1,k_2}(\eta \geq k_1 + L_{\alpha,\epsilon}) \to 1$ as $\alpha \to 0$, which implies that η is asymptotically equivalent to the stopping time $\nu_1(X)$. We can use a stopping time comparison principle:

$$\begin{aligned}
\mathbb{P}_{k_1,k_2}(\eta \geq k_1 + L_{\alpha,\epsilon}) &= \sum_{j=1}^{k_1+L_{\alpha,\epsilon}} \mathbb{P}_{k_1,k_2}\left(\log\Lambda_j(X) + r_j < \log B_\alpha\right), \\
&\geq \sum_{j=1}^{k_1+L_{\alpha,\epsilon}} \left[\mathbb{P}_{k_1,k_2}\left(\log\Lambda_j(X) < (1-\epsilon)\log B_\alpha\right)\right. \\
&- \left.\mathbb{P}_{k_1,k_2}\left(r_j \geq \epsilon\log B_\alpha\right)\right], \\
&= \mathbb{P}_{k_1,k_2}(\nu_1(X) \geq k_1 + L_{\alpha,\epsilon}) - \sum_{j=1}^{k_1+L_{\alpha,\epsilon}} \mathbb{P}_{k_1,k_2}\left(r_j \geq \epsilon\log B_\alpha\right),
\end{aligned}$$

where we used the lower bound $\mathbb{P}(X+Y < a+b) \geq \mathbb{P}(X<a) - \mathbb{P}(Y>b)$ for any random variables X and Y. Also, we identified the probability $\mathbb{P}_{k_1,k_2}(\nu_1(X) \geq k_1 + L_{\alpha,\epsilon})$ from the definition of the probability for a single change point problem. Since $\nu_1(X)$ does not depend on λ_2 Using Lemma 4.4.2 it is clear that $\mathbb{P}_{k_1,k_2}(\nu_1(X) \geq k_1 + L_{\alpha,\epsilon}) \to 1$ (notice we are running a test with false alarm $\alpha^{1-\epsilon}$) as

$$\mathbb{P}_{k_1,k_2}(\nu_1(X) \geq k_1 + L_{\alpha,\epsilon}) = 1 - \mathbb{P}_{k_1,k_2}(\nu_1(X) < k_1) - \mathbb{P}_{k_1,k_2}(k_1 \leq \nu_1(X) \leq k_1 + L_{\alpha,\epsilon}).$$

Notice r_j is a deterministic finite quantity, and thus $\mathbb{P}_{k_1,k_2}\left(r_j \geq \epsilon\log B_\alpha\right) = 0$, as soon as $\alpha \leq 1/(1+\exp(r_j/\epsilon))$.

Lemma 4.4.2 also states $\mathbb{P}_{\pi_1,\pi_2}(\nu_1(X) \geq \lambda_1 + L_{\alpha,\epsilon}) \to 1$. Also $\mathbb{P}_{\pi_1,\pi_2}\left(r_j \geq \epsilon\log B_\alpha\right) = 0$ since for each pair (k_1,k_2) the equality holds for small finite enough α, and from the definition $0 \leq \mathbb{P}_{\pi_1,\pi_2}\left(r_j \geq \epsilon\log B_\alpha\right) \leq 1$, so the dominated convergence theorem applies. □

4.11.2 Proof of Theorem 4.4.2

Before the main proof, we state an auxiliary lemma.

Lemma 4.11.1. *For any stopping time $\nu \in \mathcal{F}^n_{X,Y}$, with PFA$^{\pi_1,\infty}(\nu) < \alpha$, the following bound holds:*

$$\mathbb{P}_{\lambda_1,\lambda_2}(\nu < \lambda_1, \nu < \lambda_2) < \alpha. \tag{4.11.2}$$

Proof. First, notice that:

$$\begin{aligned}
\sum_{k_1=1}^{\infty} \pi_1(k_1)\mathbb{P}_{k_1,\infty}(\nu < k_1) &= \mathbb{P}_{\lambda_1,\infty}(\nu < \lambda_1) \\
&= \mathrm{PFA}^{\pi_1,\infty}(\nu) \\
&< \alpha
\end{aligned}$$

We can now proceed to prove the lemma:

$$\begin{aligned}
\mathbb{P}_{\lambda_1,\lambda_2}(\nu < \lambda_1, \nu < \lambda_2) &= \sum_{k_1=1}^{\infty}\sum_{k_2=1}^{\infty} \pi_1(k_1)\pi_2(k_2)\mathbb{P}_{k_1,k_2}(\nu < k_1, \nu < k_2), \\
&= \sum_{k_1=1}^{\infty}\sum_{k_2=k_1+1}^{\infty} \pi_1(k_1)\pi_2(k_2)\mathbb{P}_{k_1,\infty}(\nu < k_1) + \sum_{k_2=1}^{\infty}\sum_{k_1=k_2+1}^{\infty} \pi_1(k_1)\pi_2(k_2)\mathbb{P}_{\infty,k_2}(\nu < k_2), \\
&= \sum_{k_1=1}^{\infty}\sum_{k_2=k_1+1}^{\infty} \pi_1(k_1)\pi_2(k_2)\mathbb{P}_{\infty,\infty}(\nu < k_1) + \sum_{k_2=1}^{\infty}\sum_{k_1=k_2+1}^{\infty} \pi_1(k_1)\pi_2(k_2)\mathbb{P}_{\infty,\infty}(\nu < k_2), \\
&< \sum_{k_1=1}^{\infty}\sum_{k_2=k_1+1}^{\infty} \pi_1(k_1)\pi_2(k_2)\mathbb{P}_{\infty,\infty}(\nu < k_1) + \sum_{k_2=1}^{\infty}\sum_{k_1=k_2+1}^{\infty} \pi_1(k_1)\pi_2(k_2)\mathbb{P}_{\infty,\infty}(\nu < k_1), \\
&= \sum_{k_2=1}^{\infty} \pi_2(k_2) \sum_{k_1=1}^{\infty} \pi_1(k_1)\mathbb{P}_{k_1,\infty}(\nu < k_1) < \alpha.
\end{aligned}$$

□

The false alarm for LFDIE can be bounded using Lemma 4.11.1:

Proof. (of Theorem). **(a)** First we show item (a) for sensor 1. The analysis is analogous for sensor 2.

$$\begin{aligned}
\mathbb{P}_{\lambda_1,\lambda_2}(\bar{\nu}_1 < \lambda_1) &= \mathbb{P}_{\lambda_1,\lambda_2}(\bar{\nu}_1 < \lambda_1, \bar{\nu}_1 > \bar{\nu}_2) + \mathbb{P}_{\lambda_1,\lambda_2}(\bar{\nu}_1 < \lambda_1, \bar{\nu}_1 \le \bar{\nu}_2) && (4.11.3) \\
&= \mathbb{P}_{\lambda_1,\lambda_2}(\max(\tilde{\nu}_1, \nu_2) < \lambda_1, \nu_1 > \nu_2) + \mathbb{P}_{\lambda_1,\lambda_2}(\bar{\nu}_1 < \lambda_1, \bar{\nu}_1 \le \bar{\nu}_2) && (4.11.4) \\
&< \mathbb{P}_{\lambda_1,\lambda_2}(\tilde{\nu}_1 < \lambda_1, \nu_1 > \nu_2) + \mathbb{P}_{\lambda_1,\lambda_2}(\bar{\nu}_1 < \lambda_1, \bar{\nu}_1 \le \bar{\nu}_2) && (4.11.5) \\
&< \alpha + \mathbb{P}_{\lambda_1,\lambda_2}(\bar{\nu}_1 < \lambda_1, \bar{\nu}_1 \le \bar{\nu}_2) && (4.11.6) \\
&= \alpha + \mathbb{P}_{\lambda_1,\lambda_2}(\nu_1 < \lambda_1, \nu_1 < \lambda_2, \nu_1 \le \nu_2) + \mathbb{P}_{\lambda_1,\lambda_2}(\lambda_2 < \bar{\nu}_1 < \lambda_1, \bar{\nu}_1 \le \bar{\nu}_2) && (4.11.7) \\
&\le 2\,\alpha + \xi^{\alpha}_{\lambda_1,\lambda_2}(\bar{\nu}_1) && (4.11.8)
\end{aligned}$$

In lines (4.11.4) and (4.11.7) we use the following observations from the definitions of $\bar{\nu}_1$ and $\bar{\nu}_2$:

$$\begin{aligned}
\{\bar{\nu}_1 > \bar{\nu}_2\} \cap \{\bar{\nu}_1 < x\} &= \{\nu_1 > \nu_2\} \cap \{\max(\tilde{\nu}_1, \nu_2) < x\} \\
\{\bar{\nu}_1 \le \bar{\nu}_2\} \cap \{\bar{\nu}_1 < x\} &= \{\nu_1 \le \nu_2\} \cap \{\nu_1 < x\}
\end{aligned}$$

In line (4.11.5) we used the fact that $\tilde{\nu}_1 = \nu_S(Y)$ and so does not depend on the behavior of λ_2, so the following upper bound holds:

$$\begin{aligned}\mathbb{P}_{\lambda_1,\lambda_2}(\tilde{\nu}_1 < \lambda_1, \nu_1 > \nu_2) &< \mathbb{P}_{\lambda_1,\lambda_2}(\tilde{\nu}_1 < \lambda_1) \\ &= \mathbb{P}_{\lambda_1,\infty}(\tilde{\nu}_1 < \lambda_1) \\ &< \alpha\end{aligned}$$

For line (4.11.8) we notice that Lemma 4.11.1 applies. Proceeding in a similar fashion we can obtain the result for the false alarm of sensor 2. **(b)** Now we can show (b) for sensor 1.

From the definition of marginal probability of false alarm in Equation (4.4.26) and following the proof steps in Eqns (4.11.4,4.11.5,4.11.7):

$$\begin{aligned}\mathbb{P}_{\lambda_1,k_2}(\bar{\nu}_1 < \lambda_1) &< \mathbb{P}_{\lambda_1,k_2}(\tilde{\nu}_1 < \lambda_1, \nu_1 > \nu_2) + \mathbb{P}_{\lambda_1,k_2}(\bar{\nu}_1 < \lambda_1, \bar{\nu}_1 \leq \bar{\nu}_2), \\ &= \mathbb{P}_{\lambda_1,\infty}(\tilde{\nu}_1 < \lambda_1, \nu_1 > \nu_2) + \mathbb{P}_{\lambda_1,k_2}(\bar{\nu}_1 < \lambda_1, \bar{\nu}_1 \leq \bar{\nu}_2), \\ &\leq \alpha + \mathbb{P}_{\lambda_1,k_2}(\bar{\nu}_1 < \lambda_1, \bar{\nu}_1 \leq \bar{\nu}_2),\end{aligned}$$

The second quantity can be bound:

$$\begin{aligned}\mathbb{P}_{\lambda_1,k_2}(\bar{\nu}_1 < \lambda_1, \bar{\nu}_1 \leq \bar{\nu}_2) &= \mathbb{P}_{\lambda_1,k_2}(\nu_1 < \nu_2, \nu_1 < k_1), \\ &= \mathbb{P}_{\lambda_1,k_2}(\nu_1 < \lambda_1, \nu_1 < k_2, \nu_1 \leq \nu_2) + \mathbb{P}_{\lambda_1,k_2}(k_2 < \bar{\nu}_1 < k_1, \bar{\nu}_1 \leq \bar{\nu}_2), \\ &\leq \mathbb{P}_{\lambda_1,k_2}(\nu_1 < \lambda_1, \nu_1 < k_2) + \xi_{\lambda_1,k_2}(\bar{\nu}_1), \\ &< \mathbb{P}_{\lambda_1,\infty}(\nu_1 < \lambda_1) + \xi_{\lambda_1,k_2}(\bar{\nu}_1), \\ &< \alpha + \xi_{\lambda_1,k_2}(\bar{\nu}_1).\end{aligned}$$

□

4.11.3 Proof of Lemma 4.4.1

Proof. The proof has five parts. In the first part we decompose the probability into three tail events that determine the α-order of the confusion probability. The point at which we switch between the first two events is a parameter ($\tilde{C}_\alpha$) that needs to be optimized. For each event we compute upper bounds to the probabilities and the rate function for the speed with which the confusion probability converges to zero as $\alpha \to 0$. Using rate matching, we optimize the free parameter $\tilde{C}_\alpha$. Finally, we determine the parameter (C_α), that is when one switches from the second to the third event, based on the choice of optimized parameter.

Decomposing the confusion lemma into 3 events. First notice that (we consider $C_\alpha = \infty$ a valid possibility):

$$\mathbb{P}_{\infty,k_2}(k_2 \leq \nu_1 \leq \nu_2) \leq \mathbb{P}_{\infty,k_2}(k_2 \leq \nu_1 \leq \nu_2, \nu_2 \leq k_2 + C_\alpha) + \mathbb{P}_{\infty,k_2}(\nu_2 > k_2 + C_\alpha).$$

We decompose further the quantity:

$$\mathbb{P}_{\infty,k_2}(k_2 \le \nu_1 \le \nu_2, \nu_2 \le k_2 + C_\alpha) \le \mathbb{P}_{\infty,k_2}\left(\bigcup_{l=k_2}^{k_2+C_\alpha} \{\Lambda_l(X,Z) \ge B_\alpha\} \cap \{\Lambda_l(Y,Z) < B_\alpha\}\right),$$

where the bound follows from the definition of ν_1 and ν_2. The advantage of this particular bound is that for small l, the first event - sensor 1 mistakenly crossing the threshold -of the intersection has small probability, and for large l the second does - sensor 2 not crossing the threshold before sensor 1. From definition of the test quantities (Equation 4.4.7), we obtain the bounds:

$$\log \Lambda_n(X,Z) \le -\log \Pi_1(n) + \max_{r\in[1,n]} \{R_n^r(X) + R_n^r(Z)\}.$$

Now we can continue to bound:

$$\begin{aligned}
\mathbb{P}_{\infty,k_2}(k_2 \le \nu_1 \le \nu_2, \nu_2 \le k_2 + C_\alpha) &\le \sum_{l=k_2}^{k_2+C_\alpha} \mathbb{P}_{\infty,k_2}\left(\{\Lambda_l(X,Z) \ge B_\alpha\} \cap \{\Lambda_l(Y,Z) < B_\alpha\}\right),\\
&\le \sum_{l=k_2}^{k_2+C_\alpha} \mathbb{P}_{\infty,k_2}(\{-\log \Pi_1(l) + \max_{r\in[1,l]} \{R_l^r(X) + R_l^r(Z)\} \ge \log B_\alpha\} \cap\\
&\quad \{-\log \Pi_2(l) + \log \pi_2(r) + R_l^r(Y) + R_l^r(Z) < \log B_\alpha, \forall r \le l\}),\\
&\le \sum_{l=k_2}^{k_2+C_\alpha} \mathbb{P}_{\infty,k_2}(\{-\log \Pi_1(l) + \max_{r\in[1,l]} \{R_l^r(X) + R_l^r(Z)\} \ge \log B_\alpha\} \cap\\
&\{-\log \Pi_1(l) + \log \Pi_2(l) - \log \pi_2(k_2) + \max_{r\in[1,l]}\{R_l^r(X) + R_l^r(Z)\} - R_l^{k_2}(Y) - R_l^{k_2}(Z) > \epsilon\}),\\
&\le \sum_{l=k_2}^{k_2+C_\alpha} \min\left\{\mathbb{P}_{\infty,k_2}\left(\max_{r\in[1,l]} \{R_l^r(X) + R_l^r(Z)\} \ge \log B_\alpha + \log \Pi_1(l)\right),\right.\\
&\quad \left.\mathbb{P}_{\infty,k_2}\left(\max_{r\in[1,l]}\{R_l^r(X) + R_l^r(Z)\} - R_l^{k_2}(Y) - R_l^{k_2}(Z) > V_l\right)\right\},\\
&\le \sum_{l=k_2}^{k_2+\tilde{C}_\alpha} \mathbb{P}_{\infty,k_2}\left(\max_{r\in[1,l]} \{R_l^r(X) + R_l^r(Z)\} \ge \log B_\alpha + \log \Pi_1(l)\right) +\\
&\quad + \sum_{l=k_2+\tilde{C}_\alpha}^{k_2+C_\alpha} \mathbb{P}_{\infty,k_2}\left(\max_{r\in[1,l]}\{R_l^r(X) + R_l^r(Z)\} - R_l^{k_2}(Y) - R_l^{k_2}(Z) > V_l\right),
\end{aligned}$$

where $V_l = \epsilon + \log \Pi_1(l) - \log \Pi_2(l) + \log \pi_2(k_2)$.

Analyzing the probability of early crossing for sensor 1 (event $\mathcal{E}_1$). The following lemma will be used to bound the first event:

Lemma 4.11.2. *Consider the function $f(x) = (a + b\,x)^2/(c + d\,x)$, with $a, c, d \ge 0$. The following properties hold:*

(a) If $b > 0$, the function is decreasing in the interval $x \in [0, x_{min}]$ and increasing in $x \in (x_{min}, \infty]$,where $x_{min} = a/b - 2\,c/d$ is the point of minimum and $f(x_{min}) = 4\,b^2/d\,(a/b - c/d)$.

(b) If $b \leq 0$, the function is decreasing in the interval $x \in [0, x_{min}]$ and increasing in $x \in (x_{min}, \infty]$, where $x_{min} = -a/b$ is the point of minimum and $f(x_{min}) = 0$.

Proof. Follows from noticing that the derivative is $f'(x) = (a + b\,x)(2\,bc - ad + bd\,x)/(c + d\,x)^2$. □

We analyze the first probability in the inequality. Define $b_0 = q_0(Z) + q_0(X)$ and $b_1 = q_0(X) - q_1(Z) + d_1$. Apply Lemma 4.11.2, with $a = a_1 = \log B_\alpha + \log \Pi_1(l) - l\,d_1$, $b = b_1$, $c = c_1 = (k_2 - 1)\,\sigma_0^2(X, Z)$ and $d = d_1 = \sigma_1^2(X, Z)$:

$$
\begin{aligned}
&\mathbb{P}_{\infty,k_2}\left(\max_{r\in[1,l]}\{R_l^r(X) + R_l^r(Z)\} \geq \log B_\alpha + \log \Pi_1(l)\right)\\
&\qquad\leq l \max_{r\in[1,l]} \mathbb{P}_{\infty,k_2}\left(R_l^r(X) + R_l^r(Z) \geq \log B_\alpha + \log \Pi_1(l)\right)\\
&\qquad\leq l \max_{s\in[0,k_2-1]} \max_{r\in[k_2,l]} \exp\left\{-\frac{(\log B_\alpha + \log \Pi_1(l) - l\,d_1 + (l - r + 1)\,b_1 + s\,b_0)^2}{(l - r + 1)\,\sigma_1^2(X, Z) + s\,\sigma_0^2(X, Z)}\right\}\\
&\qquad\leq l \max_{r\in[k_2,l]} \exp\left\{-\frac{(\log B_\alpha + \log \Pi_1(l) - l\,d_1 + (l - r + 1)\,b_1)^2}{(l - r + 1)\,\sigma_1^2(X, Z) + (k_2 - 1)\,\sigma_0^2(X, Z)}\right\}\\
&\qquad\leq l \exp\left\{-\frac{(\log B_\alpha + \log \Pi_1(l) - l\,d_1 + (l^* - k_2 + 1)\,b_1)^2}{(l^* - k_2 + 1)\,\sigma_1^2(X, Z) + (k_2 - 1)\,\sigma_0^2(X, Z)}\right\}
\end{aligned}
\tag{4.11.9}
$$

where (a) $l^* = \min(l, -a_1/b_1 + k_2)$ if $b_1 < 0$; (b) $l^* = \min(l, a_1/b_1 - 2\,c_1/d_1 + k_2)$ if $b_1 > 0$ and $a_1/b_1 - 2\,c_1/d_1 > 0$; $l^* = k_2$ if $b_1 > 0$ and $a_1/b_1 - 2\,c_1/d_1 < 0$; and (c) $l^* = l$ if $b_1 = 0$. Notice if $l = \Theta(\log B_\alpha)$, only the first condition will apply as $\alpha \to 0$ for option (b).

Let us assume that $\tilde{C}_\alpha = \log B_\alpha/w$. We can then control the bound using Equation 4.11.9 and a simple observation:

$$
\sum_{l=k_2}^{k_2+\tilde{C}_\alpha} \mathbb{P}_{\infty,k_2}\left(\max_{r\in[1,l]}\{R_l^r(X) + R_l^r(Z)\} \geq \log B_\alpha + \log \Pi_1(l)\right) \leq (\tilde{C}_\alpha)^2 \exp - \min \Phi_\alpha,
$$

and Φ_α is given by

$$
\begin{aligned}
\Phi_\alpha &= \frac{(\log B_\alpha + A_c(K_\alpha) + K_\alpha\,b_1)^2}{K_\alpha\,\sigma_1^2(X, Z) + (k_2 - 1)\,\sigma_0^2(X, Z)},\\
A_c(K_\alpha) &= \log \Pi_1(K_\alpha + k_2) - (K_\alpha + k_2)\,d_1.
\end{aligned}
$$

The constant K_α is chosen as to minimize Φ_α under the constraint that $0 < K_\alpha < \tilde{C}_\alpha$. By assumption on tail of prior, there exists T, such that for all $K_\alpha > T$, $|A_c(K_\alpha)| < \epsilon$. We are in this regime. Consider the case $b_1 > 0$. Our previous calculation shows minima is achieved when $K_\alpha = (\log B_\alpha - \epsilon)/b_1 - 2\,(k_2 - 1)\,\sigma_0^2(X, Z)/\sigma_1^2(X, Z)$. For vanishing α,

$K_\alpha < \tilde{C}_\alpha$ if $b_1 > w$, else we should set $K_\alpha = \log B_\alpha / w$ to minimize Φ_α. Lemma 4.11.2 can be used to compute the rate at the minimum when either $b_1 > w$ or $b_1 \le w$:

$$\Phi_{\alpha,\min} = 4\,\frac{b_1^2}{\sigma_1^2(X,Z)}\left(\frac{\log B_\alpha - \epsilon}{b_1} - \frac{(k_2-1)\,\sigma_0^2(X,Z)}{\sigma_1^2(X,Z)}\right) \text{ for } \; b_1 > w$$
$$= \frac{\left(\log B_\alpha\left[1+\frac{b_1}{w}\right]-\epsilon\right)^2}{\log B_\alpha\,\frac{\sigma_1^2(X,Z)}{w} + (k_2-1)\,\sigma_0^2(X,Z)} \;\; \text{for } \; b_1 \le w$$

The rate that the probability goes to zero is then calculated as:

$$\lim_{\alpha\to 0}\frac{-\log[(\tilde{C}_\alpha)^2\exp -\Phi_\alpha]}{\log B_\alpha} = 4\,\frac{b_1}{\sigma_1^2(X,Z)} \;\; \text{for } \; b_1 > w,$$
$$= \frac{w}{\sigma_1^2(X,Z)}\left[1+\frac{b_1}{w}\right]^2 \;\; \text{for } \; b_1 \le w.$$

We can proceed similarly for the case $b_1 \le 0$. Notice that to obtain a vanishing probability now, we need $K_\alpha < \log B_\alpha / -b_1$, so the only interesting case is when $w > -b_1$ (else $\Phi_\alpha = 0$ is the minimum). Since for $b_1 < 0$, the function first decreases to the minimum, we can conclude that in this case:

$$\lim_{\alpha\to 0}\frac{-\log[(\tilde{C}_\alpha)^2\exp -\Phi_\alpha]}{\log B_\alpha} = \frac{w}{\sigma_1^2(X,Z)}\left[1+\frac{b_1}{w}\right]^2.$$

Analyzing the probability of sensor 2 crossing after sensor 1 (event $\mathcal{E}_2$). Let $\tilde{V}_l = \epsilon + \log\Pi_1(l) - d_1 l - \log\Pi_2(l) + d_2 l + \log\pi_2(k_2)$, $q_y(l) = (l-k_2+1)q_1(Y)$ and $\sigma_y^2(l) = (l-k_2+1)\sigma_1^2(Y)$. Similarly, for the second probability, we bound:

$$\mathbb{P}_{\infty,k_2}\left(\max_{r\in[1,l]}\{R_l^r(X)+R_l^r(Z)\} - R_l^{k_2}(Y) - R_l^{k_2}(Z) > V_l\right)$$
$$\le l\max_{r\in[1,l]}\mathbb{P}_{\infty,k_2}\left(R_l^r(X)+R_l^r(Z) - R_l^{k_2}(Y) - R_l^{k_2}(Z) \ge V_l\right)$$
$$\le l\max_{r\in[1,l]}\exp\left\{-\frac{(V_l+(l-r+1)q_0(X)+q_y(l)+[k_2-r]_+q_0(Z)+[r-k_2]_+q_1(Z))^2}{(l-r+1)\sigma_0^2(X)+\sigma_y^2(l)+[k_2-r]_+\sigma_0^2(Z)+[r-k_2]_+\sigma_1^2(Z)}\right\}$$
$$\le l\max_{r\in[1,l]}\exp\left\{-\frac{(V_l+(l-r)q_0(X)+(r-k_2)q_1(Z)+q_y(l)+q_0(X))^2}{(l-r)\,\sigma_0^2(X)+r\sigma_1^2(Z)+\sigma_y^2(l)+k_2\sigma_0^2(Z)+\sigma_0^2(X)}\right\}$$
$$\le l\,\exp\left\{-\frac{(A_e(l)+l[q_{i^*}+q_1(Y)+d_1-d_2])^2}{C_e+l[\sigma_{i^*}^2+\sigma_1^2(Y)]}\right\},$$

where $q_{i^*} = \min(q_0(X), q_1(Z))$, $\sigma_{i^*}^2 = \max(\sigma_0^2(X), \sigma_1^2(Z))$, $A_e(l) = \tilde{V}_l - k_2[q_1(Y)+q_1(Z)] + q_0(X) + q_1(Y)$ and $C_e = k_2[\sigma_0^2(Z) - \sigma_1^2(Y)] + \sigma_0^2(X) + \sigma_1^2(Y)$.

To continue the analysis, we compute the rates for the second major event:

$$\sum_{l=k_2+\tilde{C}_\alpha}^{k_2+C_\alpha} \mathbb{P}_{\infty,k_2}\left(\max_{r\in[1,l]}\{R_l^r(X)+R_l^r(Z)\} - R_l^{k_2}(Y) - R_l^{k_2}(Z) > V_l\right) \leq$$

$$(C_\alpha - \tilde{C}_\alpha)^2 \exp - \min \tilde{\Phi}_\alpha,$$

$$\tilde{\Phi}_\alpha = \frac{\left(A_e(\tilde{K}_\alpha) + \tilde{K}_\alpha[q_{i^*} + q_1(Y) + d_1 - d_2]\right)^2}{C_e + \tilde{K}_\alpha[\sigma_{i^*}^2 + \sigma_1^2(Y)]}.$$

Lemma 4.11.2 implies the minimum in this case is at $\tilde{K}_\alpha = A_e(l))/(q_{i^*} + q_1(Y) + d_1 - d_2)$. But since this is a small quantity compared to $\tilde{C}_\alpha + k_2$, assuming $(q_{i^*} + q_1(Y) + d_1 - d_2) > 0$, we have that the minimum happens at $\tilde{K}_\alpha = k_2 + \tilde{C}_\alpha$, as the function is increasing after the minima. Using similar arguments as for the first major event, it is straightforward to show that the rate function satisfies:

$$\lim_{\alpha\to 0} \frac{-\log[(C_\alpha - \tilde{C}_\alpha)^2 \exp -\tilde{\Phi}_\alpha]}{\log B_\alpha} = \frac{1}{w}\frac{[q_{i^*} + q_1(Y) + d_1 - d_2]^2}{\sigma_{i^*}^2 + \sigma_1^2(Y)}.$$

Selecting the optimizing rate. Given the bounds we have computed, the problem reduces to selecting the constant $\tilde{C}_\alpha$ so that the best rate is obtained for $\mathbb{P}_{\infty,k_2}(k_2 \leq \nu_1 \leq \nu_2)$. In rate matching, we have two rates $r_1(w)$ and $r_2(w)$, and would like to maximize the *minimum* of both, i.e., $\max\min(r_1(w), r_2(w))$, which is obtained by setting w such that $r_1(w) = r_2(w)$,where For convenience define the rate function:

$$r_1(w) = \frac{w}{\sigma_1^2(X,Z)}\left[1 + \frac{b_1}{w}\right]^2$$

$$r_2(w) = \frac{1}{w}\frac{[q_{i^*} + q_1(Y) + d_1 - d_2]^2}{\sigma_{i^*}^2 + \sigma_1^2(Y)} \text{ in lemma, } r(w),$$

There are three cases, since the first event has three behaviors for the rate r^*:

(1) Consider $b_1 > 0$. Then for $w < b_1$, we would like

$$r_2(w) > 4\,\frac{b_1}{\sigma_1^2(X,Z)},$$

so we select

$$w_1^* < \min\left(b_1, \frac{\sigma_1^2(X,Z)}{\sigma_{i^*}^2 + \sigma_1^2(Y)}\frac{[q_{i^*} + q_1(Y) + d_1 - d_2]^2}{4b_1}\right)$$

and get rate $r^* = 4\,\frac{b_1}{\sigma_1^2(X,Z)}$.

(2) Again let $b_1 > 0$. Then for $w \geq b_1$, we would like

$$r_2(w) = \frac{w}{\sigma_1^2(X,Z)}\left[1 + \frac{b_1}{w}\right]^2,$$

so

$$w_2^* = \sqrt{\frac{\sigma_1^2(X,Z)}{\sigma_{i^*}^2 + \sigma_1^2(Y)}}[q_{i^*} + q_1(Y) + d_1 - d_2] - b_1,$$

as long as it satisfies $w_2^* \geq b_1$. The obtained rate is $r^* = r_2(w_2^*)$. Else, set $w_2^* = b_1$, and obtain rate $r_2(b_1)$.
(3) Let $b_1 \leq 0$. Then for $w \geq -b_1$, we would like

$$r_2(w) = \frac{w}{\sigma_1^2(X,Z)}\left[1 + \frac{b_1}{w}\right]^2,$$

so

$$w_3^* = \sqrt{\frac{\sigma_1^2(X,Z)}{\sigma_{i^*}^2 + \sigma_1^2(Y)}}[q_{i^*} + q_1(Y) + d_1 - d_2] - b_1,$$

which satisfies $w_3^* \geq -b_1$. The obtained rate is $r^* = r_2(w_3^*)$.

Upper bounding detection of sensor 2 and selecting C_α (probability of event $\mathcal{E}_3$). We bound $\mathbb{P}_{\infty,k_2}(\nu_2 > k_2 + C_\alpha)$. Assume and assuming $C_\alpha = \beta \log B_\alpha$. From definition of the test quantity (Equation 4.4.7):

$$\log \Lambda_n(Y,Z) \geq -\log \Pi_2(n) + \log \pi_2(r) + R_n^r(Y) + R_n^r(Z).$$

Let $\eta = \min\{n : R_n^{k_2}(Y) + R_n^{k_2}(Z) \geq \log B_\alpha\}$, so $\nu_2 \leq \eta$. For arbitrary $\epsilon > 0$, let

$$T_\epsilon^{k_2} = \sup\{n : |(n - k_2 + 1)^{-1}[R_n^{k_2}(Y) + R_n^{k_2}(Z)] - (q_1(Y) + q_1(Z) + d_2)| \geq \epsilon\}.$$

It is simple to see that:

$$\log B_\alpha > R_{\eta-1}^{k_2}(Y) + R_{\eta-1}^{k_2}(Z) \geq (\eta - k_2)(q_1(Y) + q_1(Z) + d_2 - \epsilon) \text{ on } \{\eta - 1 \geq T_\epsilon^{k_2}\}.$$

So,

$$\begin{aligned}\nu_2 \leq \eta \leq &\left(k_2 + \frac{\log B_\alpha}{q_1(Y) + q_1(Z) + d - \epsilon}\right)\mathbb{I}(\eta \geq 1 + T_\epsilon^{k_2}) + (1 + T_\epsilon^{k_2})\mathbb{I}(\eta \geq 1 + T_\epsilon^{k_2}) \\ &\leq k_2 + \frac{\log B_\alpha}{q_1(Y) + q_1(Z) + d - \epsilon} + 1 + T_\epsilon^{k_2}.\end{aligned}$$

Using this result:

$$
\begin{aligned}
\mathbb{P}_{\infty,k_2}(\nu_2 > k_2 + C_\alpha) &\le \mathbb{P}_{\infty,k_2}(k_2 + C_\alpha \le k_2 + \frac{\log B_\alpha}{q_1(Y)+q_1(Z)+d-\epsilon} + 1 + T_\epsilon^{k_2}) \\
&\le \mathbb{P}_{\infty,k_2}\left[T_\epsilon^{k_2} + 1 \ge \log B_\alpha \left(\beta - \frac{1}{q_1(Y)+q_1(Z)+d-\epsilon}\right)\right] \\
&\le \mathbb{E}_{\infty,k_2} \exp(T_\epsilon^{k_2}+1)\left(\frac{\alpha}{1-\alpha}\right)^{\beta - \frac{1}{q_1(Y)+q_1(Z)+d_2-\epsilon}} \\
&\le O\left(\alpha^{\beta - \frac{1}{q_1(Y)+q_1(Z)+d_2-\epsilon}}\right) \quad (4.11.10)
\end{aligned}
$$

where we used Markov's inequality in the last line. Assumption 4.10.5 guarantees that $\mathbb{E}_{\infty,k_2} \exp(T_\epsilon^{k_2}+1) < \infty$. The constants in big-O are independent of k_2, ϵ. To obtain the best possible rate for the total confusion probability, we choose

$$
\beta = (1+\epsilon)\, r^* + \frac{1}{q_1(Y)+q_1(Z)+d_2-\epsilon}
$$

Concluding the proof. To put the elements of the proof together, we use the bound:

$$
\mathbb{P}_{\infty,k_2}(k_2 \le \nu_1 \le \nu_2) \le \mathbb{P}_{\infty,k_2}(\mathcal{E}_1) + \mathbb{P}_{\infty,k_2}(\mathcal{E}_2) + \mathbb{P}_{\infty,k_2}(\mathcal{E}_3),
$$

so the rate function has

$$
\begin{aligned}
\lim_{\alpha\to 0} \frac{-\log \mathbb{P}_{\infty,k_2}(k_2 \le \nu_1 \le \nu_2)}{\log B_\alpha} &\ge \lim_{\alpha\to 0} \frac{-\log 3\, \max_i \mathbb{P}_{\infty,k_2}(\mathcal{E}_i)}{\log B_\alpha} \\
&= \min_i \lim_{\alpha\to 0} \frac{-\log \mathbb{P}_{\infty,k_2}(\mathcal{E}_i)}{\log B_\alpha} = r^*.
\end{aligned}
$$

Taking the expectation with respect to λ_2, we can conclude that the results hold for the measure $\mathbb{P}_{\infty,\lambda_2}$, since k_2 only appears in either the denominator of the bound rates, or as $l - k_2$, but for $l > k_2$. □

4.11.4 Lemma 4.4.2

We state a Lemma that will be used repeatedly.

Lemma 4.11.3.

(i) *Let $\nu_1 \in \Delta_1(\alpha)$, then for all $n \le k_1 \le k_2$ $\mathbb{P}_{k_1,k_2}(\nu_1 < n) \le \frac{\alpha}{\Pi_n^1}$.*

(ii) *Let $\nu_1 \in \tilde{\Delta}_1(\alpha, k_2)$, then for all $n \le k_1$: $\mathbb{P}_{k_1,k_2}(\nu_1 < n) \le \frac{\alpha}{\Pi_n^1}$.*

(iii) *Let $\nu_1 \in \tilde{\Delta}_1(\alpha)$, then for all $n \le k_1$: $\mathbb{P}_{k_1,\lambda_2}(\nu_1 < n, \lambda_2 < k_1) \le \frac{\alpha}{\Pi_n^1}$.*

(iv) *Let $\nu_2 \in \Delta_2(\alpha)$, then for all $n \le k_2 \le k_1$ $\mathbb{P}_{k_1,k_2}(\nu_2 < n) \le \frac{\alpha}{\Pi_n^2}$.*

(v) *Let* $\nu_2 \in \tilde{\Delta}_2(\alpha, k_1)$*, then for all* $n \leq k_2$*:* $\mathbb{P}_{k_1,k_2}(\nu_1 < n) \leq \frac{\alpha}{\Pi_n^2}$.

(vi) *Let* $\nu_2 \in \tilde{\Delta}_2(\alpha)$*, then for all* $n \leq k_2$*:* $\mathbb{P}_{\lambda_1,k_2}(\nu_1 < n, \lambda_1 < k_2) \leq \frac{\alpha}{\Pi_n^2}$.

Proof. All assertions follow the same proof guideline. First notice that:

$$\begin{aligned}
\mathrm{PFA}^{\pi_1,\infty}(\nu_1) &\geq \mathbb{P}_{\lambda_1,\infty}(\{\nu_1 < n\} \cap \{\lambda_1 > n\}) \\
&= \mathbb{P}_{\lambda_1,\infty}(\nu_1 < n | \lambda_1 > n)\mathbb{P}_{\lambda_1,\infty}(\lambda_1 > n) \\
&= \mathbb{P}_{\infty,\infty}(\nu_1 < n)\Pi_n^1
\end{aligned}$$

Next, as $\nu_1(X, Z) \in \Delta_1(\alpha)$, we have $\mathrm{PFA}^{\pi_1,\infty}(\nu_1) \leq \alpha$. To conclude, for the choices of k_1,k_2 and n in the lemma $\mathbb{P}_{k_1,k_2}(\nu_1 < n) = \mathbb{P}_{\infty,\infty}(\nu_1 < n)$. □

Proof. (of Lemma). *(i)* We follow closely the argument in Tartakovsky and Veeravalli [2005]. We can first build our bound by a change of measure argument:

$$\begin{aligned}
&\mathbb{P}_{\infty,\infty}\left(k_1 \leq \nu_1 < k_1 + (1-\epsilon)L_1^\alpha\right) = \\
&\quad = \mathbb{E}_{k_1,k_2}\left\{\mathbb{I}\left(k_1 \leq \nu < k_1 + (1-\epsilon)L_1^\alpha\right) e^{-\left(R_{\nu_1}^{k_1}(X) + R_{\nu_1}^{k_1\wedge k_2}(Z)\right)}\right\} \\
&\quad \geq \mathbb{E}_{k_1,k_2}\left\{\mathbb{I}_{\left(k_1 \leq \nu < k_1 + (1-\epsilon)L_1^\alpha, R_{\nu_1}^{k_1}(X) + R_{\nu_1}^{k_1\wedge k_2}(Z) < C\right)} e^{-\left(R_{\nu_1}^{k_1}(X) + R_{\nu_1}^{k_1\wedge k_2}(Z)\right)}\right\} \\
&\quad \geq e^{-C}\mathbb{P}_{k_1,k_2}\left(k_1 \leq \nu < k_1 + (1-\epsilon)L_1^\alpha, \max_{k_1 \leq n < k_1 + (1-\epsilon)L_1^\alpha} R_n^{k_1}(X) + R_n^{k_1\wedge k_2}(Z) < C\right) \\
&\quad \geq e^{-C}\left[\mathbb{P}_{k_1,k_2}\left(k_1 \leq \nu < k_1 + (1-\epsilon)L_1^\alpha\right) - \right. \\
&\qquad \left. -\mathbb{P}_{k_1,k_2}\left(\max_{k_1 \leq n < k_1 + (1-\epsilon)L_1^\alpha} R_n^{k_1}(X) + R_n^{k_1\wedge k_2}(Z) \geq C\right)\right]
\end{aligned}$$

Choosing $C = (1-\epsilon^2)(q_1(X) + q_1(Z))L_1^\alpha$, and rearranging we obtain:

$$\begin{aligned}
\gamma_{\epsilon,\alpha}^{(k_1,k_2)} &\leq e^{(1-\epsilon^2)(q_1(X)+q_1(Z))L_1^\alpha}\mathbb{P}_{\infty,\infty}\left(k_1 \leq \nu_1 < k_1 + (1-\epsilon)L_1^\alpha\right) + \\
&+ \mathbb{P}_{k_1,k_2}\left(\max_{k_1 \leq n < k_1 + (1-\epsilon)L_1^\alpha} R_n^{k_1}(X) + R_n^{k_1\wedge k_2}(Z) \geq C\right)
\end{aligned} \tag{4.11.11}$$

We now analyze each of the two parts in the above. We start with the second term:

$$\begin{aligned}
\beta_{k_1,k_2}(\epsilon,\alpha) &= \mathbb{P}_{k_1,k_2}\left(\max_{k_1 \leq n < k_1 + (1-\epsilon)L_1^\alpha} R_n^{k_1}(X) + R_n^{k_1\wedge k_2}(Z) \geq C\right) \\
&\leq \mathbb{P}_{k_1,k_2}\left(\max_{k_1 \leq n < k_1 + (1-\epsilon)L_1^\alpha} R_n^{k_1}(X) + R_n^{k_1}(Z) \geq C\right) + \\
&+ \mathbb{P}_{k_1,k_2}\left(C - R_Z \leq \max_{k_1 \leq n < k_1 + (1-\epsilon)L_1^\alpha} R_n^{k_1}(X) + R_n^{k_1}(Z) < C, R_Z \geq 0\right) \\
&\leq \mathbb{P}_{k_1,k_2}\left(\max_{k_1 \leq n < k_1 + (1-\epsilon)L_1^\alpha} R_n^{k_1}(X) + R_n^{k_1}(Z) \geq C\right) +
\end{aligned}$$

$$
\begin{aligned}
&+ \quad \mathbb{P}_{k_1,k_2}\left(C - R_Z \leq \max_{k_1 \leq n < k_1+(1-\epsilon)L_1^\alpha} R_n^{k_1}(X) + R_n^{k_1}(Z) < C\right) \\
&= \quad \mathbb{P}_{k_1,k_2}\left(\max_{0 \leq n < (1-\epsilon)L_1^\alpha} R_{k_1+n}^{k_1}(X) + R_{k_1+n}^{k_1}(Z) \geq C\right) + \\
&+ \quad \mathbb{P}_{k_1,k_2}\left(C - R_Z \leq \max_{0 \leq n < (1-\epsilon)L_1^\alpha} R_{k_1+n}^{k_1}(X) + R_n^{k_1}(Z) < C\right) \\
&= \quad \mathbb{P}_{k_1,k_2}\left(\frac{1}{N_\alpha} \max_{0 \leq n < N_\alpha} R_{k_1+n}^{k_1}(X) + R_{k_1+n}^{k_1}(Z) \geq q_\epsilon\right) + \\
&+ \quad \mathbb{P}_{k_1,k_2}\left(q_\epsilon - \frac{R_Z}{N_\alpha} \leq \max_{0 \leq n < N_\alpha} R_{k_1+n}^{k_1}(X) + R_n^{k_1}(Z) < q_\epsilon\right)
\end{aligned}
$$

Where $R_Z = R_{k_1-1}^{k_2}(Z)$, $q_\epsilon = (1+\epsilon)(q_1(X) + q_1(Z))$ and $N_\alpha = \lfloor (1-\epsilon)L_1^\alpha \rfloor$. Now noticing that as $\alpha \to 0$ we have $N_\alpha \to \infty$, we have using assumption 4.10.3 and properties of measure:

$$
\begin{aligned}
&\mathbb{P}_{k_1,k_2}\left(\frac{1}{N_\alpha} \max_{0 \leq n < N_\alpha} R_{k_1+n}^{k_1}(X) + R_{k_1+n}^{k_1}(Z) \geq q_\epsilon\right) = \\
&= \mathbb{P}_{k_1,\infty}\left(\frac{1}{N_\alpha} \max_{0 \leq n < N_\alpha} R_{k_1+n}^{k_1}(X) + R_{k_1+n}^{k_1}(Z) \geq q_\epsilon\right) \to 0
\end{aligned}
$$

Because $\frac{R_Z}{N_\alpha} \to 0$ almost surely, we have the second probability going to zero. Thus $\beta_{k_1,k_2}(\epsilon,\alpha) \to 0$ as $\alpha \to 0$. We now proceed to bound the first probability in Equation (4.11.11), using the result from Lemma 4.11.3 and using the definition of N_α and $q = q_1(X) + q_1(Z)$:

$$
\begin{aligned}
p_{k_1,k_2}(\epsilon,\alpha) &= e^{(1-\epsilon^2)(q_1(X)+q_1(Z))L_1^\alpha} \mathbb{P}_{\infty,\infty}\left(k_1 \leq \nu_1 < k_1 + (1-\epsilon)L_1^\alpha\right) \\
&\leq e^{(1-\epsilon^2)(q_1(X)+q_1(Z))L_1^\alpha} \mathbb{P}_{\infty,\infty}\left(\nu_1 < k_1 + (1-\epsilon)L_1^\alpha\right) \\
&\leq \frac{\alpha}{\Pi_{k_1+N_\alpha}^1} e^{(1-\epsilon^2)qL_1^\alpha}
\end{aligned}
$$

Notice that $\alpha = e^{-(q+d_1)L_1^\alpha}$ from the definitions. Thus:

$$
\begin{aligned}
\frac{\log(p_{k_1,k_2}(\epsilon,\alpha))}{N_\alpha} &\leq \frac{(1-\epsilon^2)qL_1^\alpha}{N_\alpha} - \frac{(q+d_1)L_1^\alpha}{N_\alpha} - \frac{\log \Pi_{k_1+N_\alpha}^1}{N_\alpha} \\
&= \frac{(1-\epsilon^2)qL_1^\alpha}{N_\alpha} - \frac{(q+d_1)L_1^\alpha}{N_\alpha} - \frac{\log \Pi_{k_1+N_\alpha}^1}{k_1+N_\alpha} \frac{k_1+N_\alpha}{N_\alpha} \\
&\leq \frac{(1+\epsilon)q(N_\alpha+1)}{N_\alpha} - \frac{\frac{(q+d_1)}{1-\epsilon}N_\alpha}{N_\alpha} - \frac{\log \Pi_{k_1+N_\alpha}^1}{k_1+N_\alpha} \frac{k_1+N_\alpha}{N_\alpha} \\
&= -\frac{\epsilon^2 q + d_1}{1-\epsilon} - \frac{\log \Pi_{k_1+N_\alpha}^1}{k_1+N_\alpha}\left(1 + \frac{k_1}{N_\alpha}\right) + \frac{(1+\epsilon)q}{N_\alpha}
\end{aligned}
$$

Taking limits, and using the tail assumption:

$$\lim_{\alpha\to 0} \frac{\log(p_{k_1,k_2}(\epsilon,\alpha))}{N_\alpha} \leq -\frac{\epsilon^2 q + d_1}{1-\epsilon} + d_1 = -\frac{\epsilon^2 q + \epsilon d_1}{1-\epsilon}$$

It is now clear that $p_{k_1,k_2}(\epsilon,\alpha) \to 0$. We have shown that for all $\nu_1 \in \Delta_1(\alpha)$:

$$\gamma_{\epsilon,\alpha}^{(k_1,k_2)}(\nu_1) \leq \beta_{k_1,k_2}(\epsilon,\alpha) + p_{k_1,k_2}(\epsilon,\alpha) \to 0$$

We can complete the result by studying the behavior of $\gamma_{\epsilon,\alpha}$. Let $\tilde{N}_\alpha = \lfloor \epsilon L_1^\alpha \rfloor$. From the definition we now have:

$$\begin{aligned}
\gamma_{\epsilon,\alpha}(\nu_1) &= \sum_{k_1=1}^{\infty}\sum_{k_2=1}^{\infty} \pi_1(k_1)\pi_2(k_2)\gamma_{\epsilon,\alpha}^{(k_1,k_2)}(\nu_1) \\
&\leq \Pi^1_{\tilde{N}_\alpha} + \sum_{k_1=1}^{\tilde{N}_\alpha}\sum_{k_2=1}^{\tilde{N}_\alpha} \pi_1(k_1)\pi_2(k_2)(\beta_{k_1,k_2}(\epsilon,\alpha) + p_{k_1,k_2}(\epsilon,\alpha)) \\
&\leq \Pi^1_{\tilde{N}_\alpha} + \sup_{k_1 \leq \tilde{N}_\alpha} p_{k_1,k_2}(\epsilon,\alpha) + \sum_{k_1=1}^{\tilde{N}_\alpha}\sum_{k_2=1}^{\tilde{N}_\alpha} \pi_1(k_1)\pi_2(k_2)\beta_{k_1,k_2}(\epsilon,\alpha)
\end{aligned}$$

Now as $\alpha \to 0$, $\Pi^1_{\tilde{N}_\alpha} \to 0$ by definition, and the third term in the above sum goes to zero by Dominated Convergence Theorem and the fact that $\beta_{k_1,k_2}(\epsilon,\alpha) \to 0$. For the second term, we make a minor modification in the first proof of convergence of $p_{k_1,k_2}(\epsilon,\alpha)$, by noticing that Π^1_n is a non-increasing function of n:

$$\sup_{k_1 \leq \tilde{N}_\alpha} p_{k_1,k_2}(\epsilon,\alpha) \leq \frac{\alpha}{\Pi^1_{\tilde{N}_\alpha + N_\alpha}} e^{(1-\epsilon^2)qL_1^\alpha}$$

Then continuing as before, replacing k_1 by $\tilde{N}_\alpha$, we obtain:

$$\lim_{\alpha\to 0} \frac{\log(\sup_{k_1 \leq \tilde{N}_\alpha} p_{k_1,k_2}(\epsilon,\alpha))}{N_\alpha} \leq -\frac{\epsilon^2 q + d_1}{1-\epsilon} + d_1\left(1 + \frac{\epsilon}{1-\epsilon}\right) = -\frac{\epsilon^2 q}{1-\epsilon}$$

Clearly this shows that $\sup_{k_1 \leq \tilde{N}_\alpha} p_{k_1,k_2}(\epsilon,\alpha) \to 0$, concluding the proof. The proof for the third statement is the same the above, except the sum over the priors is only over the cases $\lambda_1 < \lambda_2$.

(ii) The proof is as in (i), except we use the change of measure for $k_2 < k_1$:

$$\mathbb{P}_{\infty,k_2}\left(k_1 \leq \nu_1 < k_1 + (1-\epsilon)L_1^\alpha\right) = \mathbb{E}_{k_1,k_2}\left\{\mathbb{I}\left(k_1 \leq \nu < k_1 + (1-\epsilon)L_1^\alpha\right) e^{-\left(R_{\nu_1}^{k_1}(X)\right)}\right\}$$

For $k_1 \leq k_2$ we use the same change of measure as in (i).

Also, we can use Lemma 4.11.3. For the other cases the proofs proceed similarly. $\square$

4.11.5 Proof of Theorem 4.4.5

We divide the proof into computing an upper bound (item (a)) and the lower bound (item (b)). First, we compute the upper bound. . We start with the definition in Equation (4.4.14). Denote by $\nu_1 = \nu(X, Z)$. We would like to bound the expectation $\mathbb{E}_{\lambda_1,\lambda_2}[(\nu_1 - \lambda_1)^+]$. In order to do this we need further assumptions 4.10.4. The assumptions 4.10.4 are stronger than those in 4.10.3, and in fact the later follow from the former [Tartakovsky and Veeravalli, 2005]. We are now ready to state the bounding lemma.

Lemma 4.11.4. *Let the stopping time $\nu_1 = \nu(X, Z)$ be defined as in Equation (4.4.14). If Assumption 4.10.4, then as $\alpha \to 0$, for all $m \leq r$, and all events $\mathcal{A}$ not necessarily independent of ν_1:*

$$\begin{aligned}
\mathbb{E}_{k_1,k_2}[(\nu_1 - \lambda_1)^+]^m &\leq \left[\frac{|\log(\alpha)|}{q_1(X) + q_1(Z) + d_1}\right]^m (1 + o(1)) \\
\mathbb{E}_{\lambda_1,\lambda_2}[(\nu_1 - \lambda_1)^+]^m &\leq \left[\frac{|\log(\alpha)|}{q_1(X) + q_1(Z) + d_1}\right]^m (1 + o(1)) \\
\mathbb{E}_{k_1,k_2}[(\nu_1 - \lambda_1)^+ \mathbb{I}(\nu_1 \in \mathcal{A})]^m &\leq \left[\frac{|\log(\alpha)|}{q_1(X) + q_1(Z) + d_1}\right]^m \mathbb{P}_{k_1,k_2}(\nu_1 \in \mathcal{A})(1 + o(1)) \\
\mathbb{E}_{\lambda_1,\lambda_2}[(\nu_1 - \lambda_1)^+ \mathbb{I}(\nu_1 \in \mathcal{A})]^m &\leq \left[\frac{|\log(\alpha)|}{q_1(X) + q_1(Z) + d_1}\right]^m \mathbb{P}_{\lambda_1,\lambda_2}(\nu_1 \in \mathcal{A})(1 + o(1))
\end{aligned}$$

Proof. By definition of ν_1, since we are using the SRP statistic:

$$\begin{aligned}
\log(\Lambda_n(X, Z)) &\geq \log\left(\frac{\pi_1(k_1)}{\Pi_n^1}\right) + R_n^{k_1}(X) + R_n^{k_1}(Z) \\
&= S_n^{k_1}
\end{aligned}$$

We can define a stopping time:

$$\eta(k_1) = \inf\left\{n : S_n^{k_1} \geq \log(B_\alpha)\right\}$$

Notice that $\nu_1 - k_1 \leq \eta(k_1)$ on $\nu_1 \geq k_1$, as $\eta(k_1)$ starts at k_1 and the Shirayev statistics only includes values in range (k_1, n) after time k_1. Define:

$$\tilde{T}_\epsilon^{(k_1)} = \sup\left\{n \geq 1 : \left|\frac{1}{n - k_1 + 1} S_n^{k_1}(X) - q_1(X) + q_1(Z) + d_1\right| > \epsilon\right\}$$

Due to Assumption 4.10.4, and because $\frac{1}{n}\log\left(\frac{\pi_1(k_1)}{\Pi_n^1}\right) \to d_1$ as $n \to \infty$, we have $\mathbb{E}_{k_1,k_2}[\tilde{T}_\epsilon^{(k_1)}] < \infty$.

Furthermore, from the definition of η and setting $q_d = q_1(X) + q_1(Z) + d_1$:

$$\log(B_\alpha) \geq S_{\eta(k_1)-1}^{k_1} \geq (\eta(k_1) - k_1)(q_d - \epsilon) \text{ on } \left\{\eta(k) - 1 > \tilde{T}_\epsilon^{(k_1)}\right\}$$

We can bound for all $0 < \epsilon < q_d$:

$$\begin{aligned}\eta(k_1) &\leq k_1 + \frac{\log(B_\alpha)}{q_d - \epsilon}\mathbb{I}_{\left\{\eta(k)-1 > \tilde{T}_\epsilon^{(k_1)}\right\}} + (\tilde{T}_\epsilon^{(k_1)} + 1)\mathbb{I}_{\left\{\eta(k)-1 \leq \tilde{T}_\epsilon^{(k_1)}\right\}} \\ &\leq \tilde{T}_\epsilon^{(k_1)} + 1 + k_1 + \frac{\log(B_\alpha)}{q_d - \epsilon}\end{aligned}$$

So:

$$\frac{\nu_1 - k_1}{\log(B_\alpha)} \leq \frac{\tilde{T}_\epsilon^{(k_1)}}{\log(B_\alpha)} + \frac{1 + k_1}{\log(B_\alpha)} + \frac{1}{q_d - \epsilon}$$

Letting $\epsilon \to 0$, noticing $\mathbb{E}_{k_1,k_2}[\tilde{T}_\epsilon^{(k_1)}] < \infty$, and letting $\alpha \to 0$ ($\log(B_\alpha) \to \infty$) we obtain the first result in the Theorem for all $m \leq r$. Averaging over the priors, noticing $\mathbb{E}_{\lambda_1,\lambda_2}[\tilde{T}_\epsilon^{(k_1)}] < \infty$ we obtain the second. For the third and fourth results, it suffices to notice that:

$$\frac{(\nu_1 - k_1)\mathbb{I}(\nu_1 \in \mathcal{A})}{\log(B_\alpha)} \leq \frac{\tilde{T}_\epsilon^{(k_1)}}{\log(B_\alpha)} + \frac{1 + k_1}{\log(B_\alpha)} + \frac{\mathbb{I}(\nu_1 \in \mathcal{A})}{q_d - \epsilon}$$

Also, using Lemma 4.4.2 applied to the stopping time $\eta(k_1)$, which belongs to the class $\Delta_1(\alpha)$, it can be shown that as $\alpha \to 0$:

$$\begin{aligned}\mathbb{P}_{k_1,k_2}(\eta(k_1) \geq (1-\epsilon)L_1^\alpha) &= \mathbb{P}_{k_1,\infty}(\eta(k_1) \geq k_1 + (1-\epsilon)L_1^\alpha) \\ &= 1 - \gamma_{k_1,\infty}(\eta(k_1)) \\ &\to 1\end{aligned}$$

Then Chebyshev's inequality gives:

$$\begin{aligned}\mathbb{E}_{k_1,k_2}[\eta(k_1)^m] &\geq \left[\frac{(1-\epsilon)|\log\alpha|}{q_d}\right]^m \mathbb{P}_{k_1,k_2}(\eta(k_1) \geq (1-\epsilon)L_1^\alpha) \\ &\to \left[\frac{|\log\alpha|}{q_d}\right]^m (1 + o(1))\end{aligned}$$

Thus the lower and upper bound of $\eta(k_1)$ taken together show:

$$\lim_{\alpha\to 0} \frac{\mathbb{E}_{k_1,k_2}[\eta(k_1)^m]}{|\log\alpha|^m} = \left[\frac{1}{q_d}\right]^m$$

□

Proof. (of Theorem.) **(a)** Define:

$$\begin{aligned}q_1^d &= q_1(X) + q_1(Z) + d_1, \tilde{q}_1^d = q_1(X) + d_1, \\ \delta_\alpha(k_1, k_2) &= \mathbb{P}_{k_1,k_2}(\nu_1 > \nu_2), \mu_\alpha(k_1, k_2) = \mathbb{P}_{k_1,k_2}(\nu_1 > \tilde{\nu}_1)\end{aligned}$$

We start by analyzing the expectation of the stopping time, using the definition of $\bar{\nu}_1$

and $\bar{\nu_2}$:

$$\begin{aligned}\mathbb{E}_{k_1,k_2}\left[(\bar{\nu}_1-\lambda_1)^+\right] &= \mathbb{E}_{k_1,k_2}\left[(\nu_1-\lambda_1)^+\mathbb{I}(\nu_1\le\nu_2)\right]+\\ &+ \mathbb{E}_{k_1,k_2}\left[(\tilde{\nu}_1-\lambda_1)^+\mathbb{I}(\nu_1>\nu_2,\tilde{\nu}_1\ge\nu_2)\right]+\\ &+ \mathbb{E}_{k_1,k_2}\left[(\nu_2-\lambda_1)^+\mathbb{I}(\nu_1>\nu_2>\tilde{\nu}_1)\right]\end{aligned}$$

Each expectation can be bounded individually. The first expectation is bounded by using Lemma 4.11.4, setting $\mathcal{A}=\{\omega\in\Omega:\nu_1(\omega)\le\nu_2(\omega)\}$:

$$\begin{aligned}\mathbb{E}_{k_1,k_2}[(\nu_1-\lambda_1)^+\mathbb{I}(\nu_1\in\mathcal{A})]^m &\le \left[\frac{|\log(\alpha)|}{q_1(X)+q_1(Z)+d_1}\right]^m\mathbb{P}_{k_1,k_2}(\nu_1\le\nu_2)(1+o(1))\\ &= \left[\frac{|\log(\alpha)|}{q_1(X)+q_1(Z)+d_1}\right]^m(1-\delta_\alpha(k_1,k_2))(1+o(1))\end{aligned}$$

For the remainder of the section, let $\mathbb{E}$ denote $\mathbb{E}_{k_1,k_2}$ and $\mathbb{P}$ denote $\mathbb{P}_{k_1,k_2}$. We return to the usual notation wherever necessary. Also, we show the results for the case $m=1$ and the modifications for the case $m\le r$ are straightforward. The second expectation is bounded as:

$$\begin{aligned}\mathbb{E}\left[(\bar{\nu}_1-\lambda_1)^+\,\mathbb{I}(\nu_1>\nu_2,\tilde{\nu}_1\ge\nu_2)\right] &\le \mathbb{E}\left[(\tilde{\nu}_1-\lambda_1)^+\,\mathbb{I}(\nu_1>\nu_2)\right]\\ &= \mathbb{E}[(\tilde{\nu}_1-\lambda_1)^+]-\mathbb{E}_{k_1,k_2}[(\tilde{\nu}_1-\lambda_1)^+\,\mathbb{I}(\nu_1\le\nu_2)]\\ &\le \mathbb{E}[(\tilde{\nu}_1-\lambda_1)^+]-\frac{\log B_\alpha}{\tilde{q}_1^d}\mathbb{P}(\tilde{\nu}_1>\lambda_1+\frac{\log B_\alpha}{\tilde{q}_1^d},\nu_1\le\nu_2)\\ &\le \mathbb{E}[(\tilde{\nu}_1-\lambda_1)^+]-\frac{\log B_\alpha}{\tilde{q}_1^d}(\mathbb{P}(\tilde{\nu}_1>\lambda_1+\frac{\log B_\alpha}{\tilde{q}_1^d})-\mathbb{P}(\nu_1>\nu_2))\\ &= \mathbb{E}[(\tilde{\nu}_1-\lambda_1)^+]-\frac{\log B_\alpha}{\tilde{q}_1^d}(\mathbb{P}_{k_1,\infty}(\tilde{\nu}_1>\lambda_1+\frac{\log B_\alpha}{\tilde{q}_1^d})-\mathbb{P}(\nu_1>\nu_2))\\ &\le \mathbb{E}[(\tilde{\nu}_1-\lambda_1)^+]-\frac{\log B_\alpha}{\tilde{q}_1^d}(1-\tilde{\epsilon}_\alpha-\delta_\alpha(k_1,k_2))\\ &= \mathbb{E}_{k_1,\infty}[(\tilde{\nu}_1-\lambda_1)^+]-\frac{\log B_\alpha}{\tilde{q}_1^d}(1-\tilde{\epsilon}_\alpha-\delta_\alpha(k_1,k_2))\\ &\le \frac{\log B_\alpha}{\tilde{q}_1^d}(\tilde{\epsilon}_\alpha+\delta_\alpha(k_1,k_2)).\end{aligned}$$

Since (1) In third line we used $P(A\cap B)\ge P(A)-P(B^c)$; (2) In fifth line, $\tilde{\nu}_1$ does not depend on k_2; (3) $\mathbb{P}_{k_1,\infty}(\tilde{\nu}_1>\lambda_1+\frac{\log B_\alpha}{\tilde{q}_1^d})>1-\tilde{\epsilon}_\alpha$, by proof of Lemma 4.4.2 and (4) standard delay computed in Theorem 3 in Tartakovsky and Veeravalli [2005]. Finally,

$$
\begin{aligned}
\mathbb{E}\left[(\bar{\nu}_1-\lambda_1)^+\,\mathbb{I}(\nu_1>\nu_2,\tilde{\nu}_1<\nu_2)\right] &\le \mathbb{E}\left[(\nu_2-\lambda_1)^+\,\mathbb{I}(\nu_1>\nu_2>\tilde{\nu}_1)\right] \\
&\le \mathbb{E}[(\nu_1-\lambda_1)^+\,\mathbb{I}(\nu_1>\tilde{\nu}_1)] \\
&\le \frac{\log B_\alpha}{q_1^d}\mathbb{P}(\nu_1>\tilde{\nu}_1)(1+o(1)) \\
&= \frac{\log B_\alpha}{q_1^d}\mu_\alpha(k_1,k_2)(1+o(1)).
\end{aligned}
$$

Where we used (1) $\nu_1 > \nu_2$ in the second line and (2)in the third line, Lemma 4.11.4, setting $\mathcal{A} = \{\omega \in \Omega : \nu_1(\omega) \le \tilde{\nu}_1(\omega)\}$. In sum, we have:

$$
\begin{aligned}
\mathbb{E}[(\bar{\nu}_1-\lambda_1)^+] &\le \frac{\log B_\alpha}{q_1^d}(1-\delta_\alpha(k_1,k_2)+\mu_\alpha(k_1,k_2))(1+o(1)) + \frac{\log B_\alpha}{\tilde{q}_1^d}(\tilde{\epsilon}_\alpha+\delta_\alpha(k_1,k_2)) \\
&= \frac{\log B_\alpha}{q_1^d}(1-\delta_\alpha(k_1,k_2)+\mu_\alpha(k_1,k_2)+o(1)) + \frac{\log B_\alpha}{\tilde{q}_1^d}(\tilde{\epsilon}_\alpha+\delta_\alpha(k_1,k_2)).
\end{aligned}
$$

To obtain the delay, divide

$$
\mathbb{E}_{\lambda_1,\lambda_2}[(\bar{\nu}_1-\lambda_1)^+] \le \frac{\log B_\alpha}{q_1^d}(1-\delta_\alpha+\mu_\alpha+o(1)) + \frac{\log B_\alpha}{\tilde{q}_1^d}(\tilde{\epsilon}_\alpha+\delta_\alpha),
$$

by (using Lemma 4.11.1),

$$
\begin{aligned}
\mathbb{P}_{\lambda_1,\lambda_2}(\bar{\nu}_1 \ge \lambda_1) &\ge 1-\alpha\left(2+\frac{1}{L_-}\right) - \xi^\alpha_{\lambda_1,\lambda_2}(\bar{\nu}_1) \\
&\to 1-o(1)
\end{aligned}
$$

and we obtain the result in the Theorem since (1) $\tilde{\epsilon}_\alpha$ and μ_α are $o(1)$ (Lemmas 4.4.2 and 4.11.5) and (2) $\xi^\alpha_{\lambda_1,\lambda_2}(\bar{\nu}_1)$ is $o(1)$ as the procedure is regular.

We can now prove the matching lower bound for the delay.

(b) For the remainder of the proof, let $\mathbb{E}$ denote $\mathbb{E}_{k_1,k_2}$ and $\mathbb{P}$ denote $\mathbb{P}_{k_1,k_2}$. First notice that:

$$
\begin{aligned}
\mathbb{E}\left[(\bar{\nu}_1-\lambda_1)^+\right] &= \mathbb{E}\left[(\nu_1-\lambda_1)^+\mathbb{I}(\nu_1\le\nu_2)\right] + \mathbb{E}\left[(\max(\tilde{\nu}_1,\nu_2)-\lambda_1)^+\mathbb{I}(\nu_1>\nu_2)\right] \\
&\ge \mathbb{E}\left[(\nu_1-\lambda_1)^+\mathbb{I}(\nu_1\le\nu_2)\right] + \mathbb{E}\left[(\tilde{\nu}_1-\lambda_1)^+\mathbb{I}(\nu_1>\nu_2)\right]
\end{aligned}
$$

We can now bound the first term.

$$
\begin{aligned}
\mathbb{E}\left[(\nu_1-\lambda_1)^+\mathbb{I}(\nu_1\le\nu_2)\right] &\ge L_1^\alpha\ \mathbb{P}(\nu_1-k_1>(1-\epsilon)L_1^\alpha,\nu_1\le\nu_2)\\
&= L_1^\alpha\ [\mathbb{P}(\nu_1\ge k_1\wedge k_2,\nu_1\le\nu_2)-\mathbb{P}(k_2<\nu_1\le k_1,\nu_1\le\nu_2)-\\
&\quad+\mathbb{P}(k_1<\nu_1\le k_1+(1-\epsilon)L_1^\alpha,\nu_1\le\nu_2)]\\
&\ge L_1^\alpha\ [\mathbb{P}(\nu_1\ge k_1\wedge k_2,\nu_1\le\nu_2)-\xi^\alpha_{k_1,k_2}(\nu_1)-\gamma^{(k_1,k_2)}_{\epsilon,\alpha}(\nu_1)]\\
&\ge L_1^\alpha\ [\mathbb{P}(\nu_1\ge k_1\wedge k_2)-\mathbb{P}(\nu_1>\nu_2)-\xi^\alpha_{k_1,k_2}(\nu_1)-\gamma^{(k_1,k_2)}_{\epsilon,\alpha}(\nu_1)]\\
&= L_1^\alpha\ [1-\mathbb{P}_{\infty,\infty}(\nu_1\le k_1\wedge k_2)-\delta_\alpha(k_1,k_2)-\xi^\alpha_{k_1,k_2}(\nu_1)-\gamma^{(k_1,k_2)}_{\epsilon,\alpha}(\nu_1)]\\
&\ge L_1^\alpha\ \left[1-\frac{\alpha}{\Pi^1_{k_1\wedge k_2}}-\delta_\alpha(k_1,k_2)-\xi^\alpha_{k_1,k_2}(\nu_1)-\gamma^{(k_1,k_2)}_{\epsilon,\alpha}(\nu_1)\right]
\end{aligned}
$$

Where in the (1) fourth line we get a lower bound since the subtracted probabilities, and we identify them with previous definitions; (2) fifth line we use $P(A\cap B)\ge P(A)-P(B^c)$; (3)sixth line we use a change of measure and (4) seventh line we use Lemma 4.11.3.

So if the procedure is (k_1,k_2) regular, $\mathbb{E}\left[(\nu_1-\lambda_1)^+\mathbb{I}(\nu_1\le\nu_2)\right]\ge L_1^\alpha\ (1-\delta_\alpha(k_1,k_2)+o(1))$. For the averaged case over the priors, the last line above should be, using Lemma 4.11.1:

$$
\ge \quad L_1^\alpha\ [1-\alpha\left(1+\frac{1}{L_-}\right)-\delta_\alpha-\xi^\alpha_{\lambda_1,\lambda_2}(\nu_1)-\gamma_{\epsilon,\alpha}(\nu_1)]
$$

The second expectation can be bound similarly:

$$
\begin{aligned}
\mathbb{E}\left[(\tilde\nu_1-\lambda_1)^+\mathbb{I}(\nu_1>\nu_2)\right] &\ge \tilde L_1^\alpha\ \mathbb{P}(\tilde\nu_1-k_1\ge\tilde L_1^\alpha,\nu_1>\nu_2)\\
&\ge \tilde L_1^\alpha\ [\mathbb{P}(\tilde\nu_1-k_1\ge\tilde L_1^\alpha)-\mathbb{P}(\nu_1\le\nu_2)]\\
&= \tilde L_1^\alpha\ [1-\mathbb{P}(\tilde\nu_1<k_1)-\gamma^{(k_1,k_2)}_{\epsilon,\alpha}(\tilde\nu_1)-1+\delta_\alpha(k_1,k_2)]\\
&= \tilde L_1^\alpha\ [\delta_\alpha(k_1,k_2)+o(1)]
\end{aligned}
$$

Finally, we use the trivial upper bound $\mathbb{P}_{k_1,k_2}(\bar\nu_1\ge k_1)\le 1-o(1)$. Thus we get the result in the theorem. □

Lemma 4.11.5. *$\mu_\alpha(k_1,k_2)=o(1)$ and $\mu_\alpha=o(1)$*

Proof. First, we note that

$$
\begin{aligned}
\mathbb{P}_{k_1,k_2}(\nu_1>\tilde\nu_1) &\le \mathbb{P}_{k_1,k_2}(\nu_1>\tilde\nu_1,\tilde\nu_1\ge k_1+\tilde L_\alpha)+\mathbb{P}_{k_1,k_2}(\tilde\nu_1<k_1)+\\
&\quad+\mathbb{P}_{k_1,k_2}(k_1\le\tilde\nu_1\le k_1+\tilde L_\alpha),
\end{aligned}
$$

and asymptotically, in α, the last two terms are $o(1)$. Next, we follow along the lines of the first part of Lemma 4.4.1, to derive the result. Let $\mathbb{P}$ denote $\mathbb{P}_{k_1,k_2}$, $\mathcal{E}(X)=$

$\{\log \Lambda_l(X) \geq \log B_\alpha\}$ and $\mathcal{I}(\tilde{L}_\alpha) = [k_1 + \tilde{L}_\alpha, \infty)$:

$$\begin{aligned}
\mathbb{P}_{k_1,k_2}(\nu_1 > \tilde{\nu}_1, \tilde{\nu}_1 \geq k_1 + \tilde{L}_\alpha) &\leq \sum_{l=k_1+\tilde{L}_\alpha}^{\infty} \mathbb{P}\left(\{\log \Lambda_l(X,Z) \leq \log B_\alpha\} \cap \mathcal{E}(X), \tilde{\nu}_1 = l\right), \\
&\leq \sum_{l=k_1+\tilde{L}_\alpha}^{\infty} \mathbb{P}\left(\left\{\log \Lambda_l(X) + \min_{s\in[1,l]} R_l^s(Z) - \log \Pi_1(l) \leq \log B_\alpha\right\} \cap \mathcal{E}(X), \tilde{\nu}_1 = l\right), \\
&\leq \sum_{l=k_1+\tilde{L}_\alpha}^{\infty} \mathbb{P}\left(\max_{s\in[1,l]} -R_l^s(Z) \geq -\log \Pi_1(l), \tilde{\nu}_1 = l\right), \\
&\leq \sum_{l=k_1+\tilde{L}_\alpha}^{\infty} \mathbb{P}(\tilde{\nu}_1 = l)\mathbb{P}\left(\max_{s\in[1,l]} -R_l^s(Z) \geq -\log \Pi_1(l)\right), \\
&\leq \max_{l\in\mathcal{I}(\tilde{L}_\alpha)} \mathbb{P}\left(\max_{s\in[1,l]} -R_l^s(Z) \geq -\log \Pi_1(l)\right), \\
&\leq \max_{l\in\mathcal{I}(\tilde{L}_\alpha)} l \max_{s\in[1,l]} \mathbb{P}\left(-R_l^s(Z) \geq -\log \Pi_1(l)\right), \\
&\leq \max_{l\in\mathcal{I}(\tilde{L}_\alpha)} l \max_{r\in[1,l]} \exp\left\{-\frac{(V_l + l\, d_1 - \min(r,k_1)\, q_0(Z) + [l - \max(r,k_1) + 1]_+ q_1(Z))^2}{l \max(\sigma_0^2(Z), \sigma_1^2(Z))}\right\},
\end{aligned}$$

where $V_l = -\log \Pi_1(l) - l\, d_1$. Note that for $l > L$ for some L, $|V_l| < \epsilon$ due to assumption 4.10.1. Thus when $r \leq k_1$, the maximum happens at $r = k_1$, with rate upper bounded by

$$r(l) = \left\{\frac{(\epsilon + l\, d_1 - k_1\, q_0(Z) + (l - k_1 + 1) q_1(Z))^2}{l \max(\sigma_0^2(Z), \sigma_1^2(Z))}\right\}.$$

Else, the maximum happens at $r = l$, with rate upper bounded by

$$r(l) = \left\{\frac{(\epsilon + l\, d_1 + q_1(Z))^2}{l \max(\sigma_0^2(Z), \sigma_1^2(Z))}\right\}.$$

In both cases, for any $l \in [k_1 + \tilde{L}_\alpha, \infty]$, $r(l) \to \infty$ as $\alpha \to 0$. Thus we obtain $\mathbb{P}_{k_1,k_2}(\nu_1 > \tilde{\nu}_1, \tilde{\nu}_1 \geq k_1 + \tilde{L}_\alpha) = o(1)$. Since k_1 only appears multiplying an exponentially small probability, as both rates go to infinity uniformly over k_1, we can apply expectation to both sides, and obtain that $\mathbb{P}_{\lambda_1,\lambda_2}(\nu_1 > \tilde{\nu}_1) = o(1)$, as $\mathbb{E}[k_1] = \lambda_1 < \infty$. $\square$

Chapter 5

Simultaneous Placement and Scheduling of Sensors

5.1 Introduction

When monitoring spatial phenomena, such as speeds on a highway, deciding where to *place* a small number of sensors to obtain best prediction accuracy is an important task. The Sensys Networks wireless traffic sensor [Haoui et al., 2008a], discussed in Chapter 2, provides 30 second aggregate speed, flow and vehicle density measurements. Currently the system is being deployed by Caltrans at different sites in California, including highways and arterial roads. When using such wireless sensor networks, power consumption is a key constraint, since every measurement drains the battery. For applications such as road speed monitoring, a minimum battery lifetime is required to ensure feasibility of the sensor network deployment. One approach to meeting such lifetime requirements is to deploy few nodes with large batteries. However, such an approach can be sensitive to node failures. Additionally, packaging constraints can limit the size of the battery deployed with the nodes. For these and other reasons, it can be more effective to deploy a larger number of nodes with smaller batteries, that are activated only a fraction of the time. Hence, to improve the lifetime of such a sensor network, the problem of *scheduling* is of crucial importance: Given a fixed placement of sensors, when should we turn each sensor on for obtaining high monitoring performance over all time steps? One approach that has been found effective in the past is to partition the sensors into k groups [Abrams et al., 2004; Deshpande et al., 2008; Koushanfar et al., 2006]. By activating a different group of sensors at each time step and cyclicly shifting through these groups, the network lifetime can effectively be increased by a factor of k. In the traffic network application, current studies indicate that an increase by a factor of $k = 4$ would be required to make sensor deployment an economically feasible option (see Section 5.7 for more details).

Traditionally, sensor placement and sensor scheduling have been considered separately – one first decides where to place the sensors, and then when to activate them. In this Chapter, we present an efficient algorithm, eSPASS (for *efficient Simultaneous Placement* *and Scheduling of Sensors*), that jointly optimizes the sensor placement and the sensor schedule. We prove that our algorithm provides a constant factor approximation to the optimal solution of this NP-hard optimization problem.

Most existing approaches to sensor placement and scheduling associate a fixed sensing region with every sensor, and then attempt to maximize the number of regions covered in every group of sensors (*c.f.*, Abrams et al. [2004]; Deshpande et al. [2008]; Hochbaum and Maas [1985]). In complex applications such as traffic or environmental monitoring however, the goal of sensor placement is a prediction problem, where one intends to predict the sensed phenomenon at the locations where no sensors are placed. Our algorithm applies to such settings where the sensing quality of a set of sensors is measured, e.g., in the improvement of prediction accuracy (more formally, our algorithm applies whenever the sensing quality function satisfies *submodularity*, an intuitive diminishing returns property).

In contrast to most existing algorithms that optimize scheduling for average case performance, our approach provides a schedule that performs uniformly well over time, hence leading to a well-balanced performance of the sensor network. For security-critical applications such as outbreak detection, such balanced performance is a crucial requirement not met by existing algorithms. In fact, our experimental results show that average-case optimal solutions can lead to arbitrarily unbalanced performance, but optimizing for balanced performance (using eSPASS) typically leads to good average-case performance.

Deploying a large number of scheduled sensors has the additional benefit that it allows trading off power for accuracy. The deployed network might have several modes of operation: a scheduled mode of operation, where only a small fraction of sensors is turned on, and a "high density" mode where all (or a larger fraction of) sensors are activated. For example, in traffic monitoring, once a traffic congestion is detected (during scheduled mode), the high density mode could be used to accurately identify the boundary of the congestion. We show how our algorithm can be extended to support such a power-accuracy tradeoff.

We present extensive empirical studies on several applications, illustrating the versatility of our algorithm. The main application is traffic monitoring using both fixed and mobile sensors. Our results show that simultaneous placing and scheduling results in drastically improved performance compared to the setting where optimization over the placement and the scheduling are performed separately.

In summary, our main contributions are:

- We study the problem of simultaneously placing and scheduling sensors as a novel optimization problem.
- We develop eSPASS, an efficient approximation algorithm for this problem, that applies to a variety of realistic sensing quality functions (such as area coverage, variance reduction, outbreak detection, etc.). Our algorithm is guaranteed to provide a near-optimal solution, that obtains at least a constant fraction of the optimal sensing quality. eSPASS furthermore allows a trade off between power consumption and accuracy.
- We perform several extensive case studies on real sensing problems in traffic monitoring, demonstrating the effectiveness of our approach.

5.2 Related Work

In the context of wireless sensor networks, where sensor nodes have limited battery life and can hence only enable a small number of measurements, optimally placing and scheduling sensors is of key importance.

Sensor Placement Many approaches for optimizing sensor placements assume that sensors have a fixed region [Hochbaum and Maas, 1985; Gonzalez-Banos and Latombe, 2001; Bai et al., 2006]. These regions are usually convex or even circular. Furthermore, it is assumed that everything within this region can be perfectly observed, and everything outside cannot be measured by the sensors. For complex applications such as traffic monitoring however, such assumptions are unrealistic, and the direct optimization of prediction accuracy is desired. The problem of selecting observations for monitoring spatial phenomena has been investigated extensively in geostatistics (*c.f.*, Cressie [1991] for an overview), and more generally (Bayesian) experimental design (*c.f.*, Chaloner and Verdinelli [1995]). Submodularity has been used to analyze algorithms for placing a fixed set of sensors [Krause et al., 2007]. These approaches however only consider the sensor placement problem, and not the scheduling aspect.

Sensor Scheduling The problem of deciding when to selectively turn on sensors in order to conserve power was first discussed by Slijepcevic and Potkonjak [2001] and Zhao et al. [2002]. Typically, it is assumed that sensors are associated with a fixed sensing region, and a spatial domain needs to be covered by the regions associated with the selected sensors. Abrams et al. [2004] presents an efficient approximation algorithm with theoretical guarantees for this problem. Deshpande et al. [2008] presents an approach for this problem based on semidefinite programming (SDP), handling more general constraints and providing tighter approximations. They also provide a randomized rounding based approach for scheduling under the balanced objective (which they call min-coverage (time)). However, in contrast to eSPASS (when specialized to scheduling) their algorithm requires to relax the constraint that each sensor location can only be selected once. Also, their guarantee only holds with high probability, whereas eSPASS is deterministic. The approaches described above do not apply to the problem of optimizing sensor schedules for more complex sensing quality functions such as, e.g., the increase in prediction accuracy and other sensing quality functions considered in this Chapter. To address these shortcomings, Koushanfar et al. [2006] developed an approach for sensor scheduling that guarantees a specified prediction accuracy based on a regression model. However, their approach relies on the solution of a Mixed Integer Program, which is intractable in general. Zhao et al. [2002] proposed heuristics for selectively querying nodes in a sensor network in order to reduce the entropy of the prediction. Unlike the algorithms presented in this Chapter, their approaches do not have any performance guarantees.

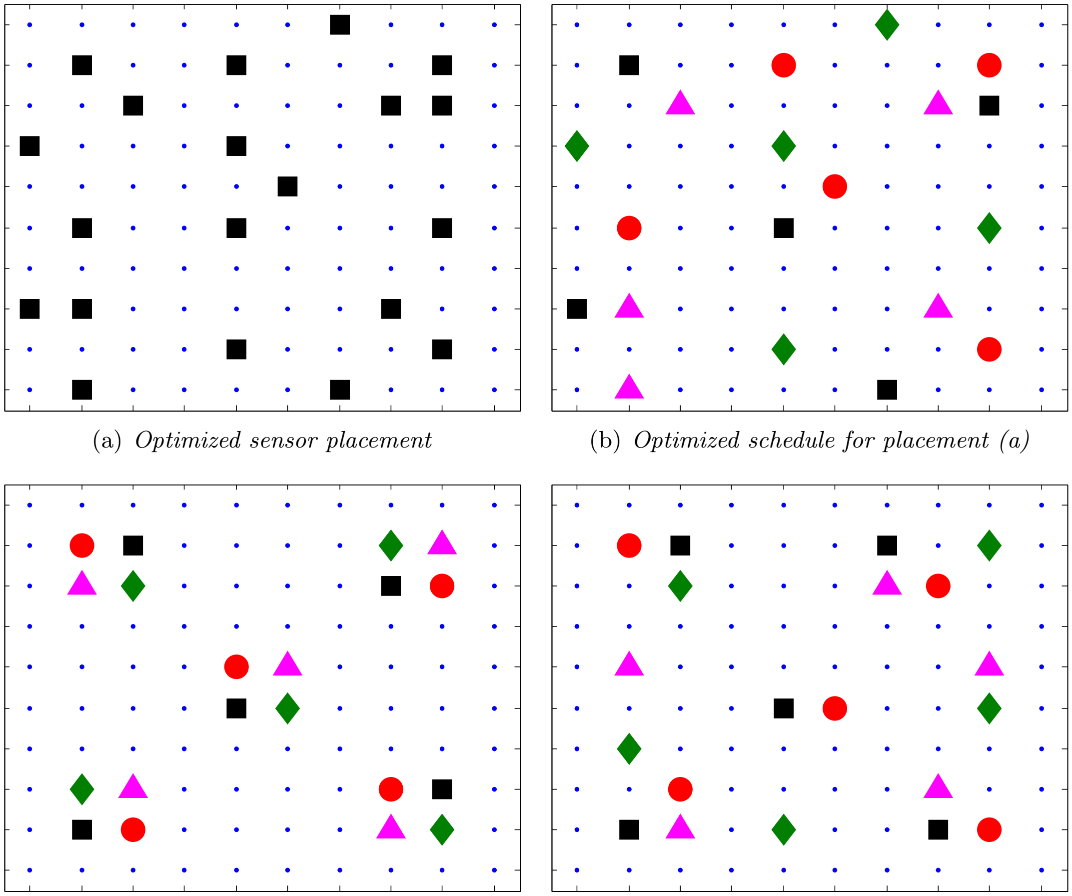
Submodular optimization The problem of maximizing a submodular function subject to a matroid constraint has been studied by Fisher et al. [1978], who proved that the greedy algorithm gives a factor 2 approximation. Recently, Vondrak [2008] showed that a more complex algorithm achieves a (1-1/e) approximation to this problem. Note that this algorithm could be applied to the Problem (5.3.1) instead of GAPS. Furthermore note that using this algorithm as a subroutine, the analysis of eSPASS can be improved to give a $\frac{e-1}{3e} \geq \frac{1}{5}$ guarantee. A related version of optimization Problem (5.3.2), where for each time step t a different function F_t is used has been studied in the context of combinatorial allocation problems. Ponnuswami and Khot [2007] present an algorithm that guarantees a $1/(2k-1)$ approximation. For the special case where the objective functions F_t are additive (modular), Asadpour and Saberi [2007] developed an algorithm that guarantees an improved $\Omega\left(\frac{1}{\sqrt{k}\log^3 k}\right)$ approximation. Both algorithms however only apply for the scheduling setting (i.e., they require that $m = |\mathcal{V}|$). Furthermore, note that the approximation performance of these algorithms very quickly decreases with k, in contrast to our eSPASS approach that provides an approximation guarantee that is independent of the number of time steps. Krause et al. [2008c] consider the problem of robust maximization of submodular functions: Given a collection of submodular functions, $F_1, \dots, F_m$, they want to find a set $|\mathcal{A}| \leq k$ that maximizes $\min_i F_i(\mathcal{A})$. While this problem appears related to the SPASS problem, where we want to maximize $\min_i F(\mathcal{A}_i)$, the solution techniques and results are very different. Firstly, there is a strong conceptual difference: In robust submodular optimization, a single set $\mathcal{A}$ is sought that maximizes multiple functions $F_1, \dots, F_m$, whereas in SPASS, a collection of sets $\mathcal{A}_1, \dots, \mathcal{A}_k$ is sought that each perform well with respect to a single function F. Hence, the two problem formulations address very different optimization tasks. Second, while both algorithms exploit the fact that truncation preserves submodularity, each contains unique algorithmic elements. Lastly, the performance guarantees vary drastically: the robust submodular optimization problem does not admit any approximation (and requires the relaxation of the constraint that $|\mathcal{A}| \leq k$), whereas for the SPASS problem, eSPASS obtains a constant-factor 6 approximation.

5.3 Problem Statement

We will first separately introduce the sensor placement and scheduling problems, and then formalize the problem of simultaneously placing and scheduling sensors.

5.3.1 Sensor Placement

In *sensor placement*, we are given a finite set $\mathcal{V}$ of possible locations where sensors can be placed. Our goal is to select a small subset $\mathcal{A} \subseteq \mathcal{V}$ of locations to place sensors at, that maximizes a sensing quality function $F(\mathcal{A})$. There are several different notions of sensing quality that we might want to optimize, each depending on the particular sensing task. For example, we can associate sensing regions with every sensor, and $F(\mathcal{A})$ can measure the total area covered when placing sensors at locations $\mathcal{A}$. In complex applications such as the traffic monitoring problem, we are interested in optimizing the prediction accuracy when obtaining measurements from locations $\mathcal{A}$. In this setting, we can model the state of the world (e.g., the traffic condition at different locations) using a collection of random variables $\mathcal{X}_{\mathcal{V}}$, one variable $\mathcal{X}_s$ for each location $s \in \mathcal{V}$. We can then use a probabilistic model (such as a Gaussian Process which is frequently used in geostatistics, *c.f.*, Cressie [1991]) that



(a) *Optimized sensor placement* (b) *Optimized schedule for placement (a)*

(c) *Simultaneous placement and schedule* (d) *Multicriterion solution*

Figure 5.1. In the stage-wise approach, sensors are first deployed (a), and the deployed sensors are then scheduled (b, sensors assigned to the same time slot are drawn using the same color and marker). In the simultaneous approach, we jointly optimize over placement and schedule (c). (d) Multicriterion solution to Problem (5.6.1) ($\lambda = .25$) that performs well both in scheduled and high-density mode.

models a joint probability distribution $P(\mathcal{X}_\mathcal{V})$ over the possible locations. Upon acquiring measurements $\mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A}$ at a subset of locations $\mathcal{A}$, we can then predict the phenomenon at the unobserved locations using the conditional distribution $P(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A})$. We can then use the expected mean squared error,

$$\mathrm{Var}(\mathcal{X}_\mathcal{V} \mid \mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A}) = \frac{1}{|\mathcal{V}|} \sum_{s \in \mathcal{V}} \mathbb{E}\left[(\mathcal{X}_s - \mathbb{E}[\mathcal{X}_s \mid \mathbf{x}_\mathcal{A}])^2 \mid \mathbf{x}_\mathcal{A}\right]$$

to quantify the uncertainty in this prediction. Since we do not know the values $\mathbf{x}_\mathcal{A}$ before placing the sensors, a natural choice of the sensing quality function $F(\mathcal{A})$ is to measure the *expected reduction in variance* at the unobserved locations,

$$F(\mathcal{A}) = \mathrm{Var}(\mathcal{X}_\mathcal{V}) - \int P(\mathbf{x}_\mathcal{A}) \mathrm{Var}(\mathcal{X}_\mathcal{V} \mid \mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A}) d\mathbf{x}_\mathcal{A}.$$

This sensing quality function has been found useful for sensor selection (*c.f.*, Deshpande et al. [2004]; Krause et al. [2008a]) and experimental design (*c.f.*, Chaloner and Verdinelli [1995]).

It can be shown that both the area covered and the variance reduction objective, as well as many other notions of sensing quality, satisfy the following intuitive diminishing returns property [Das and Kempe, 2008; Krause et al., 2007][1]: Adding a sensor helps more if we have placed few sensors so far, and less if we already have placed lots of sensors. This intuition can be formalized using the combinatorial concept of *submodularity*: A set function F is called submodular, if for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $s \in \mathcal{V} \setminus \mathcal{B}$

$$F(\mathcal{A} \cup \{s\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{s\}) - F(\mathcal{B}),$$

i.e., adding s to a small set $\mathcal{A}$ helps more than adding s to the superset $\mathcal{B}$. In addition, these sensing quality functions are *monotonic*: For all $\mathcal{A} \subseteq \mathcal{B}$ it holds that $F(\mathcal{A}) \leq F(\mathcal{B})$, i.e., adding more sensors can only improve the sensing quality.

Based on this notion of a monotonic, submodular sensing quality function, the sensor placement problem then is

$$\max_{\mathcal{A}} F(\mathcal{A}) \text{ such that } |\mathcal{A}| \leq m,$$

i.e., we want to find a set $\mathcal{A}$ of at most m locations to place sensors maximizing the sensing quality F.

5.3.2 Sensor Scheduling

In *sensor scheduling*, we are given a sensor placement (i.e., locations $\mathcal{A}$), and our goal is to assign each sensor $s \in \mathcal{A}$ one of k time slots. This assignment partitions the set $\mathcal{A}$ into disjoint sets $\mathcal{A}_1, \dots, \mathcal{A}_k$, where $\mathcal{A}_t \subseteq \mathcal{A}$ is the subset of sensors that have been assigned slot t. A round-robin schedule can then be applied that cycles through the time slots, and activates sensors $\mathcal{A}_t$ at time t. Since each sensor is active at only one out of k time slots, this procedure effectively increases the lifetime of the network by a factor of k. How can we quantify the value of a schedule $\mathcal{A} = (\mathcal{A}_1, \dots, \mathcal{A}_k)$? For each group $\mathcal{A}_t$, we can compute the sensing quality $F(\mathcal{A}_t)$[2]. One possibility would then be to optimize for the *average* performance over time,

$$\max_{\mathcal{A}_1,\dots,\mathcal{A}_k} \frac{1}{k} \sum_{t=1}^{k} F(\mathcal{A}_t).$$

[1] Variance reduction has been shown to be submodular for Gaussian distributions under certain assumptions about the covariance by Das and Kempe [2008].

[2] Note that we assume the same sensing quality function F for each time step. This assumption has been made in the past (*c.f.*, Koushanfar et al. [2006]; Abrams et al. [2004]), and is reasonable for many monitoring tasks.

However, as we show in our experiments, if we optimize for the average case performance, it can happen that a few of the time slots are very poorly covered, i.e., there is a time t such that $F(\mathcal{A}_t)$ is very low. For security-critical applications, this can be problematic. Instead, we can also optimize for a *balanced* schedule,

$$\max_{\mathcal{A}_1,\dots,\mathcal{A}_k} \min_t F(\mathcal{A}_t),$$

that performs uniformly well over time.

Note that the above formulation of the scheduling problem allows to handle settings where each sensor can be active at $r \geq 1$ timesteps. In this setting, we simply define a new ground set $\mathcal{A}' = \mathcal{A} \times \{1, \dots, r\}$ where the pair $(s, i) \in \mathcal{A}'$ refers to the i-th activation of sensor s. The sensing quality function is modified as $F'(\mathcal{A}'_j) = F(\{s : \exists_i (s, i) \in \mathcal{A}'_j\})$.

5.3.3 Simultaneous placement and scheduling

Both sensor placement and sensor scheduling have been studied separately from each other in the past. One approach towards placement and scheduling would be to first use an algorithm (such as the algorithm proposed by Krause et al. [2007]) to find a sensor placement $\mathcal{A}$, and then use a separate algorithm (such as the mixed integer approach of Koushanfar et al. [2006]) to find a schedule $\mathcal{A}_1, \dots, \mathcal{A}_k$. We call this approach a *stage-wise* approach, and illustrate it in Figures 5.1(a) and 5.1(b).

Instead of separating placement and scheduling, we can *simultaneously* optimize for the placement and the schedule. Suppose we have resources to purchase m sensors, and we would like to extend the network lifetime by a factor of k. Our goal would then be to find k disjoint sets $\mathcal{A}_1, \dots, \mathcal{A}_k \subseteq \mathcal{V}$, such that together these sets contain at most m locations, i.e., $|\bigcup_t \mathcal{A}_t| \leq m$. We call this problem the SPASS problem, for *simultaneous placement and scheduling of sensors*. Again, we can consider the average-case performance,

$$\max_{\mathcal{A}_1\dots\mathcal{A}_k} \frac{1}{k}\sum_{t=1}^{k} F(\mathcal{A}_t) \text{ s.t. } \mathcal{A}_i \cap \mathcal{A}_j = \emptyset \text{ if } i \neq j \text{ and } |\bigcup_t \mathcal{A}_t| \leq m, \tag{5.3.1}$$

and the balanced objective,

$$\max_{\mathcal{A}_1\dots\mathcal{A}_k} \min_t F(\mathcal{A}_t) \text{ s.t. } \mathcal{A}_i \cap \mathcal{A}_j = \emptyset \text{ if } i \neq j \text{ and } |\bigcup_t \mathcal{A}_t| \leq m. \tag{5.3.2}$$

By performing this simultaneous optimization, we can obtain very different solutions, as illustrated in Figure 5.1(c). In Section 5.7, we will show that this simultaneous approach can lead to drastically improved performance as compared to the traditional, stage-wise approach. In this Chapter, we present eSPASS, an efficient approximation algorithm with strong theoretical guarantees for this problem.

The placement and schedule in Figure 5.1(c) has the property that the sensors selected at each time step share very similar locations, and hence perform roughly equally well. However, if activated all at the same time, the "high-density" performance $F(\mathcal{A}_1 \cup \dots \cup \mathcal{A}_k)$ is much lower than that of the placement in Figure 5.1(a). We also develop an algorithm, mcSPASS, that leads to placements which perform well both in scheduled and in high-density mode. Figure 5.1(d) presents the solution obtained for the mcSPASS algorithm.

Note that instead of fixing the number of time slots, we could also specify a desired accuracy constraint Q and then ask for the maximum lifetime solution, i.e., the largest number k of time slots such that a solution with minimum (or average) sensing quality Q is obtained. Clearly, an algorithm that solves Problem (5.3.2) (or Problem (5.3.1)) could be used to solve this alternative problem, by simply binary searching over possible values for k[3].

Further note that it is possible to allow each sensor to be active at $r \geq 1$ timesteps by using the modification described in Section 5.3.2.

5.4 A naive greedy algorithm

We will first study the problem of optimizing the average performance over time, i.e., Problem (5.3.1), for a fixed monotonic submodular sensing quality function F. Considering the fact that simultaneously placing and scheduling is a strict generalization of sensor placement, which itself is NP-hard (*c.f.*, Krause et al. [2007]), we cannot expect to efficiently find the optimal solution to Problem (5.3.1) in general.

Instead, we will use the following intuitive greedy algorithm that we call GAPS for *Greedy Average-case Placement and Scheduling.* At every round, GAPS picks a time slot t and location s which increases the total sensing quality the most, until m location/time-slot pairs have been picked. It is formalized as Algorithm 1.

Algorithm GAPS (F, $\mathcal{V}$, k, m)
$\mathcal{A}_t \leftarrow \emptyset$ for all t;
for $i = 1$ **to** m **do**
  foreach $s \in \mathcal{V} \setminus (\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k)$, $1 \leq t \leq k$ **do**
    $\delta_{t,s} \leftarrow F(\mathcal{A}_t \cup \{s\}) - F(\mathcal{A}_t)$;
  $(t^*, s^*) \leftarrow \operatorname{argmax}_{t,s} \delta_{t,s}$;
  $\mathcal{A}_{t^*} \leftarrow \mathcal{A}_{t^*} \cup \{s^*\}$;

Algorithm 1: The greedy average-case placement and scheduling (GAPS) algorithm.

5.4.1 Theoretical guarantee

Perhaps surprisingly, we can show that this simple algorithm provides near-optimal solutions for Problem (5.3.1). In fact, it generalizes the distributed Set-k Cover algorithm proposed by Abrams et al. [2004] to arbitrary submodular sensing quality functions F, and to the setting where at most m sensors can be selected in total.

[3]However, in case an approximate algorithm is used such as the eSPASS algorithm developed in this Chapter, its guarantees are not necessarily preserved.

Theorem 5.4.1. *For any monotonic and submodular function F,* GAPS *returns a solution* $\mathcal{A}_1, \ldots, \mathcal{A}_k$ *s.t.*

$$\frac{1}{k}\sum_t F(\mathcal{A}_t) \geq \frac{1}{2}\max_{\mathcal{A}'}\frac{1}{k}\sum_t F(\mathcal{A}'_t).$$

GAPS *requires at most* $\mathcal{O}(kmn)$ *evaluations of* F*.*

The proofs of Lemma 5.5.2 and all other results are given in the Appendix. The key observation is that Problem (5.3.1) is an instance of maximizing a submodular function subject to a *matroid* constraint (*c.f.*, the Appendix for details). A fundamental result by Fisher et al. [1978] then proves that the greedy algorithm returns a solution that obtains at least one half of the optimal average-case score. Matroids for sensor scheduling have been considered before by Williams et al. [2007].

5.4.2 Greedy can lead to unbalanced solutions

If a sensor placement and schedule is sought that performs well "on-average" over time, GAPS performs well. However, even though the average performance over time, $\frac{1}{k}\sum_t F(\mathcal{A}_t)$, is high, the performance at some individual timesteps t' can be very poor, and hence the schedule can be *unfair*. In security-critical applications, where high performance is required at all times, this behavior can be problematic. In such settings, we might be interested in optimizing the *balanced* performance over time, $\min_t F(\mathcal{A}_t)$. This optimization task was raised as an open problem by [Abrams et al., 2004].

A first idea would be to try to modify the GAPS algorithm to directly optimize this balanced performance, i.e., replace Line 1 in Algorithm 1 by

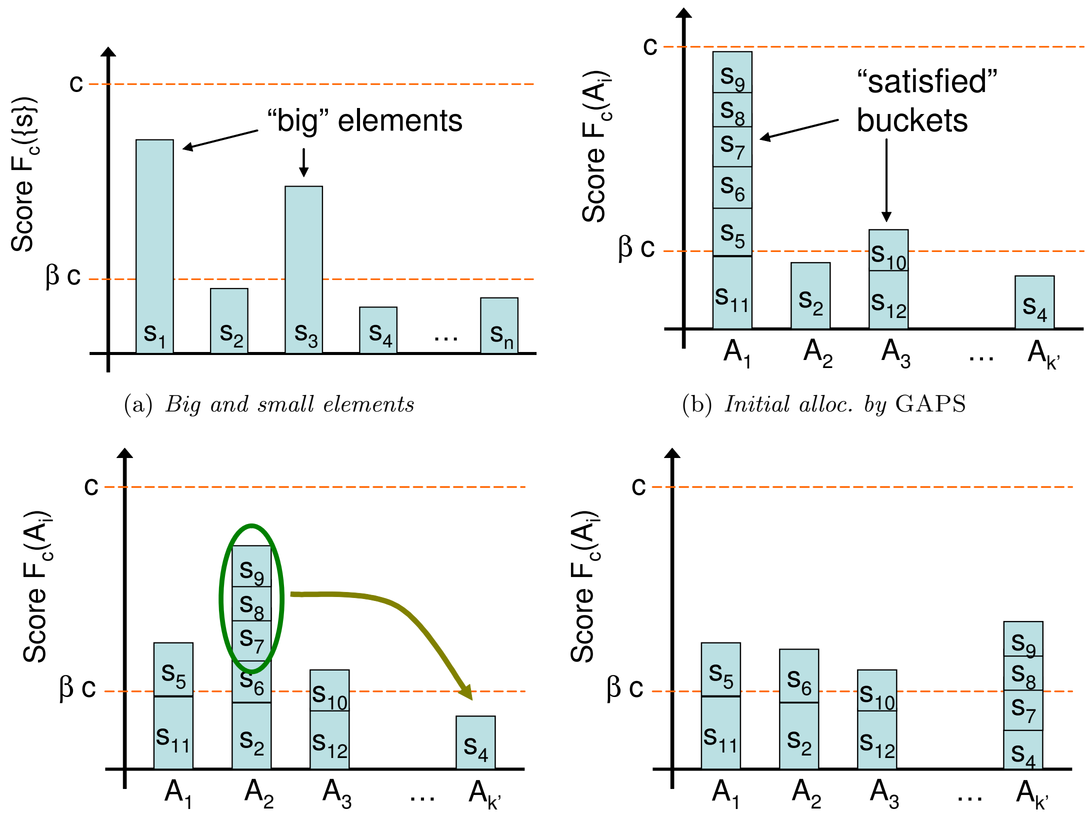
$$\delta_{t,s} \leftarrow \min_j F(\mathcal{A}_j^{+(t,s)}) - \min_j F(\mathcal{A}_j),$$

where $\mathcal{A}^{+(t,s)}$ is the solution obtained by adding location s to time slot $\mathcal{A}_t$ in solution $\mathcal{A} = \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k$. We call this modified algorithm the GBPS algorithm (for Greedy Balanced Placement and Scheduling). Unfortunately, both GAPS and GBPS can perform arbitrarily badly. Consider a simple scenario with three locations, $\mathcal{V} = \{a, b, c\}$, and the monotonic submodular function $F(\mathcal{A}) = |\mathcal{A}|$. We want to partition $\mathcal{V}$ into three timesteps, i.e., $k = 3$ and $m = 3$. Here, the optimal solution would be to pick $\mathcal{A}_1^* = \{a\}$, $\mathcal{A}_2^* = \{b\}$ and $\mathcal{A}_3^* = \{c\}$. However, both GAPS and GBPS would (ties broken unfavorably) pick $\mathcal{A}_1 = \{a, b, c\}$ and $\mathcal{A}_2 = \mathcal{A}_3 = \emptyset$, obtaining a minimum score of 0.

Unfortunately, this poor performance is not just a theoretical example – in Section 5.7 we demonstrate it empirically on real sensing tasks.

5.5 The eSPASS algorithm

In the following, we will develop an efficient algorithm, eSPASS (for *efficient Simultaneous Placement and Scheduling of Sensors*), that, as we will show in Section 5.5.2, is guaranteed to provide a near-optimal solution to the Problem (5.3.2). To the best of our



(a) *Big and small elements*

(b) *Initial alloc. by* GAPS

(c) *Reallocation move*

(d) *Reallocated elements*

Figure 5.2. Illustration of our eSPASS algorithm. The algorithm first "guesses" (binary searches for) the optimal value c. (a) Then, big elements s where $F(\{s\}) \geq \beta c$ are allocated to separate buckets. (b) Next, the remaining small elements are allocated to empty buckets using the GAPS algorithm. (c,d) Finally, elements are reallocated until all buckets are satisfied.

knowledge, our algorithm is the first with theoretical guarantees for this general problem, hence partly resolving the open problem described by Abrams et al. [2004].

5.5.1 Algorithm overview

We start with an outline of our algorithm, and then proceed to discuss each step more formally.

Our high-level goal will be to reduce the problem of optimizing the balanced objective into a sequence of modified optimization problems involving an average-case objective, which we can approximately solve using GAPS. This idea is based on the following intuition: Consider a *truncated* objective function $F_c(\mathcal{A}) = \min\{F(\mathcal{A}), c\}$. The key observation[4] is

[4] Krause et al. [2008c] used this observation to develop an algorithm for robust optimization of submodular functions.

that, for *any* constant c, it holds that

$$\min_t F(\mathcal{A}_t) \geq c \Leftrightarrow \frac{1}{k}\sum_{t=1}^{k} F_c(\mathcal{A}_t) = c,$$

i.e., the minimum score is greater than or equal to c if and only if the average truncated score is c.

Now suppose someone tells us the value c^* attained by an optimal solution:

$$\max_{\mathcal{A}} \min_t F(\mathcal{A}_t) = c^*.$$

By the above observation this problem is equivalent to solving

$$\max_{\mathcal{A}} \frac{1}{k}\sum_{t=1}^{k} F_{c^*}(\mathcal{A}_t). \tag{5.5.1}$$

It can be shown (*c.f.*, Fujito [2000]) that for $c \geq 0$, the truncated objective function F_c remains monotonic and submodular. Hence, Problem (5.5.1) is an instance of the *average-case* Problem (5.3.1). Now we face the challenge that we do *not* generally know the optimal value c^*. We could use a simple binary search procedure to find this optimal value. Hence, if we could *optimally* solve the monotonic submodular *average-case* Problem (5.3.1), we would obtain an optimal solution to the *balanced* Problem (5.3.2).

Unfortunately, as shown in Section 5.4.1, solving the average-case problem is NP-hard, and, using GAPS, we can only solve it *approximately*, obtaining a solution that achieves at least half of the optimal value. In the following, we will show how we can turn this approximate solution for the average-case problem into a near-optimal solution for the balanced problem.

Our algorithm will maintain one "bucket" $\mathcal{A}_t \subseteq \mathcal{V}$ for each time slot t. Since our goal is to develop an approximation algorithm achieving at least a fraction $\beta > 0$ of the optimal sensing quality, we need to allocate m elements $s \in \mathcal{V}$ to the k buckets such that $F(\mathcal{A}_t) \geq \beta c^*$ for all buckets $\mathcal{A}_t$. Here, β is a constant that we will specify later. We call a bucket "satisfied" if $F(\mathcal{A}_t) \geq \beta c^*$, "unsatisfied" otherwise. Here is an outline of our eSPASS algorithm, Figure 5.2 presents an illustration.

1. "Guess" the optimal value c.

2. Call an element $s \in \mathcal{V}$ "big" if $F_c(\{s\}) \geq \beta c$ and "small" otherwise. Put each big element into a separate bucket (*c.f.*, Figure 5.2(a)). From now on, we ignore those satisfied buckets, and focus on the unsatisfied buckets.

3. Run GAPS to optimize F_c and allocate the small elements to the unsatisfied buckets (*c.f.*, Figure 5.2(b)).

4. Pick a "satisfied" bucket $\mathcal{A}_t$ that contains sufficiently many elements, and reallocate enough elements to an "unsatisfied" bucket to make it satisfied (*c.f.*, Figures 5.2(c)

and 5.2(d)). Repeat step 3 until no more buckets are unsatisfied or no more reallocation is possible. We will show that this reallocation will always terminate.

5. If all buckets are satisfied, return to step 1 with a more optimistic (higher) "guess" for c. If at least one bucket remains unsatisfied, return to step 1 with a more pessimistic (lower) guess for c.

eSPASS terminates with a value for c such that *all* buckets t have been assigned elements $\mathcal{A}_t$ such that $F(\mathcal{A}_t) \geq \beta c$. It guarantees that upon termination, c is an upper bound on the value of the optimal solution, hence providing a β approximation guarantee. In Section 5.5.2 we will show that $\beta = \frac{1}{6}$ suffices. In summary, we have the following guarantee about eSPASS.

Theorem 5.5.1. *For any monotonic, submodular function F and constant $\varepsilon > 0$,* eSPASS, *using* GAPS *as subroutine, returns a solution $\mathcal{A}_1, \ldots, \mathcal{A}_k$ such that*

$$\min_t F(\mathcal{A}_t) \geq \frac{1}{6} \max_{\mathcal{A}'} \min_t F(\mathcal{A}'_t) - \varepsilon.$$

eSPASS *requires at most $\mathcal{O}\left((1 + \log_2 F(\mathcal{V})/\varepsilon)kmn\right)$ evaluations of F.*

Here, ε is an tolerance parameter that can be made arbitrarily small. The number of iterations increases only logarithmically in $1/\varepsilon$.

Algorithm eSPASS (F, $\mathcal{V}$, k, m, ε)
$c_{\min} \leftarrow 0$; $c_{\max} \leftarrow F(\mathcal{V})$; $\beta \leftarrow 1/6$;
while $c_{\max} - c_{\min} \geq \varepsilon$ **do**
    $c \leftarrow (c_{\max} + c_{\min})/2$;
1 $\mathcal{B} \leftarrow \{s \in \mathcal{V} : F_c(\{s\}) \geq \beta c\}$;
    $k' \leftarrow k$;
2   **foreach** $s \in \mathcal{B}$ **do**
        $\mathcal{A}_{k'} \leftarrow \{s\}$; $k' \leftarrow k' - 1$;
        if $k' = 0$ **then** $c_{\min} \leftarrow c$;
            $\mathcal{A}_{best} \leftarrow (\mathcal{A}_1, \ldots, \mathcal{A}_k)$;
            continue with while loop;
    $\mathcal{V}' \leftarrow \mathcal{V} \setminus \mathcal{B}$; $m' \leftarrow m - |\mathcal{B}|$;
3 $\mathcal{A}_{1:k'} \leftarrow \mathrm{GAPS}(F_c, \mathcal{V}', k', m')$;
4   **if** $\sum_t F(\mathcal{A}_t) < k'c/2$ **then** $c_{\max} \leftarrow c$; **continue**;
    else
5   **while** $\exists i, j \leq k'$: $F_c(\mathcal{A}_j) \leq \beta c$, $F_c(\mathcal{A}_i) \geq 3\beta c$ **do**
            foreach $s \in \mathcal{A}_i$ **do**
                $\mathcal{A}_j \leftarrow \mathcal{A}_j \cup \{s\}$; $\mathcal{A}_i \leftarrow \mathcal{A}_i \setminus \{s\}$;
                if $F_c(\mathcal{A}_j) \geq \beta c$ **then break**;
        $c_{\min} \leftarrow c$; $\mathcal{A}_{best} \leftarrow (\mathcal{A}_1, \ldots, \mathcal{A}_k)$;

Algorithm 2: The eSPASS algorithm for simultaneously placing and scheduling sensors.

5.5.2 Algorithm details

We will now analyze each of the steps of eSPASS in detail. The pseudocode is given in Algorithm 2.

Removing big elements The main challenge when applying the GAPS algorithm to the truncated Problem (5.5.1) is exemplified by the following pathological example. Suppose the optimal value is c. GAPS, when applied to the truncated function F_c, could pick $k/2$ elements $s_1, \ldots, s_{k/2}$, with $F(\{s_i\}) = c$ each. While this solution obtains an average-case score of $c/2$ (one half of optimal as guaranteed by Theorem 5.4.1), there is no possibility to reallocate these $k/2$ elements into k buckets, and hence some buckets will remain empty, giving a balanced score of 0.

To avoid this pathological case, we would like to eliminate such elements $s \in \mathcal{V}$ with high individual scores $F(\{s\})$, to make sure that we can rearrange the solution of GAPS to obtain high balanced score. Hence, we distinguish two kinds of elements: Big elements s with $F(\{s\}) \geq \beta c$, and small elements s with $F(\{s\}) < \beta c$. If we intend to obtain a β approximation to the optimal score c, we realize that big elements have high enough value to each satisfy an individual bucket. Let $\mathcal{B}$ be the set of big elements (this set is determined in Line 1). If $|\mathcal{B}| \geq k$, we already have a β-approximate solution: Just put one big element in each bucket. If $|\mathcal{B}| < k$, put each element in $\mathcal{B}$ in a separate bucket $\mathcal{A}_1, \ldots, \mathcal{A}_{|\mathcal{B}|}$ (*c.f.*, Line 2). We can now set these satisfied buckets aside, and look at the reduced problem instance with elements $\mathcal{V}' = \mathcal{V} \setminus \mathcal{B}$, $m' = m - |\mathcal{B}|$ and $k' = k - |\mathcal{B}|$. Our first lemma shows that if the original problem instance $(F, \mathcal{V}, k, m)$ has optimal value c, the reduced problem instance $(F, \mathcal{V}', k', m')$ still has optimal value c.

Lemma 5.5.2. *The optimal value on the new problem instance $(F, \mathcal{V}', k', m')$ is still c.*

Hence, without loss of generality, we can now assume that for all $s \in \mathcal{V}$, $F(\{s\}) \leq \beta c$.

Solving the average-case problem In the next step of the algorithm, we run an α-approximate algorithm (such as GAPS where $\alpha = \frac{1}{2}$), using the truncated objective F_c, on the reduced problem instance containing only small elements (*c.f.*, Line 3). This application results in an allocation $\mathcal{A}_1, \ldots, \mathcal{A}_{k'}$ of elements into buckets. If $\sum_t F_c(\mathcal{A}_t) < \alpha c k'$, then we know that c is an upper bound to the optimal solution, and it is safe to set c_{max} to c (*c.f.*, Line 4) and continue with the binary search. Otherwise, we have a solution where $\sum_t F_c(\mathcal{A}_t) \geq \alpha c k'$. However, as argued in Section 5.4.2, this α-approximate solution could still have balanced score 0, if all the elements are allocated to only the first $\alpha k'$ buckets. Hence, we need to reallocate elements from satisfied into unsatisfied buckets to obtain a balanced solution.

Reallocation We will transfer elements from satisfied buckets to unsatisfied buckets, until all buckets are satisfied. Let us define a "reallocation move" as follows (*c.f.*, Line 5). Pick a bucket $\mathcal{A}_i = \{a_1, \ldots, a_l\}$ for which $F_c(\mathcal{A}_i) \geq 3\beta c$ (we will guarantee that such a bucket always exists), and a bucket $\mathcal{A}_j$ that is not satisfied, i.e., $F_c(\mathcal{A}_j) < \beta c$. Choose ℓ such that $F_c(\{a_1, \ldots, a_{\ell-1}\}) < \beta c$ and $F_c(\{a_1, \ldots, a_\ell\}) \geq \beta c$. Let $\Delta = \{a_1, \ldots, a_\ell\}$. Note that Δ is not empty since each a_i is small (i.e., $F_c(\{a_i\}) < \beta c$). We reallocate the elements Δ by removing Δ from $\mathcal{A}_i$ and adding Δ to $\mathcal{A}_j$.

Lemma 5.5.3. *It holds that*

$$F_c(\mathcal{A}_j \cup \Delta) \geq \beta c, \text{ and } F_c(\mathcal{A}_i \setminus \Delta) \geq F_c(\mathcal{A}_i) - 2\beta c.$$

Hence, removing elements Δ does not decrease the value of $\mathcal{A}_i$ by more than $2\beta c$, and thus $\mathcal{A}_i$ remains satisfied. On the other hand, the previously unsatisfied bucket $\mathcal{A}_j$ becomes satisfied by adding the elements Δ. We want to make sure that we can always execute our reallocation move, until all buckets are satisfied. The following result shows that if we choose $\beta = \frac{\alpha}{3}$, this will always be the case:

Lemma 5.5.4. *If we set $\beta = \frac{\alpha}{3}$, then, after at most k reallocation moves, all buckets will be satisfied, i.e., $F_c(\mathcal{A}_i) \geq \beta c$ for all i.*

Binary search Since the optimal value c is generally not known, we have to search for it. This is done using a simple binary search strategy, starting with the interval $[0, F(\mathcal{V})]$ which is guaranteed the optimal value due to monotonicity. At every step, we test the center c of the current interval. If all buckets can be filled to βc, then the truncation threshold c can be increased. If the algorithm for maximizing the average-case score (such as GAPS) does not return a solution of value at least αc, then that implies that the optimal value has to be less than c, and the truncation threshold is decreased (*c.f.*, Line 4). If F takes only integral values, then after at most $\lceil \log_2 F(\mathcal{V}) \rceil + 1$ iterations, the binary search terminates.

5.5.3 Improving the bounds

The bound in Theorem 5.5.1 is "offline" – we can state it independently of the specified problem instance. While guaranteeing that the obtained solutions cannot be arbitrarily bad, the constant factor 6 bound for eSPASS is typically rather weak, and we can show that our obtained solutions are typically much closer to the optimal value.

We can do this by computing the following data-dependent bounds on the optimal value. Let $\mathcal{A}' = (\mathcal{A}'_1, \ldots, \mathcal{A}'_k)$ be a candidate solution to Problem (5.3.2) (e.g., obtained using the eSPASS algorithm or any other algorithm). For every $1 \leq \ell \leq k$ and $s \in \mathcal{V}$ let $\delta_{\ell,s} = F(\mathcal{A}'_\ell \cup \{s\}) - F(\mathcal{A}'_\ell)$ be the increment in function value when adding sensor s to time slot ℓ.

Theorem 5.5.5. *The optimal value $c^* = \max_{\mathcal{A}} \min_t F(\mathcal{A}_t)$ is bounded by the solution c to the following linear program:*

$$\begin{aligned} &\max_{\lambda_{i,s},c} c \text{ s.t.} \\ &c \leq F(\mathcal{A}_i) + \sum_s \lambda_{i,s}\delta_{i,s} \text{ for all } i \\ &\sum_i \lambda_{i,s} \leq 1 \text{ for all } s \text{ and } \sum_{i,s} \lambda_{i,s} \leq m \end{aligned}$$

Theorem 5.5.5 states that for any given instance of the SPASS problem, and for a candidate solution $\mathcal{A}$ obtained by using *any* algorithm (not necessarily using eSPASS), we can solve a linear program to efficiently get an upper bound on the optimal solution. In Section 5.7 we will show that these bounds prove that our solutions obtained using eSPASS are often much closer to the optimal solution than guaranteed by the bound of Theorem 5.5.1.

5.6 Trading off Power and Accuracy

As argued in Section 5.1, deploying a larger number of sensors and scheduling them has the advantage over deploying a small number of sensors with large batteries that a high density mode can be supported. In contrast to scheduled mode, where the sensors are activated according to the schedule, in high-density mode all sensors are active, to provide higher resolution sensor data (e.g., to localize the boundary of a traffic congestion in our running example). For a fixed solution $\mathcal{A}_1, \ldots, \mathcal{A}_k$ to the SPASS problem, the (balanced) scheduled-mode sensing quality is $\min_i F(\mathcal{A}_i)$, whereas the high-density sensing quality is $F(\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k)$. Note that optimizing for the scheduled sensing quality does not necessarily lead to good high-density sensing quality. Hence, if both modes of operation should be supported, then we should simultaneously optimize for both performance measures. One such approach to this multicriterion optimization problem is to define the scalarized objective

$$\widehat{F}_\lambda(\mathcal{A}_1, \ldots, \mathcal{A}_k) = \lambda \min_i F(\mathcal{A}_i) + (1-\lambda) F(\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k),$$

and then solve the problem

$$\max_{\mathcal{A}_1 \ldots \mathcal{A}_k} \widehat{F}_\lambda(\mathcal{A}_1, \ldots, \mathcal{A}_k) \text{ s.t. } \mathcal{A}_i \cap \mathcal{A}_j = \emptyset \text{ if } i \neq j, \; |\bigcup_t \mathcal{A}_t| \leq m. \tag{5.6.1}$$

Note that if $\lambda = 1$, we recover the SPASS problem. Furthermore, as $\lambda \to 0$, the high-density sensing quality $F(\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k)$ dominates, and the chosen solution will converge to the stage-wise approach, where first the set $\mathcal{A}$ of all sensors is optimized, and then this placement is partitioned into $\mathcal{A} = \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k$. Hence, by varying λ between 1 and 0, we can interpolate between the simultaneous and the stage-wise placement and scheduling.

We modify eSPASS to approximately solve Problem (5.6.1), and call the modified algorithm mcSPASS (for *multicriterion Simultaneous Placement and Scheduling of Sensors*). The basic strategy is still a binary search procedure. However, instead of simply picking all available big elements (as done by eSPASS), mcSPASS will also guess (search for) the number ℓ of big elements used in the optimal solution. It will pick these big elements in a greedy fashion, resulting in a set $\mathcal{A}_{big} \subseteq \mathcal{V}$. For a fixed guess of c and ℓ, mcSPASS will again use GAPS as a subroutine. However, the objective function used by GAPS will be modified to account for the high-density performance:

$$G(\mathcal{A}_1, \ldots, \mathcal{A}_{k'}) = \lambda \sum_i F_c(\mathcal{A}_i) + (1-\lambda) F(\mathcal{A} \cup \mathcal{A}_{big}),$$

where $\mathcal{A} = \mathcal{A}_1 \cup \ldots \mathcal{A}_{k'}$, and $k' = k - \ell$. This modified objective function combines a component (weighted by λ) that measures the scheduled performance, as well as a component (weighted by $1-\lambda$) that measures the improvement in high-density performance, taking into account the set $\mathcal{A}_{big}$ of big elements that have already been selected. The reallocation procedure remains the same as in eSPASS. The remaining details of our mcSPASS approach are presented in the proof to the following theorem, which can be found in Section 5.10.

Theorem 5.6.1. *For any monotonic, submodular function F and constants $\varepsilon > 0$ and $0 \leq \lambda \leq 1$,* mcSPASS *will efficiently find a solution $\mathcal{A}_1, \ldots, \mathcal{A}_k$ such that*

$$\widehat{F}_\lambda(\mathcal{A}_1, \ldots, \mathcal{A}_k) \geq \frac{1}{8} \max_{\mathcal{A}'} \widehat{F}_\lambda(\mathcal{A}'_1, \ldots, \mathcal{A}'_k) - \varepsilon.$$

In Section 5.7 we will see that we can use this extension to obtain placements and schedules that perform well both in scheduled and high-density mode.

5.7 Transportation Applications

In the application section we develop two transportation applications for the methodology suggested in the chapter: placing and sampling a set of fixed traffic sensors and privacy preserving sampling of mobile traffic sensors. In the first application, the optimization decision is determining *where* the sensors are positioned, and how they should be grouped to increase lifetime, without sacrificing coverage quality. Measuring placement and coverage quality requires defining a placement benefit function.

In the second application, the sensors are not fixed, but are instead drivers who record speeds as they move along the road network. In this case, the cost of an observation is the privacy cost of revealing the location of a driver at a given time of day. The benefit obtained by measurements is the coverage attained when using data. In a privacy preserving setting, we can divide drivers into groups, and observe only a single driver group for a given interval of time, preserving the privacy of other driver groups. This in turn, guarantees that a driver will be observed for at most a fraction of total time. This application also requires a function that measures the placement benefit, and a corresponding spatial statistical model to justify this benefit.

5.7.1 Modeling transportation data

We use the highway network model explained in Chapter 2, where the network is divided into a collection of sections $\mathcal{S}$. Usually, we can take the boundary of sections to be the minimum distance we would like to allow between consecutive placements. In our sensor placement experiments, each section corresponds to a single lance, centered at a loop detector. The boundaries of a section are midway to the upstream and downstream sensors.

Time is divided into 5-minute intervals. For each interval, the speed at all the sections is jointly modeled as a spatial Gaussian Process (GP) [Cressie, 1991], with mean function $\mu(s)$ and covariance function $K(s, t)$, where s is the section index. Therefore, at each section $s \in \mathcal{S}$ in a given time interval, the speed is a random variable $X(s)$. For any set of locations $\mathcal{A} = \{s_1, ..., s_n\}$, the Gaussian Process induces a multivariate normal distribution, with mean vector $\mu_{\mathcal{A}} = (\mu(s_1), ..., \mu(s_n))$ and covariance matrix $\Sigma_{\mathcal{A}\mathcal{A}} = (K(s_i, s_j))_{ij}$ obtained by evaluating the covariance function for every pair of points s_i and s_j in $\mathcal{A}$.

If the speed is observed at a set locations $X_{\mathcal{A}} = \mathbf{x}_{\mathcal{A}}$, the speed at the remaining locations $\mathcal{A}_c = \mathcal{S}/\mathcal{A}$ can be optimally estimated in the mean-squared sense through the conditional expectation $\mathbb{E}[X_{\mathcal{A}_c}|X_{\mathcal{A}} = x_{\mathcal{A}}]$. The Gaussian Process allows us to compute the expectation and corresponding mean squared error for each component (conditional covariance of X_y for $y \in \mathcal{A}_c$):

$$\begin{aligned}
\mathbb{E}[X_{\mathcal{A}_c}|X_{\mathcal{A}} = x_{\mathcal{A}}] &= \mu_{\mathcal{A}_c} + \Sigma_{\mathcal{A}_c\mathcal{A}}\Sigma_{\mathcal{A}\mathcal{A}}^{-1}(x_{\mathcal{A}} - \mu_{\mathcal{A}}), \\
\mathrm{Var}[X_y|X_{\mathcal{A}} = x_{\mathcal{A}}] &= \mathbb{E}[(X_y - E[Xs|X_{\mathcal{A}} = x_{\mathcal{A}}])^2|X_{\mathcal{A}} = x_{\mathcal{A}}], \\
&= \Sigma_{y,y} - \Sigma_{y\mathcal{A}}\Sigma_{\mathcal{A}\mathcal{A}}^{-1}\Sigma_{\mathcal{A}y}.
\end{aligned}$$

We can now evaluate the expected *variance reduction* from observing a set $\mathcal{A}$ as:

$$F(\mathcal{A}) = \sum_{y \in \mathcal{A}_c} (\Sigma_{y\mathcal{A}}\Sigma_{\mathcal{A}\mathcal{A}}^{-1}\Sigma_{\mathcal{A}y}). \tag{5.7.1}$$

The expected variance reduction function can be shown to be submodular under certain simple conditions [Das and Kempe, 2008]. Furthermore, one interesting property of the variance reduction is that it does not depend on the observations $x_{\mathcal{A}}$.

In traffic applications, the model parameters for the Gaussian Process are learned from either existing historical data, as available for many highways, or from a preliminary (small) sensor deployment. The covariance function and mean function definitions can be extended for whole highways using appropriate covariance interpolation methods [Cressie, 1991]. In some cases, when no sensors are available, or a preliminary deployment is impossible, a sequential placement methodology can be considered [Krause and Guestrin, 2007] for placing a initial set of sensors to learn the model. When existing data is used to learn the parameters, the Gaussian Process model cost function reduces to the total mean squared error resulting from regressing all data from each unobserved location against the observed location information: $X_y = \alpha_y^t X_{\mathcal{A}} + \beta_y$, $y \in \mathcal{A}_c$.

In the **mobile observation model**, we consider a similar benefit function to the one in Equation 5.7.1. In this case when a mobile sensor (driver) is added to the observation set, all his samples for each one of the road links are added to the observation set. The principal addition in the model is that a *demand* weighting is used to average the variance reduction, as described in [Krause et al., 2008a]. The idea is that more demanded links are more heavily weighted in the computation of the total benefit. Let the demand for link s be denoted by the random variable D_s, then the expected cost reduction is given by

$$F(\mathcal{A}) = \sum_{y \in \mathcal{A}_c} \mathbb{E}[D_s](\Sigma_{y\mathcal{A}}\Sigma_{\mathcal{A}\mathcal{A}}^{-1}\Sigma_{\mathcal{A}y}).$$

If we assume a simple Poisson model with rate λ_s, $\mathbb{E}[D_s] = \lambda_s$. This parameter can be estimated from preexisting traffic counts on each road link, or from information collected from driver routing desires.

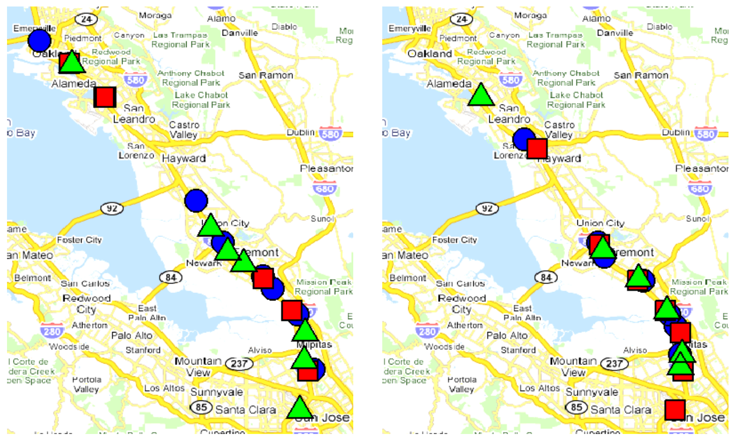
5.7.2 Highway monitoring using fixed sensors

The California highways are currently monitored by over 25,000 traffic sensors based on older technologies. As these loops fail, they are being replaced by novel wireless sensor network technologies, and it has become important to identify economic deployment strategies. PeMS [PeMS, 2009] is a website and project that integrates, cleanses and tracks real time traffic information for the whole state, computing key performance indicators. The sensors typically report speed, flow and vehicle counts every 30 seconds, and PeMS further aggregates the data into 5 minute blocks. For this case study, we use data from highway I-880 South, which extends for 35 miles in northern California (Figure 5.3) and has between 3 and 5 lanes. This highway experiences heavy traffic, and accurate measurements are essential for proper resource management. Measurement variation is mainly due to congestion and events such as accidents and road closures. There are 88 measurement sites along the highway, on average every 0.4 miles, which comprise 357 sensors covering all lanes. We use speed information from lanes, for all days of the week in a single month, excluding weekends and holidays for the period from 6AM to 11AM, which is the time when the highway is congested. This is the most difficult time for making predictions, as when there is no congestion, even a free flow speed prediction of 60 mph is accurate.

The number and locations of sensors are limited by costs and physical deployment constraints. Typically at each location, it is only possible to place one sensor at each lane. Furthermore, lane closures for sensor installation are very costly. Given these constraints, California requires that sensor technologies have a target lifetime of 10 years. This implies that most wireless sensor solutions require intelligent scheduling in order to extend the lifetime by four times, since most sensor network solutions batteries are expected to last 2 to 3 years. Including more batteries in a single sensor is not viable, as sensors have physical constraints to avoid disrupting the existing pavement structure and keep installation costs at a minimum.

As wireless sensors displace existing loop technologies, it is desirable to place as few sensors as possible, without trading off too much sensing quality. To achieve these goals in a principled manner, historical loop data from the current deployment should be used. eSPASS provides a solution which can balance these conflicting requirements, by combining scheduling and placement: more sensors are placed initially, still keeping road closures at a minimum, and scheduling is used to extend the lifetime of the network, keeping sensing quality balanced. In this section we explore this solution and compare eSPASS to other simultaneous placement and scheduling solutions.

Simultaneous vs. stage-wise optimization In our first experiment, we study the benefit of simultaneously placing and scheduling sensors. For varying numbers m of sensors and k of time slots, we use different strategies to find k disjoint sets $\mathcal{A}_1, \dots, \mathcal{A}_k$, where $\mathcal{A}_i$ is the sensors active at time slot i. We compare the simultaneous placement and schedule (optimized using eSPASS and GAPS) with solutions obtained by first placing sensors at a fixed set of locations, and then scheduling them. We consider both optimized and random sensor placements, followed by optimized and random scheduling, amounting to four stage-wise strategies. For random placements and schedules, we report the mean and standard error over 20 random trials.



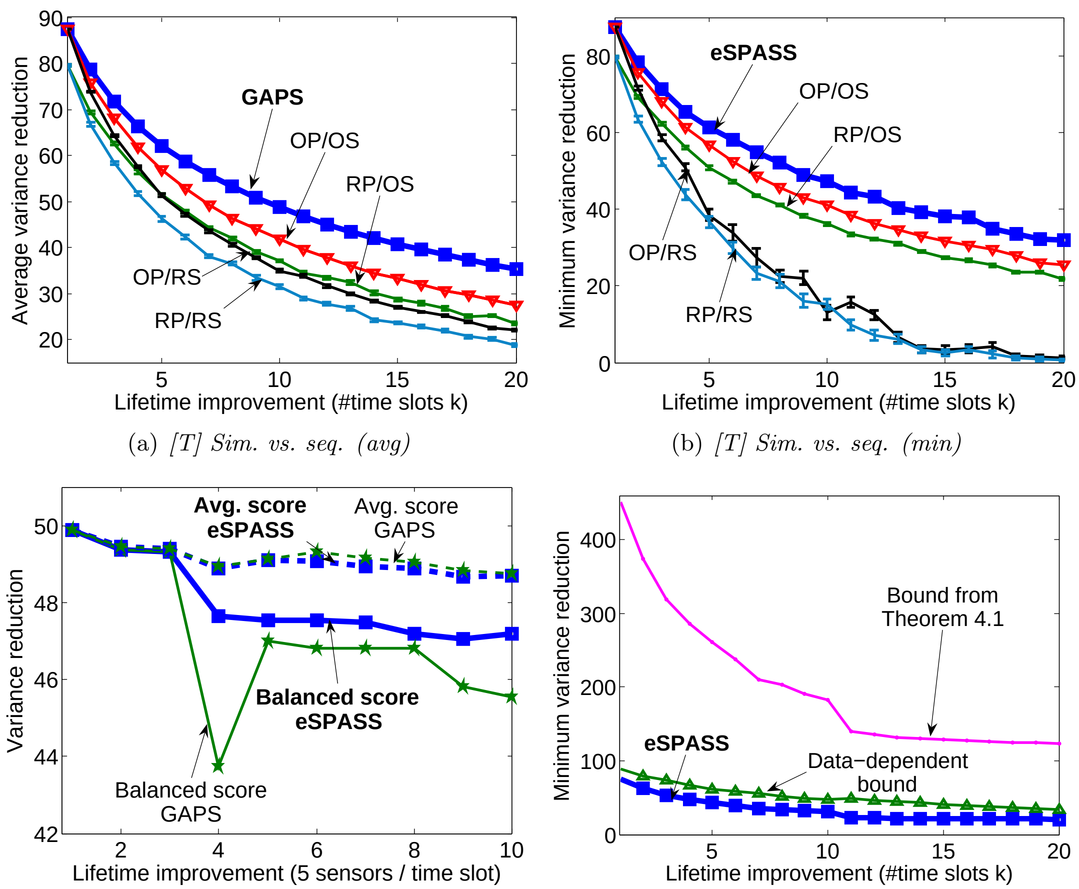
(a) *[T] stage-wise* (b) *[T] Simultaneous*

Figure 5.3. Placements and schedules for the traffic data. (a) Stage-wise approach, (b) SPASS solution.

Figure 5.4(a) presents the performance of the five strategies when optimizing the average-case performance, for a fixed number of $m = 50$ sensors and a number of time slots k varying from 1 to 20. GAPS performs best, followed by the stage-wise optimized placement and schedule (OP/OS). Of the two strategies where one component (either the placement or the schedule) is randomized (OP/RS and RP/OS), for small numbers (≤ 3) of time slots OP/RS performs slightly better, and for large numbers of time slots (≥ 10), RP/OS performs slightly better. The completely randomized solution performs significantly worse.

Figure 5.4(b) presents the same results when optimizing the balanced criterion. eSPASS outperforms the stage-wise strategies and the completely randomized strategy RP/RS performs worst as expected. Interestingly, for the balanced criterion, OP/RS performs drastically worse than RP/OS for $k \geq 4$ time slots. We hypothesize this to be due to the fact that a poor random placement can more easily be compensated for by using a good schedule than vice versa: When partitioning a sensor placement of 50 sensors randomly into a large number of time slots, it is fairly likely that at least one of the timeslots exhibits poor performance, hence leading to a poor balanced score. This insight also suggests that the larger the intended improvement in network lifetime (number of time slots), the more important it is to optimize for a balanced schedule.

To further investigate the phenomenon discussed above, we performed another experiment, where we increased the number k of time slots from 1 to 10, but instead of keeping the number m of sensors fixed, we allocate a fixed number of 5 sensors for each time slot, i.e., keep the ratio k/m fixed at 5. Figure 5.5(a) presents the result of this experiment. Again, eSPASS outperforms the other strategies by a fair margin. Note that the performance of eSPASS decreases slowly with the number of time steps. This is because as more and more timeslots are used, the algorithm has fewer and fewer choices of possible locations to balance the schedule. Also note that the performance of the RP/OS strategy actually increases with the number of time slots. Due to submodularity, as a larger number of sensors is placed, the smaller the benefit of optimizing the sensor placement becomes.



(a) *[T] Sim. vs. seq. (avg)*

(b) *[T] Sim. vs. seq. (min)*

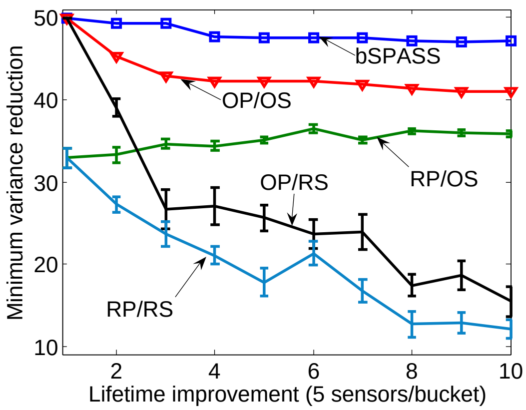
(c) *[T] Avg. vs. balanced*

(d) *[T] Bounds*

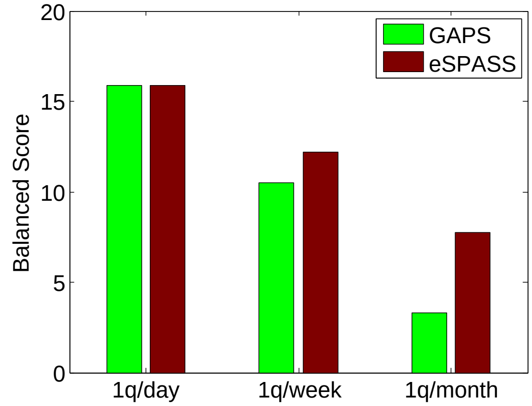
Figure 5.4. Results for traffic monitoring [T]. (a,b) compare simultaneous placement and scheduling to stage-wise strategies on (a) average-case and (b) balanced performance ($m = 50$, k varies). (c) compare average-case and balanced performance, when optimizing for average-case (using GAPS) and balanced (using eSPASS) performance. (d) "Online" (data-dependent) bounds show that the eSPASS solutions are closer to optimal than the factor 6 "offline" bound from Theorem 5.5.1 suggests.

The experiment also confirms the observation of the degrading performance of OP/RS and RP/RS as the number of time slots increases.

To summarize this analysis, we see that simultaneous placement and scheduling drastically outperforms the stage-wise strategies. For example, if we place 50 sensors at random, and then use eSPASS to schedule them into 4 time slots, we achieve an estimated minimum reduction in Mean Squared error by 58%. If we first optimize the placement and then use eSPASS for scheduling, we can achieve the same amount of variance reduction by scheduling 6 time slots (hence obtaining a 50% increase in network lifetime). If instead of stage-wise optimization we simultaneously optimize the placement and the schedule using eSPASS, we can obtain the same variance reduction by scheduling 8 time slots, hence an increase in network lifetime by 100%.



(a) *[T] Sim. vs. seq.*



(b) *[C] Avg. vs. balanced*

Figure 5.5. (a) (b) Results for community sensing [C]. When querying each car only once each week (using the eSPASS schedule), the sensing quality is only 23% lower than when querying every day.

Average vs. balanced performance We have seen that simultaneously placing and scheduling can drastically outperform stage-wise strategies, for both the average-case and the balanced objective. But which of the objectives should we use? In order to gain insight into this question, we performed the following experiment. For varying k and m, we obtain solutions to the SPASS problem using both the eSPASS and the GAPS algorithm. We then evaluate the respective solutions both using the average-case and the balanced criterion. Figure 5.4(c) presents the results of this experiment for k varying from 1 to 10, and fixed ratio of 5 sensors per time slot. As expected, eSPASS outperforms GAPS with respect to the balanced criterion, and GAPS outperforms eSPASS according to the average-case criterion. However, while eSPASS achieves average-case score very close to the solution obtained by GAPS, the balanced score of the GAPS solutions are far worse than those obtained by eSPASS. Hence, optimizing for the balanced criterion performs well for the average case, but not vice versa.

Online bounds In order to see how close the solutions obtained by eSPASS are to the optimal solution, we also compute the bounds from Theorem 5.5.5. Figure 5.4(d) presents the bounds on the maximum variance reduction achievable when placing 50 sensors and partitioning them into an increasing number of groups. We plot both the factor 6 bound due to Theorem 5.5.1, as well as the data-dependent bound due to Theorem 5.5.5. We can see that the data dependent bounds are much tighter. For example, if we partition the sensors into 2 groups, our solution is at least 78% of optimum, for 5 groups it is at least 70% of optimum (rather than the 17% of Theorem 5.5.1).

5.7.3 Highway monitoring using privacy-preserving mobile sensors

While the static deployment of sensors has become an important means for monitoring traffic on highways and arterial roads, due to high deployment and maintenance cost it is difficult to extend sensor coverage to urban side-streets. However, in order to optimize road-network utilization, accurate estimate of side-street conditions is necessary.

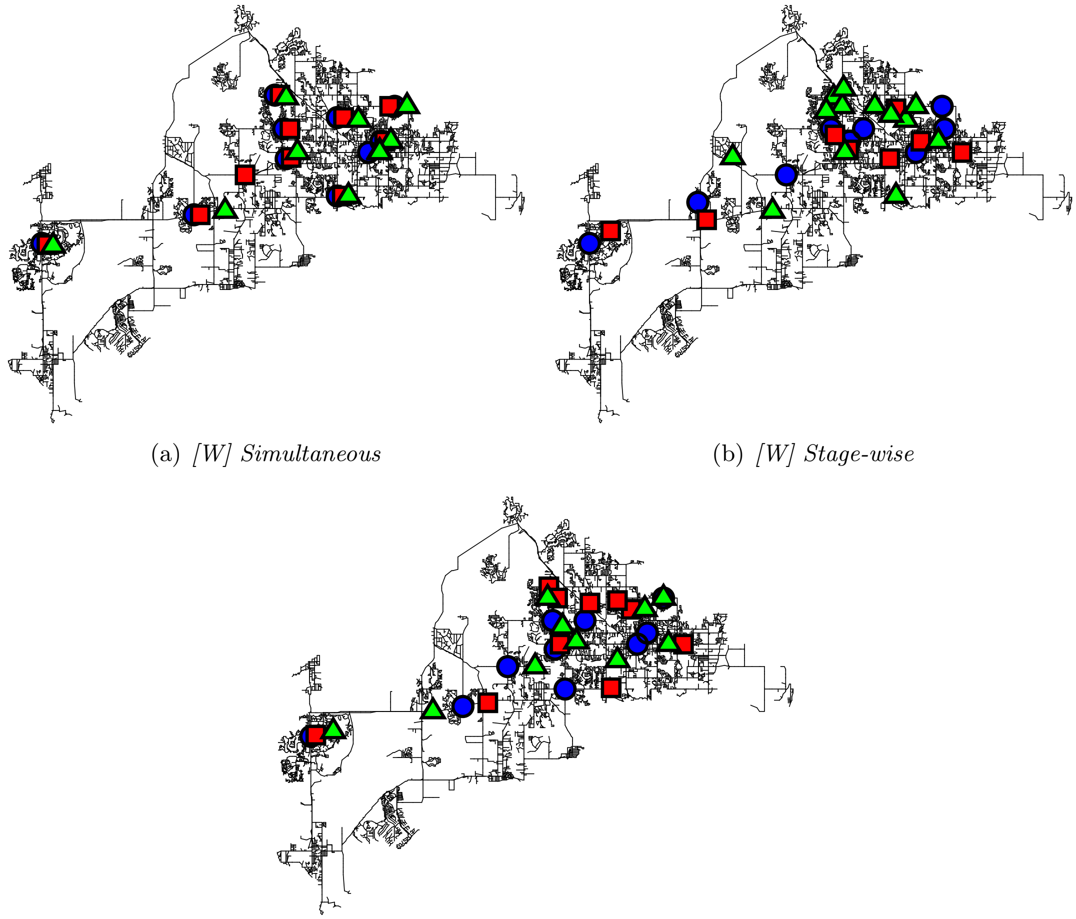
Instead of (or in addition to) statically deploying sensors, a promising approach is to utilize cars as traffic sensors: An increasing number of vehicles nowadays are equipped with GPS and Personal Navigation Devices, which can accurately localize a car on a road network. Furthermore, these devices are becoming connected to wireless networks, using, e.g., GPRS or Edge connectivity, through which they could report their location and speed. Hence, in principle, it is possible to access accurate sensor data through the network of cars.

Such a network of non-centrally owned sensors present significant challenges. While users may generally consider sharing their sensor data, they have reasonable concerns about their privacy. Krause et al. [2008a] provide methods for *community sensing*, describing strategies for selectively querying a community sensor network while maintaining preferences about privacy. They demonstrated how the selective querying of such a community sensor network can be modeled as the problem of optimizing a monotonic submodular sensing quality function that considers demand based on road usage. Preferences about privacy map to constraints in the optimization problem.

One basic preference that needs to be supported is the preference that each user is queried at most once in a specified time interval (e.g., queried at most once each week or month). Let $\mathcal{V}$ be the set of users subscribing to the community sensing service. In order to query each user at most once in k time steps, one strategy would be to partition the users into k sets $\mathcal{A}_1, \ldots, \mathcal{A}_k$, such that at time step t, users $\mathcal{A}_t$ are queried. In order to obtain continuously high performance of the monitoring service, we want to make sure that the performance $F(\mathcal{A}_t)$ is maximized simultaneously over all time steps. This is exactly an instance of the SPASS problem.

In order to evaluate the performance of the eSPASS algorithm, we used the experimental setup of Krause et al. [2008a], using real traffic data from 534 detector loops deployed underneath highways, GPS traces from 85 volunteer drivers and demand data based on directions generated in response to requests to a traffic prediction and route planning prototype named ClearFlow, developed at Microsoft Research[5]. Details about the data sets are described by Krause et al. [2008a]. Based on this experimental setup, we compare the performance of the eSPASS and GAPS algorithms. Using each algorithm, we partition the users into 7 or 31 sets (i.e., querying each user at most once each week or month, both of which are possible options for privacy preferences). We then evaluate the performance based on the worst case prediction error over all test time steps. Figure 5.5(b) presents the results of this experiment. We can see that the eSPASS solutions outperform the GAPS solutions. When partitioning into 31 sets, the worst prediction performance of eSPASS is more than twice as good than the worst prediction performance of GAPS. Most importantly, this experiment shows that, using eSPASS for scheduling, one can obtain a very high balanced performance, even when querying each individual car only very infrequently. For example, when querying each car only once each week, the balanced sensing quality is only 23% lower than that obtained by the privacy-intrusive continuous (daily) querying.

[5]The ClearFlow research system, available only to users within Microsoft Corporation, was the prototype for the Clearflow context-sensitive routing service now available publicly for North American cities at http://maps.live.com.



(a) *[W] Simultaneous* (b) *[W] Stage-wise*

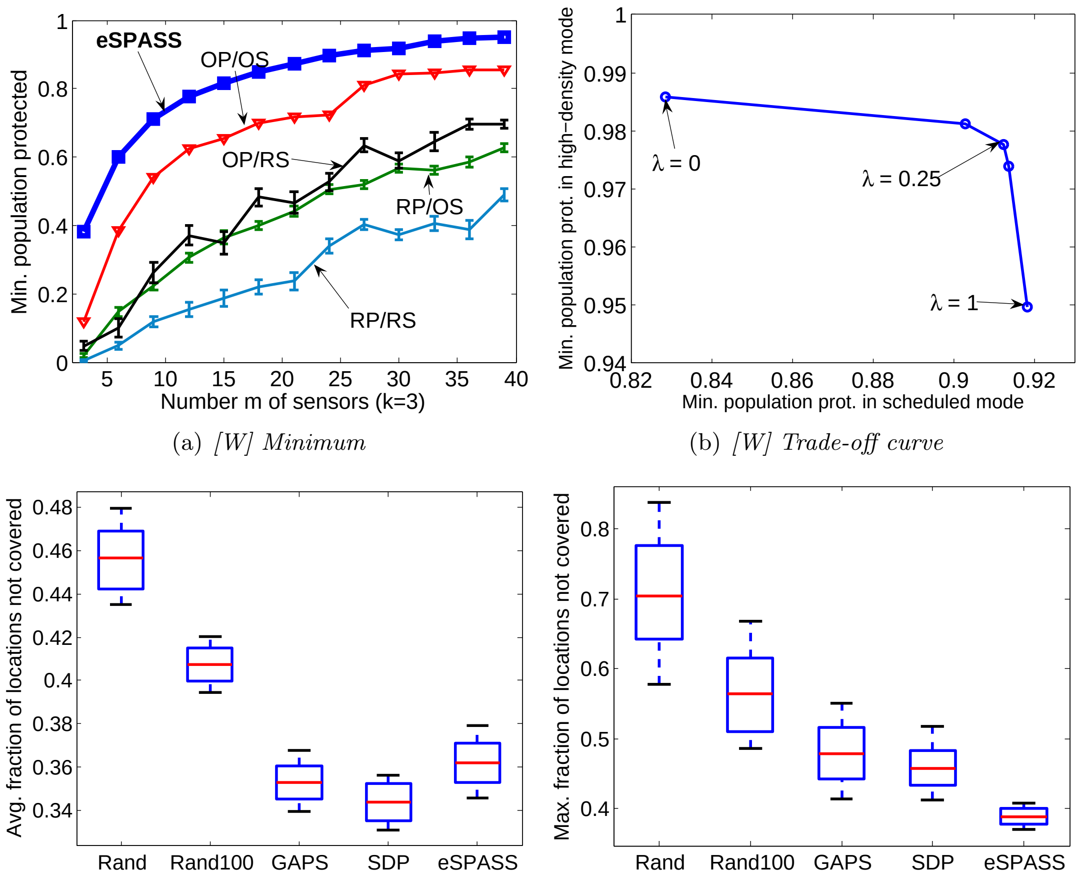
(c) *[W] Multicrit. ($\lambda = 0.25$)*

Figure 5.6: Example placements and schedules for water networks [W].

Even if querying only once each month (i.e., a factor 31 more infrequently), the balanced performance is only reduced by approximately a factor of 2. These results indicate that, using eSPASS, even stringent preferences about privacy can be met without losing much prediction accuracy.

5.8 Other Applications

In order to compare our proposed algorithm with existing approaches in the literature, we also considered alternative data sets and problem formulations, for which those algorithms were designed. The corresponding subsections describe these problems and the corresponding placement benefit function.



(a) *[W] Minimum* (b) *[W] Trade-off curve*

(c) *[S] Average performance* (d) *[S] Balanced performance*

Figure 5.7. (a,b) Contamination detection in water networks [W]. (a) compares simultaneous and stage-wise solutions. (b) power/accuracy tradeoff curve with strong knee. (c,d) compares eSPASS with existing solutions by Abrams et al. [2004] and Deshpande et al. [2008] on synthetic data [S].

5.8.1 Contamination detection

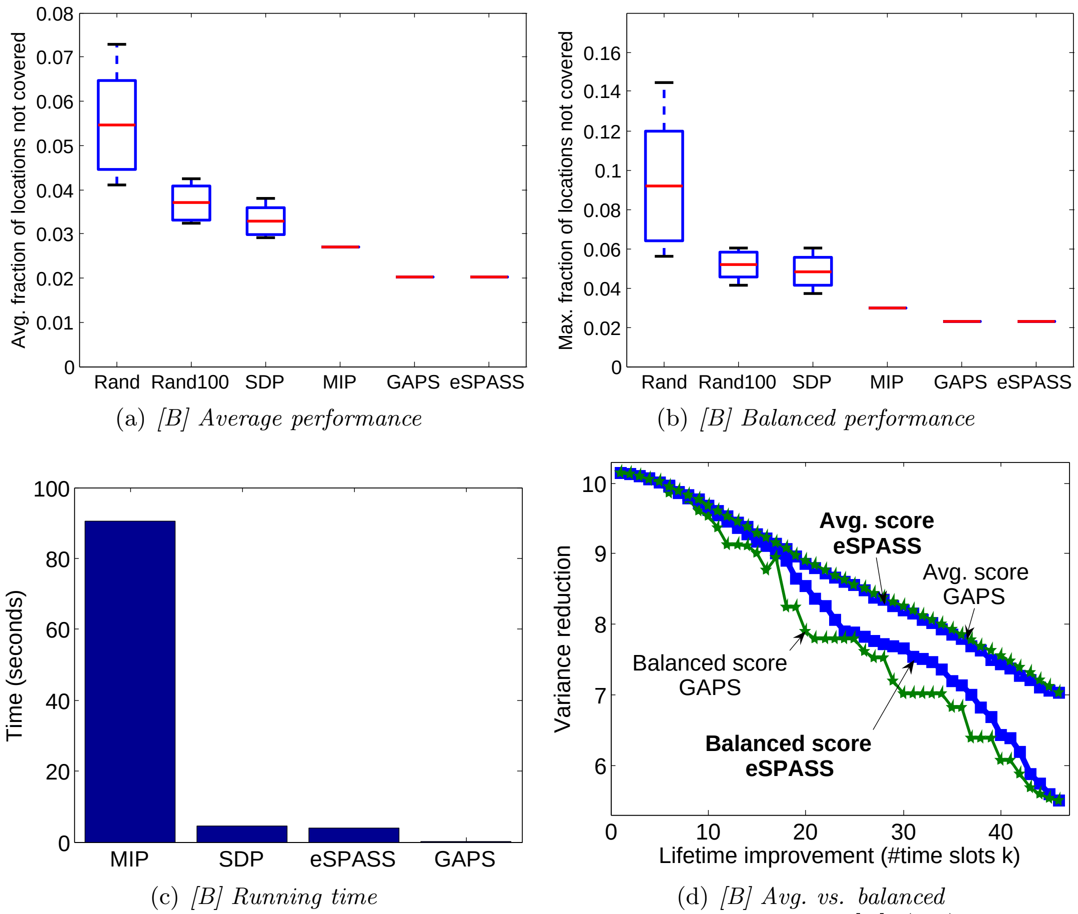
Consider a city water distribution network, delivering water to households via a system of pipes, pumps and junctions. Accidental or malicious intrusions can cause contaminants to spread over the network, and we want to select a few locations (pipe junctions) to install sensors, in order to detect these contaminations as quickly as possible. In August 2006, the Battle of Water Sensor Networks (BWSN) [et al., 2008] was organized as an international challenge to find the best sensor placements for a real (but anonymized) metropolitan water distribution network, consisting of 12,527 nodes. In this challenge, a set of intrusion scenarios is specified, and for each scenario a realistic simulator provided by the EPA [Rossman, 1999] is used to simulate the spread of the contaminant for a 48 hour period. An intrusion is considered detected when one selected node shows positive contaminant concentration.

The goal of BWSN was to minimize impact measures, such as the expected population affected, which is calculated using a realistic disease model. Krause et al. [2008b] showed that the function $F(\mathcal{A})$ which measures the expected population protected by placing sensors at location $\mathcal{A}$ is a monotonic submodular function. Water quality sondes can operate for a fairly long amount of time on battery power. For example, the YSI 6600 Sonde can sample 15 water quality parameters every 15 minutes for 75 days. However, for the long-term feasibility it is desirable to considerably improve this battery lifetime by sensor scheduling. On the other hand, high sampling rates are desirable to ensure rapid response to possible contaminations. For a security-critical sensing task such as protecting drinking water from contamination, it is important to obtain balanced, uniformly good detection performance over time. In addition, deployment and maintenance cost restrict the number of sensors that can be deployed. Hence, the problem of deploying battery powered sensors for drinking water quality monitoring is another natural instance of the SPASS problem.

We reproduce the experimental setup detailed in [Krause et al., 2008b]. However, instead of only optimizing for the sensor placement, we simultaneously optimize for placement and schedule using the eSPASS algorithm. Figure 5.7(a) compares eSPASS with the stage-wise approaches. For each algorithm, we report the population protected by placing sensors, normalized by the maximum protection achievable when placing sensors at every node in the network.

Simultaneous vs. stage-wise optimization eSPASS obtains drastically improved performance when compared to the stage-wise approaches. For example, when scheduling 3 time slots, in order to obtain 85% protection, eSPASS requires 18 sensors. The fully optimized stage-wise approach (OP/OS) requires twice the number of sensors. When placing 36 sensors, the stage-wise approach leaves 3 times more population unprotected as compared to the simultaneous eSPASS solution with the same number of sensors. eSPASS solved this large scale optimization task ($n = 12,527$, $k = 3$, $m = 30$) in 26 minutes using our MATLAB implementation.

Trading off power and accuracy We also applied our modified eSPASS algorithm in order to trade off scheduled mode and high density mode performance. For a fixed number of $m = 30$ sensors and $k = 3$ time slots, we solve Problem (5.6.1) for values of λ varying from 0 to 1. For each value of λ, we obtain a different solution, and plot the normalized expected population protected (higher is better) both in scheduled- and in high-density mode in Figure 5.7(b). We can see that this trade-off curve exhibits a prominent knee, where solutions are obtained that perform nearly optimally with respect to both criteria. Figures 5.6(a), 5.6(b) and 5.6(c) show the placements and schedules obtained for $\lambda = 1$ (i.e., ignoring the high-density sensing quality), $\lambda = 0$ (ignoring the schedule, effectively performing a stage-wise approach) and a value $\lambda = 0.25$ (from the knee in the trade-off curve) respectively. Note how the solution for $\lambda = 1$ clusters the sensors closely together, obtaining three very similar placements $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ for each time slot (similar as in Figure 5.1). The solution for $\lambda = 0$ spreads out the sensors more, having to leave, e.g., the Western part of the network uncovered in the time slot indicated by the green triangle. The multicriterion solution ($\lambda = 0.25$) is a compromise between the former two solutions: The sensors are still clustered together, but also spread out more – the Western part of the network can be covered in this solution.



(a) *[B] Average performance*

(b) *[B] Balanced performance*

(c) *[B] Running time*

(d) *[B] Avg. vs. balanced*

Figure 5.8. Results on temperature data from Intel Research Berkeley [B]. (a,b) compares eSPASS with existing solutions. (c) compares running time. (d) compares average-case and balanced performance.

5.8.2 Comparison with existing techniques

We also compare eSPASS with several existing algorithms. Since the existing algorithms apply to the scheduling problem only, we call eSPASS with $m = |\mathcal{V}|$ (i.e., allow it to select all sensors).

Set covering Most existing algorithms for sensor scheduling assume that sensors are associated with a fixed sensing region that can be perfectly observed by the sensor (*c.f.*, Abrams et al. [2004]; Deshpande et al. [2008]). In this setting, we associate with each location $s \in \mathcal{A}$ a set $\mathcal{R}_s \subseteq \mathcal{V}$ of locations that can be monitored by the sensor, and define the sensing quality $F(\mathcal{A}) = |\bigcup_{s\in\mathcal{A}} \mathcal{R}_s|$ to be the total area covered by all sensors. Since set coverage is an example of a monotonic submodular function, we can use eSPASS to optimize it.

We compare eSPASS to the greedy approach by Abrams et al. [2004], as well as the the approach by Deshpande et al. [2008] that relies on solving a semidefinite program (SDP). We use the synthetic experimental setup defined by Deshpande et al. [2008] to compare the approaches. A set of n sensors is used to cover M regions. Each sensor s is associated with a set $\mathcal{R}_s$ of regions it covers. The objective is to divide the n sensors into k groups (buckets), such that the minimum or the average number regions covered by each group is maximized.

For the SDP by Deshpande et al. [2008], we solve the SDP using SeDuMi to get a distribution over possible schedules, and then pick the best solution out of 100 random samples drawn from this distribution. For the random assignment approach (Rand100) of Abrams et al. [2004], we sample 100 random schedules and pick the best one. In addition, we run the GAPS and the eSPASS algorithms. We apply those four algorithms to 50 random set cover instances as defined by Deshpande et al. [2008]: for each sensor, a uniform random integer r between 3 and 5 is chosen, and then the first r regions from a random permutation of the set of M regions is assigned to that sensor. The sensor network size is $n = 20$, the number of desired groups $k = 5$ and the number of regions is $M = 50$.

Figure 5.7(c) presents the average performance of the four approaches. In this setting, the SDP performs best, closely followed by GAPS and eSPASS. Figure 5.7(d) presents the balanced performance of the four approaches. Here, eSPASS significantly outperforms both the SDP and the GAPS solution.

Building monitoring As argued in the introduction, for complex spatial monitoring problems, the sensing region assumption is unrealistic, and we would rather like to optimize prediction accuracy directly. The approach by Koushanfar et al. [2006] is designed to schedule sensors under constraints on the prediction accuracy. Their approach, given a required prediction accuracy, constructs a prediction graph that encodes which sensors can predict which other sensors. They then solve a set partitioning problem, selecting a maximal number of disjoint subsets that can predict all other sensors with the desired accuracy. In order to determine the domatic partitioning, their algorithm relies on the solution of a Mixed Integer Program (MIP). However, solving MIPs is NP-hard in general, and unfortunately, we were not able to scale their approach to the traffic data application. Instead, we use data from 46 temperature sensors deployed at Intel Research, Berkeley (*c.f.*, Deshpande et al. [2004]).

On this smaller data set, we first apply the MIP for domatic partitioning, with a specified accuracy constraint. The MIP was very sensitive with respect to this accuracy constraint. For just slightly too small values of ε, the MIP returned a trivial solution consisting of only a single set. For slightly too large values, the MIP had to consider partitions into a large number of possible time slots, increasing the size of the MIP such that the solver ran out of memory. Requiring that sensors can predict each other with a Root Mean Squared (RMS) error of 1.25 Kelvin leads to a selection of $m = 19$ sensors, partitioned into $k = 3$ time slots. Using this setting for m and k, we run the GAPS and eSPASS algorithms, which happen to return the same solution for this example. In order to compare these solutions with the SDP and random selection from the previous section, we apply them to the prediction graph induced by the required prediction accuracy. We first randomly select 19 locations, and then partition them into 3 groups using the SDP and Rand100 approach, respectively. As a baseline, we randomly select 3 groups totaling 19 sensors (Rand). For these randomized techniques, we report the distribution over 20 trials. All approaches are evaluated based on the variance reduction objective function.

Figure 5.8(a) presents the result for optimizing the average variance reduction, and Figure 5.8(b) for the minimum variance reduction. In both settings, GAPS and eSPASS perform best, obtaining 23% less remaining maximum variance when compared to the MIP solution of Koushanfar et al. [2006]. Furthermore, using YalMIP in Matlab, solving the MIP requires 95 seconds, as compared to 4 seconds for the SDP and 3.8 seconds for eSPASS (Figure 5.8(c)). Even though the MIP returns an optimum solution for the domatic partition of the prediction graph, eSPASS performs better since it uses the fact that the combination of multiple sensors can lead to better prediction accuracy than only using single sensors for prediction. Even the best out of 20 random trials for the SDP performs worse than the MIP, due to the approximate nature of the algorithm and the random selection of the initial 19 sensors. The Rand100 approach does performs only slightly (not significantly) worse than the SDP based approach.

5.9 Discussion

When deploying sensor networks for monitoring tasks, both placing and scheduling the sensors are of key importance, in order to ensure informative measurements and long deployment lifetime. Traditionally, the problems of sensor placement and scheduling have been considered separately from each other. In this Chapter, we have presented an efficient algorithm, eSPASS, that simultaneously optimizes the sensor placement and the schedule. We considered both the setting where the average-case performance over time is optimized, as well as the balanced setting, where uniformly good performance is required. Such balanced performance is crucial for security-critical applications such as contamination detection. Our results indicate that optimizing for balanced performance often yields good average-case performance, but not necessarily vice versa. We proved that our eSPASS algorithm provides a constant factor 6 approximation to the optimal balanced solution. To the best of our knowledge, eSPASS is the first algorithm that provides strong guarantees for this problem, partly resolving an open problem raised by Abrams et al. [2004]. Furthermore, our algorithm applies to any setting where the sensing quality function is submodular, which allows to address complex sensing tasks where one intends to optimize prediction accuracy or optimize detection performance.

We also considered complex sensor placement scenarios, where the deployed sensor network must be able to function well both in a scheduled, low power mode, but also in a high accuracy mode, where all sensors are activated simultaneously. We developed an algorithm, mcSPASS, that directly optimizes this power-accuracy tradeoff. Our results show that mcSPASS yields solutions which perform near-optimally with respect to both the scheduled and the high-density performance.

We extensively evaluated our approach on several real-world sensing case studies, including traffic and building monitoring as well as contamination detection in metropolitan area drinking water networks. When applied to the simpler special case of sensor scheduling (i.e., ignoring the placement aspect), eSPASS outperforms existing sensor scheduling algorithms on standard data sets. For the more complex, general case, our algorithm performs provably near-optimal (as demonstrated by tight, data-dependent bounds). Our results show that, for fixed deployment budget, drastic improvements in sensor network lifetime can be achieved by simultaneously optimizing the placement and the schedule, as compared to the traditional, stage-wise approach. For example, for traffic prediction, eSPASS achieves a 33% improvement in network lifetime compared to the setting where placement and scheduled are optimized separately, and a 100% improvement when compared to the traditional setting where sensors are first randomly deployed and then optimally scheduled.

We believe that the results presented in this Chapter present an important step towards understanding the deployment and maintenance of real world sensor networks, and in particular, transportation sensing networks.

5.10 Proofs

5.10.1 Proof of Theorem 5.4.1

Proof. Define a new ground set $\mathcal{V}' = \mathcal{V} \times \{1, \dots, k\}$, and a new function

$$F'(\mathcal{A}') = \sum_{t=1}^{k} F(\{s : (s,t) \in \mathcal{A}'\}).$$

F' is monotonic and submodular. Let

$$\mathcal{I} = \{\mathcal{A}' \subseteq \mathcal{V}' : |\mathcal{A}'| \leq m \wedge \big(\nexists s, i \neq j : (s,i) \in \mathcal{A}' \wedge (s,j) \in \mathcal{A}'\big)\},$$

i.e., $\mathcal{I}$ is the collection of subsets $\mathcal{A}' \subseteq \mathcal{V}'$ that do not contain two pairs (s,i) and (s,j) of elements for which $i \neq j$. It can be shown that $\mathcal{I}$ form independent sets of a matroid (*c.f.*, Fisher et al. [1978]). Note that there is a one-to-one correspondence between sets $\mathcal{A}'$ and feasible solutions $\mathcal{A}_1, \dots, \mathcal{A}_k$ to the SPASS Problem (5.3.1), and furthermore, the corresponding solutions have the same value. Hence, the SPASS problem is equivalent to solving

$$\mathcal{A}^* = \operatorname*{argmax}_{\mathcal{A}' \in \mathcal{I}} F'(\mathcal{A}').$$

As Fisher et al. [1978] proved, the greedy algorithm GAPS is guaranteed to obtain a solution that has at least 1/2 of the optimal value. □

Proof of Lemma 5.5.2. Consider an optimal allocation $\mathcal{T}_1, \dots, \mathcal{T}_m$. Let $\mathcal{B}_{opt}$ be the set $\mathcal{B}_{opt} = \{i : \mathcal{T}_i \text{ contains a big element}\}$. Throw away all buckets (and elements) $\mathcal{B}_{opt}$. Now, in order to achieve score c, the optimal solution has to fill $m - |\mathcal{B}_{opt}|$ buckets with small elements (even from a reduced set of small elements, those not thrown away) and still achieve score c on each of those buckets. This solution is in fact an optimal solution achieving score c on the new problem instance (since we will use at least as many big elements and throw away at least as many buckets). □

5.10.2 Proof of Lemma 5.5.3

Proof. Suppose $\mathcal{A}_i$ is a bucket for which $F_c(\mathcal{A}_i) \geq 3\beta c$. Now, $\mathcal{A}_i = \{a_1, \dots, a_l\}$, and $F_c(\{a_i\}) \leq \beta c$. Choose ℓ such that $F_c(\{a_1, \dots, a_{\ell-1}\}) < \beta c$ and $F_c(\{a_1, \dots, a_\ell\}) \geq \beta c$. Let $\Delta = \{a_1, \dots, a_\ell\}$.

Due to monotonicity $F_c(\mathcal{A}_j \cup \Delta) \geq F_c(\Delta) \geq \beta c$. It remains to show that $F_c(\mathcal{A}_i \setminus \Delta) \geq F_c(\mathcal{A}_i) - 2\beta c$. Suppose that $F_c(\mathcal{A}_i \setminus \Delta) < F_c(\mathcal{A}_i) - 2\beta c$. Let $\mathcal{B} = \mathcal{A}_i \setminus \Delta$. Then

$$F_c(\mathcal{B} \cup \Delta) - F_c(\mathcal{B}) > 2\beta c.$$

But

$$F_c(\Delta) \leq F_c(\{a_1, \dots, a_{\ell-1}\}) + F_c(\{a_\ell\}) < 2\beta c,$$

due to submodularity of F_c, and the fact that a_ℓ is a small element. Hence

$$F_c(\mathcal{B} \cup \Delta) - F_c(\mathcal{B}) > F_c(\Delta) - F_c(\emptyset),$$

i.e., adding Δ to $\mathcal{B}$ helps more than adding Δ to the empty set, contradicting submodularity of F_c. □

Proof of Lemma 5.5.4. To simplify notation, w.l.o.g. let us assume that $c = 1$. Since the optimal balanced performance for $F_c = F_1$ is 1, the optimal average-case performance for F_1 is 1 as well. The GAPS algorithm obtains an allocation $\mathcal{A}$ that is a fraction α of optimal. Hence, it holds that $\sum_i F_1(\mathcal{A}_i) \geq \alpha k$. We call $\sum_i F_1(\mathcal{A}_i)$ the "mass" of the allocation $\mathcal{A}$.

How many unsatisfied buckets can there maximally be? Let γ denote the fraction of unsatisfied buckets. We know that

$$k\gamma\beta + k(1-\gamma) \geq \alpha k,$$

since the maximal γ is achieved if all the satisfied buckets are completely full (containing mass $k(1-\gamma)$), and the unsatisfied buckets are as full as possible without being satisfied (hence containing mass less than $k\gamma\beta$). Hence it follows that

$$\gamma \leq \frac{1-\alpha}{1-\beta}.$$

Now consider the mass R distributed over the satisfied buckets. We know that

$$R \geq \alpha k - \gamma k \beta \geq k\frac{\alpha(1-\beta) - \beta(1-\alpha)}{1-\beta},$$

and the worst case is assumed under equality.

The first reallocation move is possible if

$$\frac{R}{k(1-\gamma)} \geq 3\beta,$$

since, if the average remaining mass over all $(1-\gamma)k$ satisfied buckets is 3β, then there must be at least one bucket to which the move can be applied. Since each reallocation move reduces the mass R by at most 2β (as proved by Lemma 5.5.3), and since we need γk moves to fill all unsatisfied buckets, it suffices to require that

$$\begin{aligned}&\frac{R - 2\gamma k \beta}{k(1-\gamma)} \geq 3\beta\\ \Leftrightarrow &R - 2\gamma k\beta \geq 3\beta k - 3\beta\gamma k\\ \Leftrightarrow &R \geq 3\beta k - \beta\gamma k\end{aligned}$$

Hence, a sufficient condition for β such that enough moves can be performed to fill all unsatisfied buckets is

$$\begin{aligned}&\frac{\alpha(1-\beta) - \beta(1-\alpha)}{1-\beta} \geq 3\beta - \beta\frac{1-\alpha}{1-\beta}\\ \Leftarrow &3\beta^2 + (-3-\alpha)\beta + \alpha \geq 0\\ \Leftarrow &\beta \leq \alpha/3,\end{aligned}$$

by solving the quadratic equation for β and ignoring the infeasible solutions $\beta \geq 1$. Now, since β is going to be our approximation factor, we want to maximize β subject to the above constraint, and hence choose $\beta = \alpha/3$. □

5.10.3 Proof of Theorem 5.5.1

Proof. The proof immediately follows from the analysis in Section 5.5.2. For the running time, notice that, in each binary search iteration, the greedy algorithm requires at most kmn function evaluations, and the reallocation step requires at most $k^2m \leq kmn$ evaluations. The binary search terminates after $\mathcal{O}(1 + \log_2 F(\mathcal{V}))$ iterations, assuming integrality of F. □

5.10.4 Proof of Theorem 5.5.5

Proof. Let $\mathcal{A}_i$ be the candidate solution for time slot i, and $\mathcal{B}_i = \{b_1, \ldots, b_{n_i}\}$ an optimal solution for time slot i. Due to monotonicity and submodularity, it holds that

$$F(\mathcal{A}_i) \leq F(\mathcal{A}_i \cup \mathcal{B}_i) \leq F(\mathcal{A}_i) + \sum_{j=1}^{n_i} \delta_{i,b_j}.$$

Hence, the optimal value of Problem (5.3.2) is upper bounded by the optimal solution to the following integer program:

$$\begin{aligned}&\max_{\lambda_{i,s},c} c \text{ s.t.}\\ &c \leq F(\mathcal{A}_i) + \sum_s \lambda_{i,s}\delta_{i,s} \text{ for all } i\\ &\sum_i \lambda_{i,s} \leq 1 \text{ for all } s \text{ and } \sum_{i,s} \lambda_{i,s} \leq m \text{ and } \lambda_{i,s} \in \{0,1\},\end{aligned}$$

since any integer solution $\lambda_{i,s}$ corresponds to a possible feasible partition $\mathcal{B}'_1, \ldots, \mathcal{B}'_k$. The linear program in Theorem 5.5.5 is the linear programming relaxation to the above integer program. $\square$

5.10.5 Proof Sketch of Theorem 5.6.1

```
Algorithm mcGAPS (F, B, V, k, m, c, λ)
A_t ← ∅ for all t; A ← ∅;
for i = 1 to m do
  foreach s ∈ V \ A, 1 ≤ t ≤ k do
    δ_{t,s} ← λF_c(A_t ∪ {s}) + (1 − λ)F(A ∪ B ∪ {s});
  (t*, s*) ← argmax_{t,s} δ_{t,s};
  A_{t*} ← A_{t*} ∪ {s*}; A ← A ∪ {s*};
```

Algorithm 3: The greedy average-case placement and scheduling (mcGAPS) algorithm.

Proof. We will modify eSPASS in the following way. The modified algorithm, mcSPASS (see the pseudo code in Algorithm 4), will "guess" (binary search for) the value c^* attained by the optimal solution $\mathcal{A}^*$. It will then guess (search for) the number ℓ of large elements used in the optimal solution, where we redefine "large" as $F(\{s\}) \geq \frac{c^*}{8}$. For such a guess, mcSPASS will first greedily select the ℓ large elements (according to F), giving a set $\mathcal{A}_G$. It will then set these large elements aside, and continue on the small elements. Define the function

$$G(\mathcal{A}_1, \ldots, \mathcal{A}_k) = (1-\lambda)(F(\mathcal{A}_G \cup \mathcal{A}) - F(\mathcal{A}_G)) + \frac{\lambda}{m-\ell} \sum_i \min\{F(\mathcal{A}_i), c\}$$

where $\mathcal{A} = \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k$. mcSPASS will greedily maximize G on the partition matroid (similarly to using GAPS). Suppose c^* is the optimal value $c^* = \widehat{F}_\lambda(\mathcal{A}^*)$. Then the greedy procedure will find a solution $\mathcal{A}'$ such that $G(\mathcal{A}') + (1-\lambda)F(\mathcal{A}_G) \geq \frac{1}{2}c^*$ (since greedy selection of big elements followed by greedy selection of small elements amounts to the "local" greedy optimization over a partition matroid as analyzed by Fisher et al. [1978]).

Now, at least one of $(1-\lambda)F(\mathcal{A}_G \cup \mathcal{A}') \geq c^*/8$, or $\frac{\lambda}{m-\ell}\sum_i F(\mathcal{A}'_i) \geq \frac{3c^*}{8}$, since $G(\mathcal{A}') + (1-\lambda)F(\mathcal{A}_G) = (1-\lambda)F(\mathcal{A}_G \cup \mathcal{A}') + \lambda\frac{1}{l-m}\sum_i F(\mathcal{A}'_i)$. In former case we do not need to reallocate and set $\mathcal{A}_R = \mathcal{A}'$. In latter case, we use the reallocation procedure, and arrive at a solution $\mathcal{A}_R$ where all buckets are satisfied (since $\mathcal{A}'$ contains only small elements), i.e., $\min_i F(\mathcal{A}_{R,i}) \geq \frac{3c^*}{3\cdot 8} = c^*/8$.

Now, $\mathcal{A}_R \cup \mathcal{A}_G$ is a feasible solution to the multicriterion SPASS problem, with

$$\begin{aligned}\widehat{F}_\lambda(\mathcal{A}_R \cup \mathcal{A}_G) &= (1-\lambda)F(\mathcal{A}_G) + G(\mathcal{A}_R)\\ &= (1-\lambda)F(\mathcal{A}_G \cup \mathcal{A}_R) + \lambda \min_i F(\mathcal{A}_{R,i}) \geq c^*/8.\end{aligned}$$

$\square$

Algorithm MCSPASS (F, $\mathcal{V}$, k, m, ε, λ)
$c_{\min} \leftarrow 0$; $c_{\max} \leftarrow F(\mathcal{V})$; $\beta \leftarrow 1/8$;
while $c_{\max} - c_{\min} \geq \varepsilon$ **do**
  for $\ell = 0$ **to** k **do**
    $c \leftarrow (c_{\max} + c_{\min})/2$;
1    $\mathcal{B} \leftarrow \{s \in \mathcal{V} : F_c(\{s\}) \geq \beta c\}$;
    if $|\mathcal{B}| < \ell$ **then break**;
    $\mathcal{A}_{big} \leftarrow \emptyset$;
    for $i = 1$ **to** ℓ **do**
      $\mathcal{A}_{big} \leftarrow \text{argmax}_{s \in \mathcal{B} \setminus \mathcal{A}_{big}} F(\mathcal{A}_{big} \cup \{s\})$;
      $\mathcal{A}_{k-i+1} \leftarrow \{s\}$;
    $k' \leftarrow k - \ell$;
    if $k' = 0$ **then** $\mathcal{A}_{best,\ell} \leftarrow (\mathcal{A}_1, \ldots, \mathcal{A}_k)$;
     continue;
    $\mathcal{V}' \leftarrow \mathcal{V} \setminus \mathcal{A}_{big}$; $m' \leftarrow m - \ell$;
2    $\mathcal{A}_{1:k'} \leftarrow$ MCGAPS $(F, \mathcal{A}_{big}, \mathcal{V}', k', m', c, \lambda)$;
3   **if** $\sum_t F(\mathcal{A}_t) < k'c/2$ **then** $c_{\max} \leftarrow c$; **continue**;
    else
4   **while** $\exists i, j \leq k'$: $F_c(\mathcal{A}_j) \leq \beta c$, $F_c(\mathcal{A}_i) \geq 3\beta c$ **do**
        foreach $s \in \mathcal{A}_i$ **do**
          $\mathcal{A}_j \leftarrow \mathcal{A}_j \cup \{s\}$; $\mathcal{A}_i \leftarrow \mathcal{A}_i \setminus \{s\}$;
          if $F_c(\mathcal{A}_j) \geq \beta c$ **then break**;
      $\mathcal{A}_{best,\ell} \leftarrow (\mathcal{A}_1, \ldots, \mathcal{A}_k)$;
  $\ell^* \leftarrow \text{argmax}_\ell \widehat{F}_\lambda(\mathcal{A}_{best,\ell})$;
  $c_{\min} \leftarrow c$; $\mathcal{A}_{best} \leftarrow \mathcal{A}_{best,\ell^*}$;

Algorithm 4: The MCSPASS algorithm for simultaneously optimizing scheduled and high-density performance.

Chapter 6

Estimating Traffic Statistics in a Data Communication-Constrained Setting

6.1 Introduction

Chapter 2 discussed two forms of inferring traffic variables for urban streets: mobile and fixed smart sensors. In both cases, constraints on power consumption impose communication constraints for the sensing devices. One way of decreasing communication requirements is by using local processing to process the inference. For example, the estimate of the mean value of repeated measurements from a group of sensors can be obtained by averaging the local mean estimates of each sensor.

Independently of how a measurement is made, we argued that statistical *quantiles* are more meaningful than average estimators for characterizing the statistical process of the dynamic variables involved. We are interested in creating order estimators that can reduce data gathering requirements by using local processing. Order estimators are nonlinear and therefore appropriate protocols for combining local estimates need to be defined.

There are two principal usage scenarios for estimating quantiles: for travel times from individual vehicle measurements (*mobile sensing*) and for road section variables from embedded smart sensors (*fixed smart sensing*).

To explain both scenarios we divide time into blocks of fixed size, and assume that the statistical behavior of traffic is stationary for N consecutive blocks. In *mobile sensing*, we assume a service provider is interested in offering real-time estimates of link travel time quantiles. Groups of M vehicles cross a link during time n. In a real scenario, these are the group observable vehicles due to privacy constraints. Each vehicle driver already has a current estimate of the quantile of interest θ_n provided to him by the travel time service. Driver m also *independently* experiences travel time $X_{n+1}(m)$, but his device has limited power therefore he is not willing to send $X_{n+1}(m)$ to the provider as he gains no future benefit from updates to θ_n. How can the service provider update θ_n to reflect new knowledge under this constraint? Notice that in this case the communication constraint is only in the direction from the driver (i.e. *sensor*) to the service provider (i.e. *fusion center*).

In the *fixed smart sensing* scenario, M sensors are deployed in the link to measure variables such as traffic flow. Each sensor makes an independent measurement $X_{n+1}(m)$ at time n, and both smart sensors and the fusion center objective is to achieve an estimate of the quantile at time n. Sensors are battery operated and therefore communication is constrained. Unlike in the mobile sensing scenario, the constraint is in both directions, as usage of the radio to both send and receive information is costly. Is there a communication efficient protocol for message exchange that allows sensors and fusion center to obtain updated quantile estimates for each time n?

In this Chapter we investigate both these questions and provide efficient protocols to achieve these goals. The protocols are, in a strictly defined sense, as efficient as in a situation without any communication constraints showing that local computation can indeed provide a mechanism to overcome data transfer constraints.

The remainder of the Chapter is organized as follows. We begin in Section 6.2 by discussing existing literature on methods for decentralized inference. Section 6.3 describes the required background on quantile estimation, and optimal rates in the centralized setting. We then describe two algorithms for solving the corresponding decentralized version, and provide an asymptotic characterization of their performance. These theoretical results are complemented with empirical simulations. Section 6.6 contains the proofs of our main results, and we conclude in Section 6.5 with a discussion.

6.2 Related Work

Whereas classical statistical inference is performed in a centralized manner, many modern scientific problems and engineering systems are inherently *decentralized*: data are distributed, and cannot be aggregated due to various forms of communication constraints. An important example of such a decentralized system is a sensor network [Chong and Kumar, 2003]: a set of spatially-distributed sensors collect data about the environmental state (e.g., temperature, humidity or light). Typically, these networks are based on ad hoc deployments, in which the individual sensors are low-cost, and must operate under very severe power constraints (e.g., limited battery life). In statistical terms, such communication constraints imply that the individual sensors cannot transmit the raw data; rather, they must compress or quantize the data—for instance, by reducing a continuous-valued observation to a single bit—and transmit only this compressed representation back to the fusion center.

By now, there is a rich literature in both information theory and statistical signal processing on problems of decentralized statistical inference. A number of researchers, dating back to the seminal paper of Tenney and Sandell [Tenney and Sandell, 1981], have studied the problem of hypothesis testing under communication-constraints; see the survey papers [Tsitsiklis, 1993; Veeravalli et al., 1993; Blum et al., 1997; Viswanathan and Varshney, 1997; Chamberland and Veeravalli, 2004] and references therein for overviews of this line of work. The hypothesis-testing problem has also been studied in the information theory community, where the analysis is asymptotic and Shannon-theoretic in nature [Amari and Han, 1989; Han and Kobayashi, 1989]. A parallel line of work deals with problem of decentralized estimation. Work in signal processing typically formulates it as a quantizer design problem and considers finite sample behavior [Ayanoglu, 1990; Gubner, 1993]; in contrast, the information-theoretic approach is asymptotic in nature, based on rate-distortion theory [Zhang and Berger, 1988; Han and Amari, 1998]. In much of the literature on decentralized statistical inference, it is assumed that the underlying distributions are known with a specified parametric form (e.g., Gaussian). More recent work has addressed non-parametric and data-driven formulations of these problems, in which the decision-maker is simply provided samples from the unknown distribution [Nguyen et al., 2005; Luo, 2005; Han et al., 1990]. For instance, Nguyen et al. [Nguyen et al., 2005] established statistical consistency for non-parametric approaches to decentralized hypothesis testing based on reproducing kernel Hilbert spaces. Luo [Luo, 2005] analyzed a non-parametric formulation of decentralized mean estimation, in which a fixed but unknown parameter is corrupted by noise with bounded support but otherwise arbitrary distribution, and shown that decentralized approaches can achieve error rates that are order-optimal with respect to the centralized optimum.

This Chapter addresses a different problem in decentralized non-parametric inference—namely, that of estimating an arbitrary quantile of an unknown distribution. Since there exists no unbiased estimator based on a single sample, we consider the performance of a network of m sensors, each of which collects total of n observations in a sequential manner. Our analysis treats the standard fusion-based architecture, in which each of the m sensors transmits information to the fusion center via a communication-constrained channel. More concretely, at each of the n observation rounds, each sensor is allowed to transmit a single bit to the fusion center, which in turn is permitted to send some number k bits of feedback. For a decentralized protocol with $k = \log(m)$ bits of feedback, we prove that the algorithm achieves the order-optimal rate of the best centralized method (i.e., one with access to the full collection of raw data). We also consider a protocol that permits only a single bit of feedback, and establish that it achieves the same rate. This single-bit protocol is advantageous in that, with for a fixed target mean-squared error of the quantile estimate, it yields longer sensor lifetimes than either the centralized or full feedback protocols.

6.3 Problem Set-up and Decentralized Algorithms

In this section, we begin with some background material on (centralized) quantile estimation, before introducing our decentralized algorithms, and stating our main theoretical results.

6.3.1 Centralized Quantile Estimation

We begin with the classical background on the problem of quantile estimation, and refer the interested reader to Serfling [Serfling, 1980] for further details. Given a real-valued random variable X, let $F(x) := \mathbb{P}[X \leq x]$ be its cumulative distribution function (CDF), which is non-decreasing and right-continuous. For any $0 < \alpha < 1$, the α^{th}-quantile of X is defined as $F^{-1}(\alpha) = \theta(\alpha) := \inf \{x \in \mathbb{R} \mid F(x) \geq \alpha\}$. Moreover, if F is continuous at α, then we have $\alpha = F(\theta(\alpha))$. As a particular example, for $\alpha = 0.5$, the associated quantile is simply the median.

Now suppose that for a fixed level $\alpha^* \in (0,1)$, we wish to estimate the quantile $\theta^* = \theta(\alpha^*)$. Rather than impose a particular parameterized form on F, we work in a non-parametric setting, in which we assume only that the distribution function F is differentiable, so that X has the density function $p_X(x) = F'(x)$ (w.r.t Lebesgue measure), and moreover that $p_X(x) > 0$ for all $x \in \mathbb{R}$. In this setting, a standard estimator for θ^* is the *sample quantile* $\xi_N(\alpha^*) := F_N^{-1}(\alpha^*)$ where F_N denotes the empirical distribution function based on i.i.d. samples $(X_1, \ldots, X_N)$. Under the conditions given above, it can be shown [Serfling, 1980] that $\xi_N(\alpha^*)$ is strongly consistent for θ^* (i.e., $\xi_N \stackrel{a.s.}{\to} \theta^*$), and moreover that asymptotic normality holds

$$\sqrt{N}(\xi_N - \theta^*) \quad \stackrel{d}{\to} \quad \mathcal{N}\left(0, \frac{\alpha^*(1-\alpha^*)}{p_X^2(\theta^*)}\right), \tag{6.3.1}$$

so that the asymptotic MSE decreases as $\mathcal{O}(1/N)$, where N is the total number of samples. Although this $1/N$ rate is optimal, the precise form of the asymptotic variance (6.3.1) need not be in general; see Zielinski [Zielinski, 2004] for in-depth discussion of the optimal asymptotic variances that can be obtained with variants of this basic estimator under different conditions.

6.3.2 Distributed Quantile Estimation

We consider the standard network architecture illustrated in Figure 6.1. There are m sensors, each of which has a dedicated two-way link to a fusion center. We assume that each sensor $i \in \{1, \ldots, m\}$ collects independent samples $X(i)$ of the random variable $X \in \mathbb{R}$ with distribution function $F(\theta) := \mathbb{P}[X \leq \theta]$. We consider a sequential version of the quantile estimation problem, in which sensor i receives measurements $X_n(i)$ at time steps $n = 0, 1, 2, \ldots$, and the fusion center forms an estimate θ_n of the quantile. The key condition—giving rise to the decentralized nature of the problem—is that communication between each sensor and the central processor is constrained, so that the sensor cannot simply relay its measurement $X(i)$ to the central location, but rather must perform local computation, and then transmit a summary statistic to the fusion center. More concretely, we impose the following restrictions on the protocol. First, at each time step $n = 0, 1, 2, \ldots$, each sensor $i = 1, \ldots, m$ can transmit a single bit $Y_n(i)$ to the fusion center. Second, the fusion center can broadcast k bits back to the sensor nodes at each time step. We analyze two distinct protocols, depending on whether $k = \log(m)$ or $k = 1$.

6.3.3 Protocol specification

For each protocol, all sensors are initialized with some fixed θ_0. The algorithms are specified in terms of a constant $K > 0$ and step sizes $\epsilon_n > 0$ that satisfy the conditions

$$\sum_{n=0}^{\infty} \epsilon_n = \infty \qquad \text{and} \qquad \sum_{n=0}^{\infty} \epsilon_n^2 < \infty. \tag{6.3.2}$$

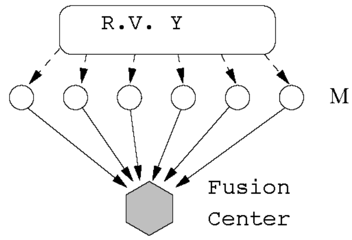


Figure 6.1. Sensor network for quantile estimation with m sensors. Each sensor is permitted to transmit a 1-bit message to the fusion center; in turn, the fusion center is permitted to broadcast k bits of feedback.

The first condition ensures infinite travel (i.e., that the sequence θ_n can reach θ^* from any starting condition), whereas the second condition (which implies that $\epsilon_n \to 0$) is required for variance reduction. A standard choice satisfying these conditions—and the one that we assume herein—is $\epsilon_n = 1/n$. With this set-up, the $\log(m)$-bit scheme consists of the steps given in Table 6.1. Although the most straightforward feedback protocol is to

Algorithm: Decentralized quantile estimation with $\log(m)$-bit feedback
Given $K > 0$ and variable step sizes $\epsilon_n > 0$:

(a) *Local decision:* each sensor computes the binary decision

$$Y_{n+1}(i) \equiv Y_{n+1}(i; \theta_n) \; := \; \mathbb{I}(X_{n+1}(i) \leq \theta_n), \tag{6.3.3}$$

and transmits it to the fusion center.

(b) *Parameter update:* the fusion center updates its current estimate θ_{n+1} of the quantile parameter as follows:

$$\theta_{n+1} \;\; = \;\; \theta_n + \epsilon_n K \left(\alpha^* - \frac{\sum_{i=1}^m Y_{n+1}(i)}{m} \right) \tag{6.3.4}$$

(c) *Feedback:* the fusion broadcasts the m received bits $\{Y_{n+1}(1), \ldots, Y_{n+1}(m)\}$ back to the sensors. Each sensor can then compute the updated parameter θ_{n+1}.

Table 6.1: Description of the $\log(m)$-bf algorithm.

broadcast back the m received bits $\{Y_{n+1}(1), \ldots, Y_{n+1}(m)\}$, as described in step (c), in fact it suffices to transmit only the $\log(m)$ bits required to perfectly describe the binomial random variable $\sum_{i=1}^m Y_{n+1}(i)$ in order to update θ_n. In either case, after the feedback step, each sensor knows the value of the sum $\sum_{i=1}^m Y_{n+1}(i)$, which (in conjunction with knowledge of m, α^* and ϵ_n) allow it to compute the updated parameter θ_{n+1}. Finally, knowledge of θ_{n+1} allows each sensor to then compute the local decision (6.3.3) in the following round.

The 1-bit feedback scheme detailed in Table 6.2 is similar, except that it requires broadcasting only a single bit (Z_{n+1}), and involves an extra step size parameter K_m, which is specified in the statement of Theorem 6.3.2. After the feedback step of the 1-bf algorithm,

Algorithm: Decentralized quantile estimation with 1-bit feedback
Given $K_m > 0$ (possibly depending on number of sensors m) and variable step sizes $\epsilon_n > 0$:

(a) *Local decision:* each sensor computes the binary decision

$$Y_{n+1}(i) \;=\; \mathbb{I}(X_{n+1}(i) \leq \theta_n) \tag{6.3.5}$$

and transmits it to the fusion center.

(b) *Aggregate decision and parameter update:* The fusion center computes the aggregate decision

$$Z_{n+1} \;=\; \mathbb{I}\left(\frac{\sum_{i=1}^{m} Y_{n+1}(i)}{m} \leq \alpha^*\right), \tag{6.3.6}$$

and uses it update the parameter according to

$$\theta_{n+1} \;=\; \theta_n + \epsilon_n K_m \left(Z_{n+1} - \beta\right) \tag{6.3.7}$$

where the constant β is chosen as

$$\beta \;=\; \sum_{i=0}^{\lfloor m\alpha^* \rfloor} \binom{m}{i} (\alpha^*)^i \left(1 - \alpha^*\right)^{m-i}. \tag{6.3.8}$$

(c) *Feedback:* The fusion center broadcasts the aggregate decision Z_{n+1} back to the sensor nodes (one bit of feedback). Each sensor can then compute the updated parameter θ_{n+1}.

Table 6.2: Description of the 1-bf algorithm.

each sensor has knowledge of the aggregate decision Z_{n+1}, which (in conjunction with ϵ_n and the constant β) allow it to compute the updated parameter θ_{n+1}. Knowledge of this parameter suffices to compute the local decision (6.3.5).

6.3.4 Convergence results

We now state our main results on the convergence behavior of these two distributed protocols. In all cases, we assume the step size choice $\epsilon_n = 1/n$. Given fixed $\alpha^* \in (0, 1)$, we use θ^* to denote the α^*-level quantile (i.e., such that $\mathbb{P}(X \leq \theta^*) = \alpha^*$); note that our assumption of a strictly positive density guarantees that θ^* is unique.

Theorem 6.3.1 (m-bit feedback)**.** *For any $\alpha^* \in (0, 1)$, consider a random sequence $\{\theta_n\}$ generated by the m-bit feedback protocol. Then*
(a) For all initial conditions θ_0, the sequence θ_n converges almost surely to the α^*-quantile θ^*.

(b) Moreover, if the constant K is chosen to satisfy $p_X(\theta^*)\,K > \frac{1}{2}$, then

$$\sqrt{n}\,(\theta_n - \theta^*) \xrightarrow{d} \mathcal{N}\left(0, \frac{K^2\,\alpha^*\,(1-\alpha^*)}{\left[2Kp_X(\theta^*) - 1\right]}\,\frac{1}{m}\right), \tag{6.3.9}$$

so that the asymptotic MSE is $O(\frac{1}{mn})$.

Remarks: After n steps of this decentralized protocol, a total of $N = nm$ observations have been made, so that our discussion in Section 6.3.1 dictates (see equation (6.3.1)) that the optimal asymptotic MSE is $O(\frac{1}{nm})$. Interestingly, then, the m-bit feedback decentralized protocol is order-optimal with respect to the centralized gold standard.

Before stating the analogous result for the 1-bit feedback protocol, we begin by introducing some useful notation. First, we define for any fixed $\theta \in \mathbb{R}$ the random variable

$$\bar{Y}(\theta) := \frac{1}{m}\sum_{i=1}^{m} Y(i;\theta) \;=\; \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(X(i) \leq \theta).$$

Note that for each fixed θ, the distribution of $\bar{Y}(\theta)$ is binomial with parameters m and $F(\theta)$. It is convenient to define the function

$$G_m(r,y) \;:=\; \sum_{i=0}^{\lfloor my \rfloor} \binom{m}{i} r^i\,(1-r)^{m-i}, \tag{6.3.10}$$

with domain $(r,y) \in [0,1] \times [0,1]$. With this notation, we have

$$\mathbb{P}(\bar{Y}(\theta) \leq y) \;=\; G_m(F(\theta), y).$$

Again, we fix an arbitrary $\alpha^* \in (0,1)$ and let θ^* be the associated α^*-quantile satisfying $\mathbb{P}(X \leq \theta^*) = \alpha^*$.

Theorem 6.3.2 (1-bit feedback)**.** *Given a random sequence $\{\theta_n\}$ generated by the 1-bit feedback protocol, we have*

(a) For any initial condition, the sequence $\theta_n \xrightarrow{a.s.} \theta^*$.

(b) Suppose that the step size K_m is chosen such that $K_m > \frac{\sqrt{2\pi\alpha^*(1-\alpha^*)}}{2p_X(\theta^*)\sqrt{m}}$, or equivalently such that

$$\gamma_m(\theta^*) \;:=\; K_m\left|\frac{\partial G_m}{\partial r}(r;\alpha^*)\big|_{r=\alpha^*}\right| p_X(\theta^*) \;>\; \frac{1}{2}, \tag{6.3.11}$$

then

$$\sqrt{n}\,(\theta_n - \theta^*) \xrightarrow{d} \mathcal{N}\left(0, \frac{K_m^2 G_m(\alpha^*,\theta^*)\left[1 - G_m(\alpha^*,\theta^*)\right]}{2\gamma_m(\theta^*) - 1}\right) \tag{6.3.12}$$

(c) If we choose a constant step size *$K_m = K$, then as $n \to \infty$, the asymptotic variance*

behaves as

$$\left[\frac{K^2\sqrt{2\pi\alpha^*(1-\alpha^*)}}{8Kp_X(\theta^*)\sqrt{m}-4\sqrt{2\pi\alpha^*(1-\alpha^*)}}\right], \tag{6.3.13}$$

so that the asymptotic MSE is $O\left(\frac{1}{n\sqrt{m}}\right)$.

(d) *If we choose a* decaying step size $K_m = \frac{K}{\sqrt{m}}$, *then*

$$\frac{1}{m}\left[\frac{K^2\sqrt{2\pi\alpha^*(1-\alpha^*)}}{8Kp_X(\theta^*)-4\sqrt{2\pi\alpha^*(1-\alpha^*)}}\right], \tag{6.3.14}$$

so that the asymptotic MSE is $O\left(\frac{1}{nm}\right)$.

6.3.5 Comparative Analysis

It is interesting to compare the performance of each proposed decentralized algorithm to the centralized performance. Considering first the m-bf scheme, suppose that we set $K = 1/p_X(\theta^*)$. Using the formula (6.3.9) from Theorem 6.3.1, we obtain that the asymptotic variance of the m-bf scheme with this choice of K is given by $\frac{\alpha^*(1-\alpha^*)}{p_X^2(\theta^*)}\frac{1}{mn}$, thus matching the asymptotics of the centralized quantile estimator (6.3.1). In fact, it can be shown that the choice $K = 1/p_X(\theta^*)$ is optimal in the sense of minimizing the asymptotic variance for our scheme, when K is constrained by the stability criterion in Theorem 6.3.1. In practice, however, the value $p_X(\theta^*)$ is typically not known, so that it may not be possible to implement exactly this scheme. An interesting question is whether an adaptive scheme could be used to estimate $p_X(\theta^*)$ (and hence the optimal K simultaneously), thereby achieving this optimal asymptotic variance. We leave this question open as an interesting direction for future work.

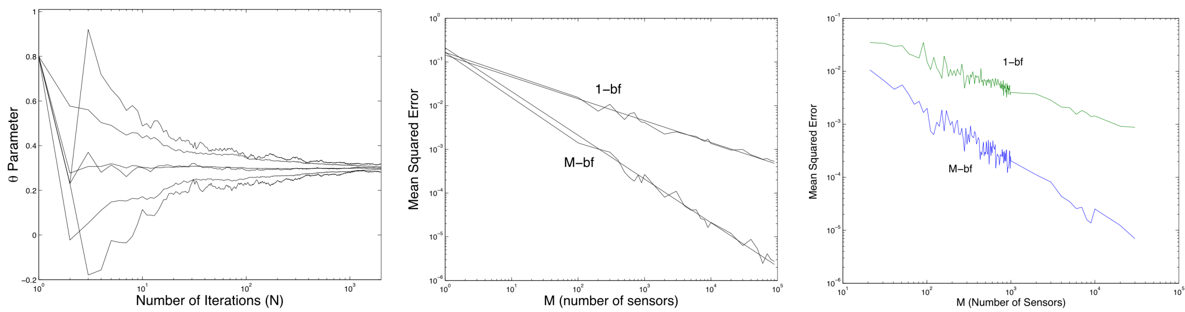
Turning now to the algorithm 1-bf, if we make the substitution $\bar{K} = K/\sqrt{2\pi\alpha^*(1-\alpha^*)}$ in equation (6.3.14), then we obtain the asymptotic variance

$$\frac{\pi}{2}\,\frac{\bar{K}^2\,\alpha^*\,(1-\alpha^*)}{\left[2\bar{K}p_X(\theta^*)-1\right]}\,\frac{1}{m}. \tag{6.3.15}$$

Since the stability criterion is the same as that for m-bf, the optimal choice is $\bar{K} = 1/p_X(\theta^*)$. Consequently, while the $(1/[mn])$ rate is the same as both the centralized and decentralized m-bf protocols, the pre-factor for the 1-bf algorithm is $\frac{\pi}{2} \approx 1.57$ times larger than the optimized m-bf scheme. However, despite this loss in the pre-factor, the 1-bf protocol has substantial advantages over the m-bf; in particular, the network lifetime scales as $O(m)$ compared to $\mathcal{O}(m/\log(m))$ for the $\log(m)$-bf scheme.

6.3.6 Simulation example

We now provide some simulation results in order to illustrate the two decentralized protocols, and the agreement between theory and practice. In particular, we consider the quantile estimation problem when the underlying distribution (which, of course, is unknown



(a) (b) (c)

Figure 6.2. Convergence of θ_n to θ^* with $m = 11$ nodes, and quantile level $\alpha^* = 0.3$. (b) Log-log plots of the variance against m for both algorithms ($\log(m)$-bf and 1-bf) with constant step sizes, and theoretically-predicted rate. (b) Log-log plots of the variance against m for $\log(m)$-bf and 1-bf algorithms with constant step size. (c) Log-log plots of $\log(m)$-bf with constant step size versus 1-bf algorithm with decaying step size.

to the algorithm) is uniform on $[0, 1]$ random. In this case, we have $p_X(x) = 1$ uniformly for all $x \in [0, 1]$, so that taking the constant $K = 1$ ensures that the stability conditions in both Theorem 6.3.1 and 6.3.2 are satisfied. We simulate the behavior of both algorithms for $\alpha^* = 0.3$ over a range of choices for the network size m. Figure 6.2(a) illustrates several sample paths of m-bit feedback protocol, showing the convergence to the correct θ^*.

For comparison to our theory, we measure the empirical variance by averaging the error $\hat{e}_n = \sqrt{n}(\theta_n - \theta^*)$ over $L = 20$ runs. The normalization by $\sqrt{n}$ is used to isolate the effect of increasing m, the number of nodes in the network. We estimate the variance by running algorithm for $n = 2000$ steps, and computing the empirical variance of $\hat{e}_n$ for time steps $n = 1800$ through to $n = 2000$. Figure 6.2(b) shows these empirically computed variances, and a comparison to the theoretical predictions of Theorems 6.3.1 and 6.3.2 for constant step size; note the excellent agreement between theory and practice. Panel (c) shows the comparison between the $\log(m)$-bf algorithm, and the 1-bf algorithm with decaying $1/\sqrt{m}$ step size. Here the asymptotic MSE of both algorithms decays like $1/m$ for $\log m$ up to roughly 500; after this point, our fixed choice of n is insufficient to reveal the asymptotic behavior.

6.4 Some Extensions

In this section we consider some extensions of the algorithms and analysis from the preceding sections, including variations in the number of feedback bits, and the effects of noise.

6.4.1 Different levels of feedback

We first consider the generalization of the preceding analysis to the case when the fusion center communicates some number of bits between 1 and m. The basic idea is to apply a quantizer with 2ℓ levels, corresponding to $\log_2(2\ell)$ bits, on the update of the stochastic gradient algorithm. Note that the extremes $\ell = 1$ and $\ell = 2^{m-1}$ correspond to the previously studied protocols. Given 2ℓ levels, we partition the real line as

$$-\infty \;=\; s_{-\ell} \;<\; s_{-\ell+1} \;<\; \ldots \;<\; s_{\ell-1} \;<\; s_{\ell} = +\infty, \tag{6.4.1}$$

where the remaining breakpoints $\{s_k\}$ are to be specified. With this partition fixed, we define a quantization function $\mathcal{Q}_\ell$

$$\mathcal{Q}_\ell(X) \;\;:=\;\; r_k \qquad \text{if } X \in (s_k, s_{k+1}] \text{ for } k = -\ell, \ldots, \ell-1, \tag{6.4.2}$$

where the 2ℓ quantized values $(r_{-\ell}, \ldots, r_{\ell-1})$ are to be chosen. In the setting of the algorithm to be proposed, the quantizer is applied to binomial random variables X with parameters (m, r). Recall the function $G_m(r, x)$, as defined in equation (6.3.10), corresponding to the probability $\mathbb{P}[X \leq mx]$. Let us define a new function $G_{m,\ell}$, corresponding to the expected value of the quantizer when applied to such a binomial variate, as follows

$$G_{m,\ell}(r, x) \;\;:=\;\; \sum_{k=-\ell}^{\ell-1} r_k \left\{ G_m(r, x - s_k) - G_m(r, x - s_{k+1}) \right\}. \tag{6.4.3}$$

With these definitions, the general $\log_2(2\ell)$ feedback algorithm takes the form shown in Table 6.3.

In order to understand the choice of the offset parameter β defined in equation (6.4.7), we compute the expected value of the quantizer function, when $\theta_n = \theta^*$, as follows

$$\begin{aligned}\mathbb{E}\Big[\mathcal{Q}_\ell\left[\alpha^* - \frac{\sum_{i=1}^m Y_{n+1}(i)}{m}\right] \;\mid\; \theta_n = \theta^*\Big] &= \sum_{k=-\ell}^{\ell-1} r_k \mathbb{P}\left[(\alpha^* - s_{k+1}) < \frac{\bar{Y}(\theta^*)}{m} \leq (\alpha^* - s_k)\right] \\ &= \sum_{k=-\ell}^{\ell-1} r_k \left[G_m(F(\theta^*), \alpha^* - s_k) - G_m(F(\theta^*), \alpha^* - s_{k+1})\right] \\ &= G_{m,\ell}(F(\theta^*), \alpha^*).\end{aligned}$$

The following result, analogous to Theorem 6.3.2, characterizes the behavior of this general protocol:

Theorem 6.4.1 (General feedback scheme)**.** *Given a random sequence $\{\theta_n\}$ generated by the general $\log_2(2\ell)$-bit feedback protocol, there exist choices of partition $\{s_k\}$ and quantization levels $\{r_k\}$ such that:*

(a) For any initial condition, the sequence $\theta_n \xrightarrow{a.s.} \theta^*$.

(b) There exists a choice of decaying step size *(i.e., $K_m \asymp \frac{1}{\sqrt{m}}$) such that the asymptotic variance of the protocol is given by $\frac{\kappa(\alpha^*, \mathcal{Q}_\ell)}{mn}$, where the constant has the form*

$$\kappa(\alpha^*, \mathcal{Q}_\ell) \;\;:=\;\; 2\pi \, \frac{\sum_{k=-\ell}^{\ell-1} r_k^2 \Delta G_m(s_k, s_{k+1}) - \beta^2}{\sum_{k=-\ell}^{\ell-1} r_k \Delta_m(s_k, s_{k+1})}, \tag{6.4.8}$$

Algorithm: Decentralized quantile estimation with $\log_2(2\ell)$-bits feedback
Given $K_m > 0$ (possibly depending on number of sensors m) and variable step sizes $\epsilon_n > 0$:

(a) *Local decision:* each sensor computes the binary decision

$$Y_{n+1}(i) \quad = \quad \mathbb{I}(X_{n+1}(i) \leq \theta_n) \tag{6.4.4}$$

and transmits it to the fusion center.

(b) *Aggregate decision and parameter update:* The fusion center computes the quantized aggregate decision variable

$$Z_{n+1} \quad = \quad \mathcal{Q}_\ell \left[\alpha^* - \frac{\sum_{i=1}^m Y_{n+1}(i)}{m} \right], \tag{6.4.5}$$

and uses it update the parameter according to

$$\theta_{n+1} \quad = \quad \theta_n + \epsilon_n K_m \left(Z_{n+1} - \beta \right) \tag{6.4.6}$$

where the constant β is chosen as

$$\beta \quad := \quad G_{m,\ell}(F(\theta^*), \alpha^*). \tag{6.4.7}$$

(c) *Feedback:* The fusion center broadcasts the aggregate quantized decision Z_{n+1} back to the sensor nodes, using its $\log_2(2\ell)$ bits of feedback. The sensor nodes can then compute the updated parameter θ_{n+1}.

Table 6.3: Description of the general algorithm, with $\log_2(2\ell)$ bits of feedback.

with

$$\begin{aligned}
\Delta G_m(s_k, s_{k+1}) \quad &= \quad G_m(F(\theta^*), \alpha^* - s_k) - G_m(F(\theta^*), \alpha^* - s_{k+1}), \quad \textit{and} \\
\Delta_m(s_k, s_{k+1}) \quad &= \quad \exp\left(-\frac{m s_k^2}{2\alpha^*(1-\alpha^*)} \right) - \exp\left(-\frac{m s_{k+1}^2}{2\alpha^*(1-\alpha^*)} \right).
\end{aligned}$$

We provide a formal proof of Theorem 6.4.1 in Section 6.6. Figure 6.3(a) illustrates how the constant factor κ, as defined in equation (6.4.8) decreases as of levels ℓ in an uniform quantizer is increased. Note

In order to provide comparison with results from the previous section, let us see how the two extreme cases (1 bit and m feedback) can be obtained as special case. For the 1-bit case, the quantizer has $\ell = 1$ levels with breakpoints $s_{-1} = -\infty$, $s_0 = 0$, $s_1 = +\infty$, and quantizer outputs $r_{-1} = 0$ and $r_1 = 1$. By making the appropriate substitutions, we obtain:

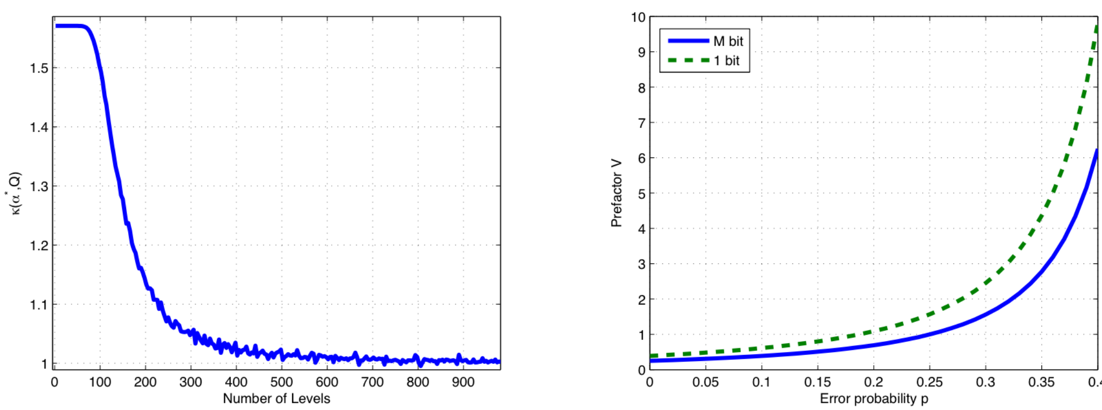
$$\kappa(\alpha^*, \mathcal{Q}_1) \;=\; 2\pi\,\frac{\Delta G_m(s_0, s_1) - \beta^2}{\Delta_m(s_0, s_1)}, \qquad \beta^2 \;=\; G_{m,\ell}(F(\theta^*), \alpha^*)^2,$$
$$\Delta G_m(s_0, s_1) \;=\; G_{m,\ell}(F(\theta^*), \alpha^*) \qquad \text{and} \qquad \Delta_m(s_0, s_1)) \;=\; 1.$$

By applying the central limit theorem, we conclude that

$$\Delta G_m(s_0, s_1) - \beta^2 = G_{m,\ell}(F(\theta^*), \alpha^*)(1 - G_{m,\ell}(F(\theta^*), \alpha^*)) \to 1/4,$$

as established earlier. Thus $\kappa(\alpha^*, \mathcal{Q}_1) \to \pi/2$ as $m \to \infty$, recovering the result of Theorem 6.3.2. Similarly, the results for m-bf can be recovered by setting the parameters

$$\begin{aligned} r_{k-\ell} &= \alpha^* - \frac{k}{m}, \quad \text{for} \quad k = 0, ..., M, \qquad \text{and} \\ s_i &= r_i. \end{aligned} \tag{6.4.10}$$



(a) (b)

Figure 6.3. (a) Plots of the asymptotic variance $\kappa(\alpha^*, \mathcal{Q}_\ell)$ defined in equation (6.4.8) versus the number of levels ℓ in a uniform quantizer, corresponding to $\log_2(2\ell)$ bits of feedback, for a sensor network with $m = 4000$ nodes. The plots show the asymptotic variance rescaled by the centralized gold standard, so that it starts at $\pi/2$ for $\ell = 2$, and decreases towards 1 as ℓ is increased towards $m/2$. (b) Plots of the asymptotic variances $V_m(\epsilon)$ and $V_1(\epsilon)$ defined in equation (6.4.13) as the feedforward noise parameter ϵ is increased from 0 towards $\frac{1}{2}$.

6.4.2 Extensions to noisy links

We now briefly consider the effect of communication noise on our algorithms. There are two types of noise to consider: (a) *feedforward*, meaning noise in the link from sensor node to fusion center, and (b) *feedback*, meaning noise in the feedback link from fusion center to the sensor nodes. Here we show that feedforward noise can be handled in a relatively straightforward way in our algorithmic framework. On the other hand, feedback noise requires a different analysis, as the different sensors may loose synchronicity in their updating procedure. Although a thorough analysis of such asynchronicity is an interesting topic for future research, we note that assuming noiseless feedback is not unreasonable, since the fusion center typically has greater transmission power.

Focusing then on the case of feedforward noise, let us assume that the link between each sensor and the fusion center acts as a binary symmetric channel (BSC) with probability $\epsilon \in [0, \frac{1}{2})$. More precisely, if a bit $x \in \{0, 1\}$ is transmitted, then the received bit y has the (conditional) distribution

$$\mathbb{P}(y \mid x) \quad = \quad \begin{cases} 1-\epsilon & \text{if } x = y \\ \epsilon & \text{if } x \neq y. \end{cases} \tag{6.4.11}$$

With this bit-flipping noise, the updates (both equation (6.3.4) and (6.3.7)) need to be modified so as to correct for the bias introduced by the channel noise. If α^* denotes the desired quantile, then in the presence of BSC(ϵ) noise, both algorithms should be run with the modified parameter

$$\widetilde{\alpha}(\epsilon) \quad := \quad (1-2\epsilon)\alpha^* + \epsilon. \tag{6.4.12}$$

Note that $\widetilde{\alpha}(\epsilon)$ ranges between α^* (for the noiseless case $\epsilon = 0$), to a quantity arbitrarily close to $\frac{1}{2}$, as the channel approaches the extreme of pure noise ($\epsilon = \frac{1}{2}$). The following lemma shows that for all $\epsilon < \frac{1}{2}$, this adjustment (6.4.12) suffices to correct the algorithm. Moreover, it specifies how the resulting asymptotic variance depends on the noise parameter:

Proposition 6.4.1. *Suppose that each of the m feedforward links from sensor to fusion center are modeled as i.i.d. BSC channels with probability $\epsilon \in [0, \frac{1}{2})$. Then the m-bf or 1-bf algorithms, with the adjusted $\widetilde{\alpha}(\epsilon)$, are strongly consistent in computing the α^*-quantile. Moreover, with appropriate step size choices, their asymptotic MSEs scale as $1/(mn)$ with respective pre-factors given by*

$$V_m(\epsilon) \quad := \quad \frac{K^2\, \widetilde{\alpha}(\epsilon)\,(1-\widetilde{\alpha}(\epsilon))}{\left[2K(1-2\epsilon)p_X(\theta^*) - 1\right]} \tag{6.4.13a}$$

$$V_1(\epsilon) \quad := \quad \left[\frac{K^2\sqrt{2\pi\widetilde{\alpha}(\epsilon)(1-\widetilde{\alpha}(\epsilon))}}{8K(1-2\epsilon)p_X(\theta^*) - 4\sqrt{2\pi\widetilde{\alpha}(\epsilon)(1-\widetilde{\alpha}(\epsilon))}}\right]. \tag{6.4.13b}$$

In both cases, the asymptotic MSE is minimal for $\epsilon = 0$.

Proof: If sensor node i transmits a bit $Y_{n+1}(i)$ at round $n+1$, then the fusion center receives the random variable

$$\widetilde{Y}_{n+1}(i) \quad = \quad Y_{n+1}(i) \oplus W_{n+1},$$

where W_{n+1} is Bernoulli with parameter ϵ, and $\oplus$ denotes addition modulo two. Since W_{n+1} is independent of the transmitted bit (which is Bernoulli with parameter $F(\theta_n)$), the received value $\widetilde{Y}_{n+1}(i)$ is also Bernoulli, with parameter

$$\epsilon * F(\theta_n) \;=\; \epsilon\,(1 - F(\theta_n)) + (1-\epsilon)\,F(\theta_n) \;=\; \epsilon + (1-2\epsilon)\,F(\theta_n). \qquad (6.4.14)$$

Consequently, if we set $\widetilde{\alpha}(\epsilon)$ according to equation (6.4.12), both algorithms will have their unique fixed point when $F(\theta) = \alpha^*$, so will compute the α^*-quantile of X. The claimed form of the asymptotic variances follows from by performing calculations analogous to the proofs of Theorems 6.3.1 and 6.3.2. In particular, the partial derivative with respect to θ now has a multiplicative factor $(1-2\epsilon)$, arising from equation (6.4.14) and the chain rule. To establish that the asymptotic variance is minimized at $\epsilon = 0$, it suffices to note that the derivative of the MSE with respect to ϵ is positive, so that it is an increasing function of ϵ. □

Of course, both the algorithms will fail, as would be expected, if $\epsilon = 1/2$ corresponding to pure noise. However, as summarized in Proposition 6.4.1, as long as $\epsilon < \frac{1}{2}$, feedforward noise does not affect the asymptotic rate itself, but rather only the pre-factor in front of the $1/(mn)$ rate. Figure 6.3(b) shows how the asymptotic variances $V_m(\epsilon)$ and $V_1(\epsilon)$ as ϵ is increased towards $\epsilon = \frac{1}{2}$.

6.5 Discussion

In this Chapter, we have proposed and analyzed different approaches to the problem of decentralized quantile estimation under communication constraints. Our analysis focused on the fusion-centric architecture, in which a set of m sensor nodes each collect an observation at each time step. After n rounds of this process, the centralized oracle would be able to estimate an arbitrary quantile with mean-squared error of the order $\mathcal{O}(1/(mn))$. In the decentralized formulation considered here, each sensor node is allowed to transmit only a single bit of information to the fusion center. We then considered a range of decentralized algorithms, indexed by the number of feedback bits that the fusion center is allowed to transmit back to the sensor nodes. In the simplest case, we showed that an $\log m$-bit feedback algorithm achieves the same asymptotic variance $\mathcal{O}(1/(mn))$ as the centralized estimator. More interestingly, we also showed that that a 1-bit feedback scheme, with suitably designed step sizes, can also achieve the same asymptotic variance as the centralized oracle. We also showed that using intermediate amounts of feedback (between 1 and m bits) does not alter the scaling behavior, but improves the constant. Finally, we showed how our algorithm can be adapted to the case of noise in the feedforward links from sensor nodes to fusion center, and the resulting effect on the asymptotic variance.

Our analysis in this Chapter has focused only on the fusion center architecture illustrated in Figure 6.1. A natural generalization is to consider a more general communication network, specified by an undirected graph on the sensor nodes. One possible formulation is to allow only pairs of sensor nodes connected by an edge in this communication graph to exchange a bit of information at each round. In this framework, the problem considered in this Chapter effectively corresponds to the complete graph, in which every node communicates with every other node at each round. This more general formulation raises interesting questions as to the effect of graph topology on the achievable rates and asymptotic variances.

6.6 Proofs

In this section, we turn to the proofs of Theorem 6.3.1 and 6.3.2, which exploit results from the stochastic approximation literature [Kushner and Yin, 1997; Benveniste et al., 1990]. In particular, both types of parameter updates (6.3.4) and (6.3.7) can be written in the general form

$$\theta_{n+1} \;=\; \theta_n + \epsilon_n H(\theta_n, Y_{n+1}), \tag{6.6.1}$$

where $Y_{n+1} = (Y_{n+1}(1), \ldots Y_{n+1}(m))$. Note that the step size choice $\epsilon_n = 1/n$ satisfies the conditions in equation (6.3.2). Moreover, the sequence (θ_n, Y_{n+1}) is Markov, since θ_n and Y_{n+1} depend on the past only via θ_{n-1} and Y_n. We begin by stating some known results from stochastic approximation, applicable to such Markov sequences, that will be used in our analysis.

In addition to these assumptions, convergence requires an additional attractiveness condition. For each fixed $\theta \in \mathbb{R}$, let $\mu_\theta(\,\cdot\,)$ denote the distribution of Y conditioned on θ. A key quantity in the analysis of stochastic approximation algorithms is the averaged function

$$h(\theta) \;:=\; \int H(\theta, y)\mu_\theta(dy) \;=\; \mathbb{E}\left[H(\theta, Y) \mid \theta\right]. \tag{6.6.2}$$

We assume (as is true for our cases) that this expectation exists. Now the differential equation method dictates that under suitable conditions, the asymptotic behavior of the update (6.6.1) is determined essentially by the behavior of the ODE $\frac{d\theta}{dt} = h(\theta(t))$.
Almost sure convergence: Suppose that the following *attractiveness condition*

$$h(\theta)\;[\theta - \theta^*] \;<\; 0 \qquad \text{for all } \theta \neq \theta^* \tag{6.6.3}$$

is satisfied. If, in addition, the variance $R(\theta) := \mathrm{Var}[H(\theta; Y) \mid \theta]$ is bounded, then we are are guaranteed that $\theta_n \stackrel{a.s.}{\to} \theta^*$ (see §5.1 in [Benveniste et al., 1990]).
Asymptotic normality: In our updates, the random variables Y_n take the form $Y_n = g(X_n, \theta_n)$ where the X_n are i.i.d. random variables. Suppose that the following stability condition is satisfied:

$$\gamma(\theta^*) \;:=\; -\frac{dh}{d\theta}(\theta^*) \;>\; \frac{1}{2}. \tag{6.6.4}$$

Then we have

$$\sqrt{n}\;(\theta_n - \theta^*) \;\stackrel{d}{\to}\; \mathcal{N}\left(0, \frac{R(\theta^*)}{2\gamma(\theta^*) - 1)}\right) \tag{6.6.5}$$

See §3.1.2 in [Benveniste et al., 1990] for further details.

6.6.1 Proof of Theorem 6.3.1

(a) The m-bit feedback algorithm is a special case of the general update (6.6.1), with $\epsilon_n = \frac{1}{n}$ and $H(\theta_n, Y_{n+1}) = K\left[\alpha^* - \frac{1}{m}\sum_{i=1}^{m} Y_{n+1}(i; \theta_n)\right]$. Computing the averaged function (6.6.2), we have

$$\begin{aligned} h(\theta) &= K\mathbb{E}\left[\alpha^* - \frac{1}{m}\sum_{i=1}^{m} Y_{n+1}(i) \mid \theta_n\right] \\ &= K\left(\alpha^* - F(\theta_n)\right), \end{aligned}$$

where $F(\theta_n) = \mathbb{P}(X \leq \theta_n)$. We then observe that θ^* satisfies the attractiveness condition (6.6.3), since

$$[\theta - \theta^*]\, h(\theta_n) \;=\; K\,[\theta - \theta^*]\,[\alpha^* - F(\theta_n)] \;<\; 0$$

for all $\theta \neq \theta^*$, by the monotonicity of the cumulative distribution function. Finally, we compute the conditional variance of H as follows:

$$\begin{aligned} R(\theta_n) &= K^2 \mathrm{Var}\left[\alpha^* - \frac{\sum_{i=1}^{m} Y_{n+1}(i)}{m} \mid \theta_n\right] \\ &= \frac{K^2}{m} F(\theta_n)\,[1 - F(\theta_n)] \;\leq\; \frac{K^2}{4m}, \end{aligned} \tag{6.6.6}$$

using the fact that H is a sum of m Bernoulli variables that are conditionally i.i.d. (given θ_n). Thus, we can conclude that $\theta_n \to \theta^*$ almost surely.
(b) Note that $\gamma(\theta^*) \;=\; -\frac{dh}{d\theta}(\theta^*) = Kp_X(\theta^*) \;>\; \frac{1}{2}$, so that the stability condition (6.6.4) holds. Applying the asymptotic normality result (6.6.5) with the variance $R(\theta^*) = \frac{K^2}{m}\alpha^*(1-\alpha^*)$ (computed from equation (6.6.6)) yields the claim.

□

6.6.2 Proof of Theorem 6.3.2

This argument involves additional analysis, due to the aggregate decision (6.3.6) taken by the fusion center. Since the decision Z_{n+1} is a Bernoulli random variable; we begin by computing its parameter. Each transmitted bit $Y_{n+1}(i)$ is $\mathrm{Ber}(F(\theta_n))$, where we recall the notation $F(\theta) := \mathbb{P}(X \leq \theta)$. Using the definition (6.3.10), we have the equivalences

$$\mathbb{P}(Z_{n+1} = 1) \;=\; G_m(F(\theta_n), \alpha^*) \tag{6.6.7a}$$

$$\beta \;=\; G_m(\alpha^*, \alpha^*) \;=\; G_m(F(\theta^*), \alpha^*). \tag{6.6.7b}$$

We start with the following result.

Lemma 6.6.1. *For fixed $x \in [0, 1]$, the function $f(r) := G_m(r, x)$ is non-negative, differentiable and monotonically decreasing.*

Proof: Non-negativity and differentiability are immediate. To establish monotonicity, note that $f(r) = \mathbb{P}(\sum_{i=1}^{m} Y_i \leq xm)$, where the Y_i are i.i.d. $\mathrm{Ber}(r)$ variates. Consider a second $\mathrm{Ber}(r')$ sequence Y_i' with $r' > r$. Then the sum $\sum_{i=1}^{m} Y_i'$ stochastically dominates $\sum_{i=1}^{m} Y_i$, so that $f(r) < f(r')$ as required.

□

To establish almost sure convergence, we use a similar approach as in the previous theorem. Using the equivalences (6.6.7), we compute the function h as follows

$$\begin{aligned} h(\theta) &= K_m \mathbb{E}\left[Z_{n+1} - \beta \mid \theta\right] \\ &= K_m \left[G_m(F(\theta), \alpha^*) - G_m(F(\theta^*), \alpha^*)\right]. \end{aligned}$$

Next we establish the attractiveness condition (6.6.3). In particular, for any θ such that $F(\theta) \neq F(\theta^*)$, we calculate that $h(\theta)\ [\theta - \theta^*]$ is given by

$$K_m \Big\{ G_m(F(\theta_n), \alpha^*) - G_m(F(\theta^*), \alpha^*) \Big\}\ [\theta_n - \theta^*] \quad < \quad 0,$$

where the inequality follows from the fact that $G_m(r, x)$ is monotonically decreasing in r for each fixed $x \in [0, 1]$ (using Lemma 6.6.1), and that the function F is monotonically increasing. Finally, computing the variance $R(\theta) := \mathrm{Var}\left[H(\theta, Y) \mid \theta\right]$, we have

$$R(\theta) \quad = \quad K_m^2 G_m(F(\theta), \alpha^*)\ \left[1 - G_m(F(\theta), \alpha^*)\right] \ \leq \ \frac{K_m^2}{4},$$

since (conditioned on θ), the decision Z_{n+1} is Bernoulli with parameter $G_m(F(\theta); \alpha^*)$. Thus, we can conclude that $\theta_n \to \theta^*$ almost surely.

(b) To show asymptotic normality, we need to verify the stability condition. By chain rule, we have $\frac{h}{d\theta}(\theta^*) = K_m \frac{\partial G_m}{\partial r}(r, \alpha^*)\Big|_{r=F(\theta)} p_X(\theta)$. From Lemma 6.6.1, we have $\frac{\partial G_m}{\partial r}(F(\theta), \alpha^*) < 0$, so that the stability condition holds as long as $\gamma_m(\theta^*) > \frac{1}{2}$ (where γ_m is defined in the statement). Thus, asymptotic normality holds.

In order to compute the asymptotic variance, we need to investigate the behavior of $R(\theta^*)$ and $\gamma(\theta^*)$ as $m \to +\infty$. First examining $R(\theta^*)$, the central limit theorem guarantees that $G_m(F(\theta^*), y) \to \Phi\left(\sqrt{m} \frac{y - \alpha^*}{\alpha^*(1-\alpha^*)}\right)$. Consequently, we have

$$R(\theta^*) \ = \ K_m^2 G_m(F(\theta^*), \alpha^*)\ \left[1 - G_m(F(\theta^*), \alpha^*)\right] \to \frac{K_m^2}{4}.$$

We now turn to the behavior of $\gamma(\theta^*)$. We first prove a lemma to characterize the asymptotic behavior of $G_m(r, \alpha^*)$:

Lemma 6.6.2. *(a) The partial derivative of $G_m(r, x)$ with respect to r is given by:*

$$\frac{\partial G_m(r, x)}{\partial r} \quad = \quad \frac{\mathbb{E}[X \mathbb{I}(X \leq xm)] - \mathbb{E}[X]\mathbb{E}[\mathbb{I}(X \leq xm)]}{r(1-r)}, \tag{6.6.8}$$

where X is binomial with parameters (m, x), and mean $\mathbb{E}[X] = xm$.
(b) Moreover, as $m \to +\infty$, we have

$$\frac{\partial G_m(r, \alpha^*)}{\partial r}\Big|_{r=F(\theta^*)} \quad \to \quad -\sqrt{\frac{m}{2\pi\alpha^*(1-\alpha^*)}}.$$

Proof: (a) Computing the partial derivative, we have

$$\begin{aligned}
\frac{\partial G_m(r,x)}{\partial r} &= \sum_{i=0}^{\lfloor m\alpha^* \rfloor} \binom{m}{i} \left[i r^{i-1}(1-r)^{m-i} - (m-i) r^i (1-r)^{m-i-1} \right] \\
&= \frac{1}{r(1-r)} \sum_{i=0}^{\lfloor mx \rfloor} \left(\begin{array}{c} m \\ i \end{array} \right) (i - mr) r^i (1-r)^{m-i} \\
&= \frac{1}{r(1-r)} \left(\sum_{i=0}^{\lfloor mx \rfloor} \left(\begin{array}{c} m \\ i \end{array} \right) r^i (1-r)^{m-i} - mr \sum_{i=0}^{\lfloor mx \rfloor} \left(\begin{array}{c} m \\ i \end{array} \right) r^i (1-r)^{m-i} \right) \\
&= \frac{1}{r(1-r)} \left(\mathbb{E}[X \mathbb{I}(X \leq mx)] - \mathbb{E}[X] \mathbb{E}[\mathbb{I}(X \leq mx)] \right),
\end{aligned}$$

as claimed.
(b) We derive this limiting behavior by applying classical asymptotics to the form of $\frac{\partial G_m(r,\alpha^*)}{\partial r}$ given in part (a). Defining $Z_m = \frac{X - \alpha^* m}{\sqrt{m}}$, the central limit theorem yields that:

$$\begin{aligned}
Z_m &\stackrel{d}{\to} Z \sim N(0, a) \\
a &:= \alpha^* \left(1 - \alpha^*\right)
\end{aligned} \tag{6.6.9}$$

Moreover, in this binomial case, we actually have $\mathbb{E}[|Z_m|] \to \mathbb{E}[|Z|] = \sqrt{\frac{2a}{\pi}}$.

First, since $\mathbb{E}[X] = \alpha^* m$ and $\mathbb{E}[\mathbb{I}(X \leq \alpha^* m)] \to \frac{1}{2}$ by the CLT, we have

$$\mathbb{E}[X] \; \mathbb{E}[\mathbb{I}(X \leq \alpha^* m)] \quad \to \quad \frac{\alpha^* m}{2}. \tag{6.6.10}$$

Let us now re-write the first term in the representation (6.6.8) of $\frac{\partial G_m(r,\alpha^*)}{\partial r}$ as

$$\begin{aligned}
\mathbb{E}[X \mathbb{I}(X \leq \alpha^* m)] &= \alpha^* m \mathbb{E}[\mathbb{I}(X \leq \alpha^* m)] + \sqrt{m} \mathbb{E}[Z_m \, \mathbb{I}(Z_m \leq 0)] \\
&\to \frac{\alpha^* m}{2} - \sqrt{m} \sqrt{\frac{a}{2\pi}}
\end{aligned} \tag{6.6.11}$$

since $\mathbb{E}[\mathbb{I}(X \leq \alpha^* m)] \to 1/2$ and

$$\mathbb{E}[Z_m \, \mathbb{I}(Z_m \leq 0)] \; \to \; \mathbb{E}[Z \mathbb{I}(Z \leq 0)] = \frac{1}{2} \mathbb{E}[|Z|] = \sqrt{\frac{a}{2\pi}}.$$

Putting together the limits (6.6.10) and (6.6.11), we conclude that $\frac{\partial G_m(r,\alpha^*)}{\partial r}\Big|_{r=\alpha^*}$ converges to

$$\frac{1}{\alpha^*(1-\alpha^*)} \left[\left\{ \frac{\alpha^* m}{2} - \sqrt{m} \sqrt{\frac{\alpha^* \left(1 - \alpha^*\right)}{2\pi}} \right\} - \frac{\alpha^* m}{2} \right] \quad = \quad -\sqrt{\frac{m}{2\pi \alpha^* (1 - \alpha^*)}},$$

as claimed. $\square$

Returning now to the proof of the theorem, we use Lemma 6.6.2 and put the pieces together to obtain that $\frac{R(\theta^*)}{2K_m\left|\frac{\partial G_m(r,\theta^*)}{\partial r}\Big|_{r=\alpha^*}\right|p_X(\theta^*)-1}$ converges to

$$\frac{K_m^2/4}{\frac{2K_m\sqrt{m}p_X(\theta^*)}{\sqrt{2\pi\alpha^*(1-\alpha^*)}}-1} \quad = \quad \frac{1}{m}\left[\frac{K^2\sqrt{2\pi\alpha^*(1-\alpha^*)}}{8Kp_X(\theta^*)-4\sqrt{2\pi\alpha^*(1-\alpha^*)}}\right],$$

with $K > \frac{\sqrt{2\pi\alpha^*(1-\alpha^*)}}{2p_X(\theta^*)}$ for stability, thus completing the proof of the theorem.

□

6.6.3 Proof of Theorem 6.4.1

We proceed in an analogous manner to the proof of Theorem 6.3.1:

Lemma 6.6.3. *For fixed $x \in [0,1]$, the function $G_{m,\ell}(r,x)$ is non-negative, differentiable and monotonically decreasing.*

Proof: Some straightforward algebra using the results of Lemma 6.6.2 shows that the partial derivative $\frac{\partial G_{m,\ell}(r,x)}{\partial r}$ is

$$\frac{1}{r(1-r)}\sum_{k=-\ell}^{\ell-1} r_k\left\{\mathbb{E}\left[X\,\mathbb{I}\left(x-s_{k+1}\le\frac{X}{m}\le x-s_k\right)\right]-\mathbb{E}[X]\,\mathbb{P}\left[x-s_{k+1}\le\frac{X}{m}\le x-s_k\right]\right\}, \tag{6.6.12}$$

which can be seen to be non-positive. □

The finiteness of the variance of the quantization step is clear by construction; more specifically, a crude upper bound is r_ℓ^2. Thus, analogous to the previous theorems, Lemma 6.6.3 is used to establish almost sure convergence.

To compute the asymptotic variance, we again exploit asymptotic normality (see equation (6.6.9)) as before:

$$\begin{aligned}
\mathbb{E}[X\mathbb{I}(m(\alpha^*-s_{k+1})\le X\le m(\alpha^*-s_k))] &= \mathbb{E}\left[X\mathbb{I}\left(-\sqrt{m}s_{k+1}\le\frac{X-\alpha^*m}{\sqrt{m}}\le-\sqrt{m}s_k\right)\right]\\
&= \sqrt{m}\mathbb{E}\left[(Z+\alpha^*\sqrt{m})\mathbb{I}\left(-\sqrt{m}s_{k+1}\le Z\le-\sqrt{m}s_k\right)\right]\\
&= \sqrt{m}\mathbb{E}\left[Z\mathbb{I}\left(-\sqrt{m}s_{k+1}\le Z\le-\sqrt{m}s_k\right)\right]+S\\
&\to -\sqrt{m}\int_{\sqrt{m}s_k}^{\sqrt{m}s_{k+1}} z\frac{\exp\left(\frac{-z^2}{2a}\right)}{\sqrt{2\pi a}}dz+S,
\end{aligned}$$

where

$$S := \mathbb{E}[X]P(m(x-s_{k+1})\le X\le m(x-s_k)).$$

Solving the integral above:

$$\Delta_m(s_k,s_{k+1}) = \left(\exp\left(-\frac{ms_k^2}{2\alpha^*(1-\alpha^*)}\right)-\exp\left(-\frac{ms_{k+1}^2}{2\alpha^*(1-\alpha^*)}\right)\right).$$

Thus, plugging into Equation 6.6.12, noticing that S cancels:

$$\frac{\partial G_{m,\ell}(r,\alpha^*)}{\partial r}\Big|_{r=F(\theta^*)} \to -\sqrt{\frac{m}{2\pi\alpha^*(1-\alpha^*)}}\sum_{k=-\ell}^{\ell-1} r_k \Delta_m(s_k, s_{k+1}).$$

A side note is that if one chooses $s_0 = 0$, we are guaranteed that at least one $\Delta_m(s_k, s_{k+1})$ does not go to zero in a fixed quantizer (i.e. a quantizer where the levels s_k do not depend on m). But the correction factor expression, and as a matter of fact, the optimum quantization of Gaussian, suggests that the levels s_k scale as $1/\sqrt{m}$. In this case, the factor is a constant, independent of m.

We now need to compute $R(\theta^*)$ for the quantized updated. It is also straightforward to see that this quantity is given by:

$$R(\theta^*) = K_m^2 \sum_{k=-\ell}^{\ell-1} r_k^2 (G_m(F(\theta^*), \alpha^* - s_k) - G_m(F(\theta^*), \alpha^* - s_{k+1})) - \beta^2.$$

Putting everything together we obtain the asymptotic variance estimate for the more general quantizer converges to:

$$\frac{R(\theta^*)}{2K_m \left| \frac{\partial G_{m,\ell}(r,\theta^*)}{\partial r}\Big|_{r=\alpha^*} \right| p_X(\theta^*) - 1} \to \frac{K_m^2 \sum_{k=-\ell}^{\ell-1} r_k^2 (G_m(F(\theta^*), \alpha^* - s_k) - G_m(F(\theta^*), \alpha^* - s_{k+1})) - \beta^2}{\frac{2K_m\sqrt{m}\sum_{k=-\ell}^{\ell-1} r_k \Delta_m(s_k, s_{k+1}) p_X(\theta^*)}{\sqrt{2\pi\alpha^*(1-\alpha^*)}} - 1}.$$

Set a gain $K = \frac{K_m\sqrt{m}\sum_{k=-\ell}^{\ell-1} r_k \Delta_m(s_k, s_{k+1})}{\sqrt{2\pi\alpha^*(1-\alpha^*)}}$ and we have the final expression for the variance:

$$2\pi\frac{\sum_{k=-\ell}^{\ell-1} r_k^2 \Delta G_m(s_k, s_{k+1}) - \beta^2}{\sum_{k=-\ell}^{\ell-1} r_k \Delta_m(s_k, s_{k+1})}\left[\frac{K^2\alpha^*(1-\alpha^*)}{2Kp_X(\theta^*) - 1}\frac{1}{m}\right],$$

where $\Delta G_m(s_k, s_{k+1}) = G_m(\alpha^*, \alpha^* - s_k) - G_m(\alpha^*, \alpha^* - s_{k+1})$. The constant $\kappa(\alpha^*, \mathcal{Q}_\ell)$ defines the performance of the algorithm for different quantization choices:

$$\kappa(\alpha^*, \mathcal{Q}_\ell) = 2\pi\frac{\sum_{k=-\ell}^{\ell-1} r_k^2 \Delta G_m(s_k, s_{k+1}) - \beta^2}{\sum_{k=-\ell}^{\ell-1} r_k \Delta_m(s_k, s_{k+1})}.$$

The rate with respect to m is the same, independent of quantization. It is clear from previous analysis that if the best quantizers are chosen $1 \le \kappa(\alpha^*, \mathcal{Q}_\ell) \le \frac{2\pi}{4}$. Obviously $\kappa(\alpha^*, \mathcal{Q}_\ell)$ over the class of optimal quantizers is a decreasing function of ℓ.

$\square$

Chapter 7

Real-time measurement of link vehicle count and travel time in a road network

7.1 Introduction

In Chapter 2 we saw that a road network is an interconnection of links such as freeway sections, on- and off-ramps, and urban road segments. At any time a link has a certain *spatial occupancy* or *vehicle count*, the number of vehicles in the link. At the prevailing speed the number of vehicles that will move from an upstream link depends on its vehicle count, and the number of vehicles that a downstream link can accept is limited by the downstream link vehicle count. Thus the *state of the road network* at any time consists of the vehicle count and speed (or travel time) in every link.

At some link interconnection junctions, vehicle movement is controlled by programmable field elements such as intersection signals, ramp-metering signals, and message signs announcing emergency conditions, speed limits, tolls, and travel time estimates.

The evolution of traffic in the road network is governed by its state and the signal settings and messages selected by the algorithms being executed in the field elements. These selections are based on an estimate of the current network state. The better the quality of this estimate, the more effectively can algorithms improve road network performance. Because current detectors (loops, video, radar) measure vehicle volume, speed, and occupancy at fixed locations, they cannot directly measure the state of a road network.

We describe a system that measures the link vehicle count and travel time of a road network in real time. The system deploys magnetic wireless sensors at the ends of links. As vehicles move over the sensors, their magnetic signatures are recorded and a matching procedure is used to track their movement. A very efficient algorithm calculates optimum matchings. The system is tested in an urban arterial road segment. This appears to be the first reliable and cost effective means for measuring vehicle counts and travel times in arterial roads and freeways.

Related prior work is summarized in Section 7.2. The test site and the measurement system are described in Section 7.3. The matching problem and the statistical model used to evaluate matching algorithms occupy Section 7.4. Optimal matching algorithms are presented in Section 7.5. A method to calibrate the statistical signature model is given in Section 7.6. A real time extension of the matching algorithm is described in Section 7.7. Empirical results and an evaluation of the algorithm's performance are presented in Section 7.9. Conclusions and directions for future work can be found in Section 7.10.

7.2 Related Work

Schemes for estimating travel times based on matching inductive loop signatures at two detector locations were demonstrated in Sun et al. [1999]; Oh and Ritchie [2002]; Ndoye et al. [2008]. All require an independent speed measurement that is used to 'normalize' the signature waveform, assuming constant vehicle speed. If a vehicle is accelerating or decelerating, this assumption is invalid and, as Ndoye et al. [2008] report, the rate of correct matching then drops drastically. Lengths of vehicles in platoons at the two locations are compared in Coifman [1999], also requiring an independent speed measurement. None of these schemes would work satisfactorily in a link with traffic signals causing stop-and-go movement. Platoon vehicle lengths used in Coifman [1999] and platoon vehicle colors used in Sun et al. [2004] would not work well for the additional reason that intersections would break platoons up. Vehicles can be re-identified by matching unique tags or license plates; but besides raising privacy concerns, these schemes require mounting an overhead camera or tag reader in each lane making it too expensive to deploy over an arterial network.

The method of Skabardonis and Geroliminis [2005] estimates the average travel time across a signalized link based on a kinematic wave model, using 30-second flow and occupancy measurements from an upstream loop detector and the exact times of the red and green phases. The similar method of Liu and Ma [2008] uses the exact time each vehicle crosses the detector. The two approaches require precise signal phase times, which must be synchronized with the detector times. For a link with multiple intersections, each intersection must be instrumented, which is expensive.

The system presented here requires no information about signal timings and does not require every intersection to be instrumented. Furthermore, it gives individual vehicle travel times and, unlike all these methods, also measures link vehicle counts. The use of the system to deduce signal phases and measure arterial performance is discussed in detail in Kwong et al. [2008].

7.3 Measuring Link Vehicle Count and Travel Time

On the left in Figure 7.1 is a map of the 0.9 mile-long test segment of southbound San Pablo Avenue in Albany, CA, starting at A (Fairmount) and ending at D (Buchanan). The segment is divided into three links, $A \to B$, $B \to C$, $C \to D$. Link $A \to B$ spans four signalized intersections (the three circles plus the intersection at Washington), links $B \to C$ and $C \to D$ each span one signalized intersection. Sensors at A, B, C, and D are

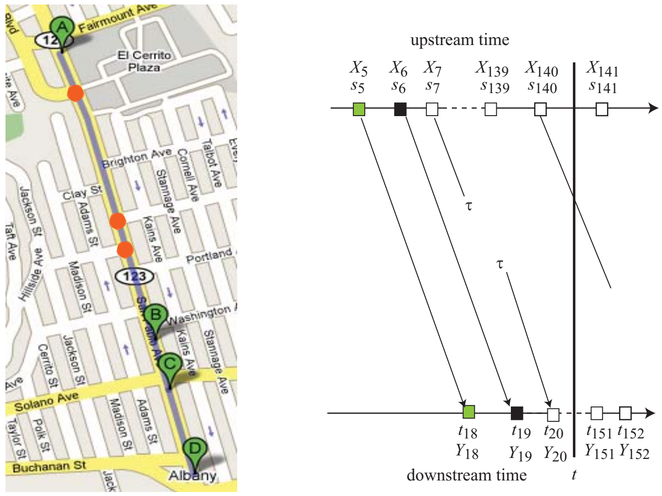


Figure 7.1: Vehicle re-identification by signature matching.

located immediately downstream (12m) of the corresponding intersection. Thus each link is delimited by one upstream and one downstream sensor.

As a vehicle crosses a sensor, it numbers the vehicle consecutively, registers the time, and records its magnetic signature (described more fully in Section 7.4). Thus the upstream sensor generates a triple (i, s_i, X_i) for each vehicle: $(5, s_5, X_5), (6, s_6, X_6), \cdots$; the downstream sensor generates triples $(j, t_j, Y_j) : (18, t_{18}, Y_{18}), (19, t_{19}, Y_{19}), \cdots$. The measurement triples are sent via radio to an Access Point (AP) on the side of the road.

The AP matches the signatures in real time. As suggested by the figure, upstream vehicle 5 is the same as downstream vehicle 18, that is, their signatures X_5 and Y_{18} match; similarly, X_6 and Y_{19} match. On the other hand, upstream vehicle $i = 7$ has turned away from the lane before reaching the downstream sensor; similarly $\tau \to Y_{20}$ indicates that downstream vehicle $j = 20$ turned into the lane but did not cross the upstream sensor.

We now describe how the AP estimates link vehicle count and travel time. At any time t, the AP finds the number K of the most recent upstream vehicle that was registered before t ($K = 140$ in the figure) and the number I of the most recent upstream vehicle that was matched with a downstream signature J ($I = 6$, $J = 19$). Then the number of vehicles in the link at time t is estimated as $K - I$ ($140 - 6 = 134$), and the link travel time of the most recent departing vehicle is $t_J - s_I$ ($t_{19} - s_6$). This 'vehicle re-identification via signature matching' scheme gives in real time the link vehicle count and travel time.

We can bound the vehicle count measurement error assuming that every vehicle is detected, which is justified [Haoui et al., 2008b], but some vehicle matches may be missed. The index K of the most recent upstream vehicle is then correct. But the most recent upstream vehicle $I_{max} \geq I$ that leaves before t may be missed by the matching algorithm. Hence, if there are no turning movements, the true vehicle count $N(t)$ is

$$N(t) = K - I_{max} \leq K - I.$$

Equality will not hold if $I < I_{max}$, which happens when upstream vehicle I_{max} is not

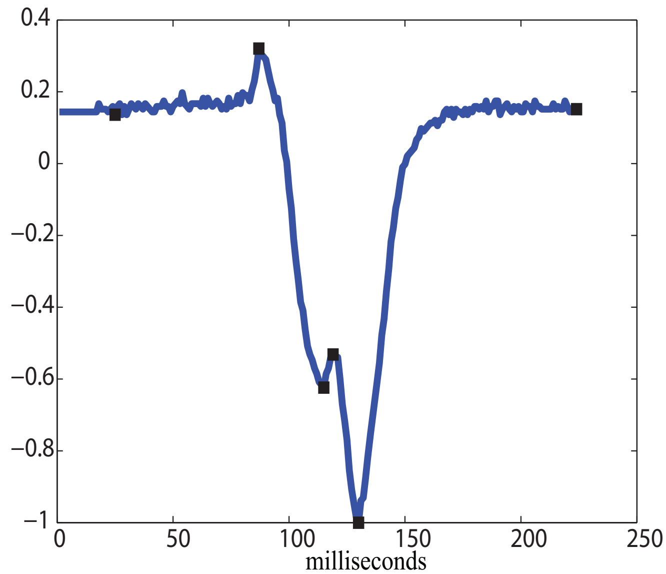


Figure 7.2: Raw z-axis magnetic signal recorded by a vehicle and peak values.

matched. If the matching probability is p, on average $I_{max} - I = p^{-1} - 1$, so if $p > 0.5$ (which it is in the test results), the estimate $K - I$ differs from $N(t)$ by at most 1 on average.

If there are turning movements, the bound changes to

$$N(t) \leq K - I - n_{out} + n_{in},$$

in which n_{out} is the number of upstream vehicles with index between I and K (like $i = 7$ in the figure) that turned before reaching the downstream sensor, and n_{out} is the number of vehicle with index larger than J (like $j = 20$) that came into the link without crossing the upstream vehicle.

7.4 Matching Problem

Suppose over an observation interval the upstream and downstream sensors generate the arrays of triples $\{(i, s_i, X_i), 1 \leq i \leq N\}$ and $\{(j, t_j, Y_j), 1 \leq j \leq M\}$. The matching is done in two steps. In the *signal processing* step each pair (X_i, Y_j) of upstream and downstream signatures is compared to produce a distance $d(i, j) = \delta(X_i, Y_j) \geq 0$ between them. This step thereby reduces the signature data to the $N \times M$ matrix $D = \{d(i, j) \mid 1 \leq i \leq N, 1 \leq j \leq M\}$.

We will shortly describe the signal processing step, followed by a statistical model of the distance function $d(i, j)$. The model is used to evaluate any *matching* or function μ,

$$\mu : \{1, \cdots, N\} \to \{1, \cdots, M, \tau\}, \tag{7.4.1}$$

with this interpretation: $\mu(i) = j$ or τ accordingly as upstream signature X_i is matched to

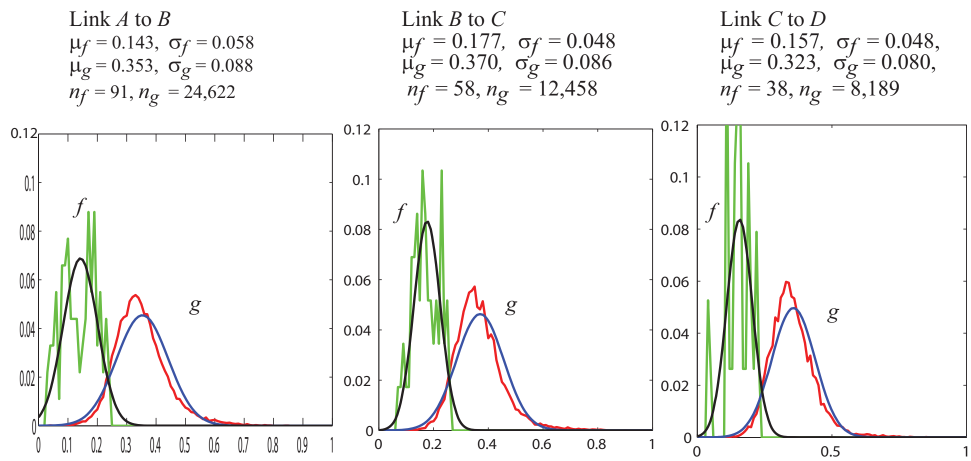


Figure 7.3. The empirical pdfs f and g and their Gaussian approximations for links $A \to B$, $B \to C$ and $C \to D$.

downstream signature Y_j or is not matched to any downstream signature, as in Figure 7.1. Let $\mathcal{M}$ be the set of all matchings.

The second step formulates the matching problem, which is to find the matching $\mu \in \mathcal{M}$ that is 'closest' to the true matching, denoted $\bar{\mu}$.

7.4.1 Signal processing

A sensor comprises an array of five nodes, each with a three-axis magnetometer that measures the x, y, z directions of the earth's magnetic field sampled at 128Hz as a vehicle goes over the node. Figure 7.2 shows the raw z-axis measured signal from one node. The other axes measurements are similar.

The microprocessor in the node automatically extracts the sequence of peak values (local maxima and minima) from each of these signals. In the figure, there are six peak values (including the initial and terminal values of the signal), denoted by squares. The node transmits the array of these peak values to the AP. The three axes yield three such arrays. The three arrays form a slice of the vehicle's two-dimensional magnetic 'footprint'. A slice is determined by the distribution of the ferrous material in the vehicle within 12" from the node. For each vehicle, the AP receives one slice from each of the five nodes. The five slices constitute the vehicle's signature at the sensor. A vehicle's signature is unique, making it possible to distinguish vehicles with the same model and make. (See [Haoui et al., 2008b] for a full description of a node and an AP.)

The signal processing algorithm takes two signatures, say $X = (X^1, \cdots, X^7)$ and $Y = (Y^1, \cdots, Y^7)$ (X^i, Y^j are the slices), and computes a distance (a measure of dissimilarity) between each pair of slices. The distance $\delta(X, Y)$ is the minimum of the distances between all pairs of slices.

7.4.2 Statistical model of distance

We assume that the distance matrix is characterized by two probability density functions (pdf), f and g: f is the pdf of the distance $\delta(X_v, Y_v)$ between the signatures at the upstream and downstream sensors of the *same* randomly selected vehicle v,

$$f(d) = p(\delta(X_v, Y_v) = d);$$

and g is the pdf of the distance $\delta(X_v, Y_w)$ between two *different* randomly selected vehicles v and w:

$$g(d) = p(\delta(X_v, Y_w) = d).$$

Then, conditional on the true matching $\bar{\mu}$, the coefficients of the random observation matrix D have the pdf

$$d(i,j) \approx \begin{cases} f, & \bar{\mu}(i) = j \\ g, & \bar{\mu}(i) \neq j \text{ or } \bar{\mu}(i) = \tau \end{cases}$$

We assume that conditional on the true matching $\bar{\mu}$ the $d(i,j)$ are independent random variables. Let $D_i = \{d(i,j),\ 1 \leq j \leq M\}$ be the array of distances between X_i and all the Y_j. Then

$$p(D \mid \bar{\mu}) = \prod_i p(D_i \mid \bar{\mu}(i)) \tag{7.4.2}$$

$$\begin{aligned} p(D_i \mid \bar{\mu}(i)) &= \begin{cases} f(d(i,j)) \prod_{k\neq j} g(d(i,k)), & \bar{\mu}(i) = j \\ \prod_k g(d(i,k)), & \bar{\mu}(i) = \tau \end{cases} \\ &= \begin{cases} L(d(i,j))\gamma(D_i), & \bar{\mu}(i) = j \\ \gamma(D_i), & \bar{\mu}(i) = \tau \end{cases} \end{aligned} \tag{7.4.3}$$

in which

$$L(d(i,j)) = \frac{f(d(i,j))}{g(d(i,j))}, \quad \gamma(D_i) = \prod_{k=1}^{M} g(d(i,k)). \tag{7.4.4}$$

Relations (7.4.2)-(7.4.4) constitute the signature distance statistical model.

Figure 7.3 displays the empirical pdfs and the Gaussian approximations of f and g for the three links. The annotation above the left plot for link $A \rightarrow B$ means that μ_f and σ_f are the mean and standard deviation for f; μ_g and σ_g are the mean and standard deviation for g; $n_f = 91$ and $n_g = 24,622$ are the number of samples used to estimate the statistics for f and g, respectively. That is, there were 91 matched vehicle pairs and 24,622 unmatched pairs. (There always are many more unmatched pairs.) Section 7.5 describes how the distributions in Figure 7.3 are estimated.

7.4.3 Matching problem

An *unconstrained matching* in the general form (7.4.1) permits duplicate matches and overtaking, i.e., two upstream vehicles i_1, i_2 with $i_2 > i_1$ may be matched with j_1, j_2 in the reverse order $j_2 < j_1$. A *constrained matching* does not permit this. Thus, it is a pair of matched sequences like $(Up, Down)$:

$$\begin{array}{ccccccccc} Up & = & X_1 & \tau & X_2 & X_3 & X_4 & X_5 & X_6 \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ Down & = & \tau & Y_1 & Y_2 & \tau & Y_3 & Y_4 & Y_5 \end{array} \tag{7.4.5}$$

The interpretation of (7.4.5) is clear. Formally, a constrained matching is a matching μ without duplicates (except for τ) and without overtaking, i.e.,

$$i_2 \geq i_1 \Rightarrow \mu(i_1) \not\geq \mu(i_2).$$

We want to find μ_c^*, the *maximum a posteriori* (MAP) matching for the constrained case. μ_c^* maximizes the posterior probability

$$p(\bar{\mu} \mid D) = \frac{p(D \mid \bar{\mu}) p_c(\bar{\mu})}{p(D)}, \tag{7.4.6}$$

in which $p_c(\bar{\mu})$ denotes the prior probability that $\bar{\mu}$ is the true constrained matching. In (7.4.6), $p(D \mid \bar{\mu})$ is given by the signature distance model (7.4.2)-(7.4.4), so we only need to specify the prior $p_c(\bar{\mu})$, which we take to be the unconstrained prior $p(\bar{\mu})$ given below in (7.4.8), conditioned by the requirement that $\bar{\mu}$ is a constrained matching. That is,

$$p_c(\bar{\mu}) = \begin{cases} p(\bar{\mu}) / \sum\limits_{\bar{\mu} \in \mathcal{M}_c} p(\bar{\mu}) & \bar{\mu} \in \mathcal{M}_c \\ 0 & \bar{\mu} \notin \mathcal{M}_c \end{cases} \tag{7.4.7}$$

in which $\mathcal{M}_c$ denotes the set of constrained matchings.

The unconstrained prior $p(\bar{\mu})$ is the uniform distribution on $\bar{\mu}$ with turning probability β:

$$p(\bar{\mu}) = \prod_i p(\bar{\mu}(i)); \quad p(\bar{\mu}(i) = j) = \alpha,\ j = 1, \cdots, M;$$
$$p(\bar{\mu}(i) = \tau) = \beta, \tag{7.4.8}$$

with $M\alpha + \beta = 1$. Using (7.4.2)-(7.4.4) and (7.4.8) gives

$$p(D \mid \bar{\mu}) p(\bar{\mu}) = \prod_i p(D_i \mid \bar{\mu}(i)) p(\bar{\mu}(i)), \tag{7.4.9}$$

$$p(D_i \mid \bar{\mu}(i)) p(\bar{\mu}(i)) = \begin{cases} L(d(i,j)) \gamma(D_i) \alpha, & \bar{\mu}(i) = j \\ \gamma(D_i) \beta, & \bar{\mu}(i) = \tau \end{cases} \tag{7.4.10}$$

Thus μ_c^* is given by

$$\mu_c^* = \arg \max_{\bar{\mu} \in \mathcal{M}_c} \frac{p(D \mid \bar{\mu}) p(\bar{\mu})}{p(D)} = \arg \max_{\bar{\mu} \in \mathcal{M}_c} p(D \mid \bar{\mu}) p(\bar{\mu}). \tag{7.4.11}$$

The last equality follows from $p(D) = \sum_{\bar{\mu} \in \mathcal{M}_c} p(D|\bar{\mu}) p(\bar{\mu})$ not depending on $\bar{\mu}$. To find μ_c^*

we must maximize the likelihood (7.4.9) over the set $\mathcal{M}_c$. The algorithm for μ_c^* is developed in section 7.5.

The MAP *unconstrained matching* μ_u^*, is given by

$$\mu_u^* = \arg\max_{\bar{\mu}\in\mathcal{M}} p(D \mid \bar{\mu})p(\bar{\mu}). \tag{7.4.12}$$

The unconstrained MAP matching $\mu_u^*(i)$ is determined independently for each upstream vehicle i. But the constrained MAP matchings $\mu_c^*(i)$ are all jointly determined.

7.4.4 Multiple lane matching

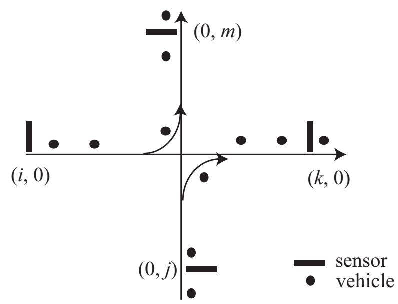


Figure 7.4: Multiple lane matching.

We consider matching vehicles that may switch from one lane to another lane downstream, as in the junction in Figure 7.4. Vehicles i and j, denoted by $(i, 0)$ and $(0, j)$ in the figure, may continue straight ahead or turn at the junction. The no-overtaking condition now means that if vehicles i and i' with $i < i'$ are in the same lane upstream, and happen to be matched to vehicles in the same lane downstream, the match respects $\mu(i) < \mu(i')$.

Suppose I vehicles from the first input sequence and J vehicles from the second input sequence are to be matched to K vehicles in the first output sequence and M vehicles in the second output sequence. An input or output vehicle maybe unmatched, denoted as matching with τ.

We use the notation $(i, 0)$, $(0, j)$, $(k, 0)$, $(0, m)$ to index the four vehicle types, so we can distinguish between (say) the third vehicle in the first input sequence $(3, 0)$ and the third vehicle in the second input sequence $(0, 3)$. However, we reserve i, j, k, m to denote a generic vehicle in these four sequences, sometimes writing i or m instead of $(i, 0)$ or $(0, m)$.

We are given the data array

$$\begin{aligned} D = \{d(i,k), d(i,m), d(j,k), d(j,m) | i \le I, \\ j \le J, k \le K, m \le M\} \end{aligned}$$

of distances between the signatures of each input vehicle and each output vehicle, together with the times t_i, t_j, t_m, t_k when the signatures were recorded. A matching μ is now any function

$$\begin{gathered}\mu : \{(i,0), 1 \le i \le I\} \cup \{(0,j), 1 \le j \le J\} \to \\ \{(k,0), 1 \le k \le K\} \cup \{(0,m), 1 \le m \le M\} \cup \{\tau\},\end{gathered}$$

with the natural interpretation: for example, $\mu(i,0) = (0,m)$ means vehicle i in the first input sequence is matched with vehicle m in the second output sequence; $m(0,j) = \tau$ means that vehicle j in the second input sequence is unmatched. Let $\mathcal{M}$ be the set of all matchings. A *constrained matching* μ is a matching without duplicates (except for τ) and without overtaking, i.e.,

$$(i,j) \ge (i',j') \Rightarrow \mu(i',j') \not\ge \mu(i,j).$$

Here $(i,j) \ge (i',j')$ means $i \ge j, i' \ge j'$. Let $\mathcal{M}_c \subset \mathcal{M}$ be the set of all constrained matchings.

We now generalize the statistical model. Conditional on the true matching $\bar{\mu}$, the data array D has the following distribution:

$$\begin{aligned}
p(D|\bar{\mu}) &= \prod_i p(D_{(i,0)}|\bar{\mu}(i,0)) \prod_j p(D_{(0,j)}|\bar{\mu}(0,j)) \\
p(D_{(i,0)}|\bar{\mu}(i,0)) &= \begin{cases} \frac{f(d(i,\bar{\mu}(i,0)))}{g(d(i,\bar{\mu}(i,0)))} \prod_k g(d(i,k)) \prod_m g(d(i,m)) \\ \prod_k g(d(i,k)) \prod_m g(d(i,m)) \end{cases} \\
&= \begin{cases} L(d(i,\bar{\mu}(i,0)))\gamma(i,0), & \bar{\mu}(i,0) \ne \tau \\ \gamma(i,0)), & \bar{\mu}(i,0) = \tau \end{cases} \\
p(D_{(0,j)} \mid \bar{\mu}(0,j)) &= \begin{cases} L(d(j,\bar{\mu}(0,j)))\gamma(0,j), & \bar{\mu}(0,j)) \ne \tau \\ \gamma(0,j), & \bar{\mu}(0,j) = \tau \end{cases}
\end{aligned}$$

Here

$$\begin{aligned}
L(d(i,\bar{\mu}(i,0))) &= \tfrac{f(d(i,\bar{\mu}(i,0)))}{g(d(i,\bar{\mu}(i,0)))}, \quad \bar{\mu}(i,0) \ne \tau \\
L(d(j,\bar{\mu}(0,j))) &= \tfrac{f(d(j,\bar{\mu}(0,j)))}{g(d(j,\bar{\mu}(0,j)))}, \quad \bar{\mu}(0,j) \ne \tau
\end{aligned}$$

and

$$\begin{aligned}
\gamma(i,0)) &= \prod_k g(d(i,k)) \prod_m g(d(i,m)) \\
\gamma(0,j) &= \prod_k g(d(j,k)) \prod_m g(d(j,m))
\end{aligned}$$

Similarly to the one-dimensional case, we assume that under the *prior distribution* on $\mathcal{M}$ the $\mu(i,j)$ are all independent and uniformly distributed, subject to the condition that $p(\mu(i,j) = \tau) = \beta$. That is,

$$p(\mu(i,j) = (k,m) \ne \tau) = \alpha, \quad p(\mu(i,j) = \tau) = \beta,$$

with

$$(K+M)\alpha+\beta=1.$$

For constrained matchings the prior distribution is

$$p_c(\mu)=\begin{cases}\frac{p(\mu)}{\sum_{\mathcal{M}_c}p(\mu)}, & \mu\in\mathcal{M}_c\\ 0, & \mu\notin\mathcal{M}_c\end{cases},$$

and the optimal constrained matching μ_c^* is

$$\mu_c^*=\arg\max_{\bar{\mu}\in\mathcal{M}_c}\frac{p(D\mid\bar{\mu})p(\bar{\mu})}{p(D)}=\arg\max_{\bar{\mu}\in\mathcal{M}_c}p(D\mid\bar{\mu})p(\bar{\mu}). \tag{7.4.13}$$

7.4.5 Applications

The multiple lane formulation applies with obvious changes to the junction of Figure 7.5 with several input and output streams of vehicles, provided the non-overtaking or first-in first-out (fifo) condition holds. (Each violation of the fifo condition will lead to one τ matching. If there are not too many fifo violations, the technique is sound.)

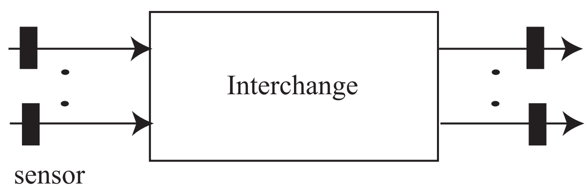


Figure 7.5: A generalized setup.

Intersections

Figure 7.5 could represent an intersection of two streets one going West to East, the other going South to North as in Figure 7.4. The matching results yield an estimate of the number of vehicles making a turn; the start and end times of the matched vehicle pairs give the travel time to cross the intersection and to make a turn.

Roundabout

This is just an extension of the intersection with R input and output sequences if R roads terminate on the roundabout.

Weaving

Estimating the number and nature of weaving movements in (say) a four-lane weaving section on a freeway corresponds to the arrangement of Figure 7.5 with four input and output sequences and sensors placed at the beginning and end of the weaving sections.

Ramp queue and delay

A ramp with two lanes that merge at the signal corresponds to Figure 7.5 with two input and one output sequence. The vehicle matchings give the number of vehicles in each ramp lane and the delay each vehicle encounters on the ramp.

Work zone

A four-lane freeway that terminates into a two-lane freeway, because a work zone takes two lanes out of service, corresponds to Figure 7.5 with four input sequences and two output sequences. The sensors are placed upstream and downstream of the work zone. The matchings will estimate the number of vehicles queued up upstream of the work zone and the delay experienced by those vehicles. The information could be displayed on a message sign.

7.5 Matching Algorithm

We now develop an efficient algorithm for the optimal constrained single lane matching problem. The multiple lane extension follows as a direct generalization.

7.5.1 Single lane matching

Instead of maximizing the likelihood (7.4.9) it is more convenient to minimize the negative 'log-likelihood',

$$\begin{aligned} -\ln p(D \mid \bar{\mu})p(\bar{\mu}) &= -\sum_i \ln p(D_i \mid \bar{\mu}(i)) - \ln p(\bar{\mu}(i)) \\ &= \sum_i \sum_j \lambda(i,j)\mathbf{1}(\bar{\mu}(i) = j) \\ &+ \sum_i \lambda(i,\tau)\mathbf{1}(\bar{\mu}(i) = \tau), \end{aligned} \tag{7.5.1}$$

in which $\mathbf{1}(\cdot)$ denotes the indicator function and

$$\lambda(i,j) = -\ln L(d(i,j)) - \ln \gamma(D_i) - \ln \alpha, \tag{7.5.2}$$

$$\lambda(i,\tau) = -\ln \gamma(D_i) - \ln \beta. \tag{7.5.3}$$

Thus to find μ_u^* we must minimize the linear form (7.5.1) over the set $\bar{\mu} \in \mathcal{M}$, which leads to

$$\mu_u^*(i) = \begin{cases} j, & \lambda(i,j) \le \lambda(i,k), \text{ all } k, \tau \\ \tau, & \lambda(i,\tau) \le \lambda(i,k), \text{ all } k \end{cases}$$

To find μ_c^* we must minimize the linear form (7.5.1) over the "combinatorial" constraint $\bar{\mu} \in \mathcal{M}_c$. The difficulty is to find a convenient representation of $\mathcal{M}_c$.

We now describe a graph $\mathcal{G}(N, M)$ whose paths are in one-one correspondence with the set $\mathcal{M}_c$ of all constrained matchings. Its $(N+1)\times(M+1)$ nodes are arranged in the form of a grid like in Figure 7.6, which is the graph for example (7.4.5) with $N = 6,\ M = 5$. $\mathcal{G}(N, M)$ is called the *edit graph* in the context of sequence comparison algorithms in molecular biology [Myers, 1986].

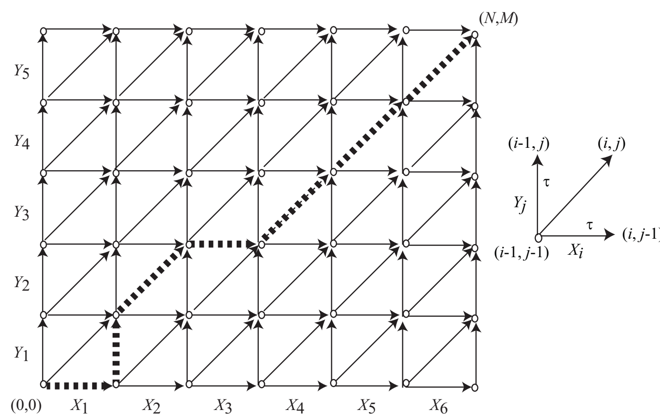


Figure 7.6. The edit graph for example (7.4.5). A diagonal edge corresponds to a signature match; a horizontal or vertical edge corresponds to a turn (match with τ).

$\mathcal{G}(N, M)$ is constructed as follows. Its nodes are labeled $(i, j),\ 0 \le i \le N,\ 0 \le j \le M$. A node $(i-1, j-1)$ has three directed edges connected to nodes $(i-1, j)$, $(i, j-1)$ and (i, j) (unless $i > N$ or $j > M$). The 'diagonal' edge $(i-1, j-1) \to (i, j)$ indicates the match $X_i \to Y_j$; the 'horizontal' edge $(i-1, j-1) \to (i, j-1)$ indicates the match $X_i \to \tau$; the 'vertical' edge $(i-1, j-1) \to (i-1, j)$ indicates the match $Y_j \to \tau$.

An obvious but very important fact is that each path in $\mathcal{G}(N, M)$ from node $(0, 0)$ to (N, M) corresponds to a constrained matching and vice versa. The constrained matching (7.4.5) corresponds to the path in Figure 7.6 indicated by the thick dashed lines.

Having identified constrained matchings with paths in the edit graph, we identify (7.5.1) with the sum of the weights of the edges along the path, assigning edge weights according to

$$\begin{array}{lcl} w((i-1, j-1) \to (i, j)) & = & \lambda(i, j) \\ w((i-1, j-1) \to (i, j-1)) & = & \lambda(i, \tau) \\ w((i-1, j-1) \to (i-1, j)) & = & 0 \end{array} . \tag{7.5.4}$$

The value (7.5.1) for a constrained matching $\bar{\mu}$ is equal to the weight of the corresponding path (defined as the sum of the edge weights) in the edit graph. Thus μ_c^* is obtained by finding the minimum weight path, which is accomplished by the following algorithm.

Let $\mathcal{N}$ be the nodes of $\mathcal{G}(N, M)$. For each $(i, j) \in \mathcal{N}$ let $Pr(i, j)$ be the *predecessor* nodes of (i, j), i.e., the nodes from which there is an edge to (i, j). Evidently,

$$Pr(i, j) = \{(i-1, j), (i, j-1), (i-1, j-1)\}.$$

Let $\prec$ be a total ordering of $\mathcal{N}$ which respects the Pr relation, i.e., for $n = (i, j)$

$$n' = (i', j') \in Pr(n) \Rightarrow n' \prec n.$$

There are many such total orders, including lexicographic order.

Algorithm

1. Set $W(0,0) = 0$.
2. Suppose $W(n')$ has been evaluated for all $n' \prec n$. Calculate

$$W(n) = \min\{W(n') + w(n' \to n) \mid n' \in Pr(n)\}, \tag{7.5.5}$$

in which $w(n' \to n)$ is the weight of the edge $n' \to n$ given by (7.5.4), and let $pr(n)$ be a minimizing predecessor node in (7.5.5).

3. Return to step 2 with the node following n in the total order $\prec$.

Then $W(i,j)$ is the minimum weight of paths connecting $(0,0)$ to (i,j). The minimum weight path can be constructed by backtracking through $pr(n)$.

The algorithm requires $N \times M$ iterations. In each iteration (7.5.5) requires evaluation of the three edge weights $w(n' \to n)$ given by (7.5.4).

7.5.2 Multiple lane matching

To find μ_c^* requires minimizing the negative of the likelihood (7.4.13) over $\mu \in \mathcal{M}_c$, which can be written as

$$\begin{aligned}
-\ln p(D \mid \bar{\mu})p(\bar{\mu}) = &\sum_i [\sum_k \lambda((i,0),(k,0))\mathbf{1}(\bar{\mu}(i,0) = (k,0)) \\
&+ \sum_m \lambda((i,0),(0,m))\mathbf{1}(\bar{\mu}(i,0) = (0,m)) \\
&+ \lambda((i,0),\tau)\mathbf{1}(\bar{\mu}(i,0) = \tau)] \\
&+ \sum_j [\sum_k \lambda((0,j),(k,0))\mathbf{1}(\bar{\mu}(0,j) = (k,0)) \\
&+ \sum_m \lambda((0,j),(0,m))\mathbf{1}(\bar{\mu}(0,j) = (0,m)) \\
&\quad + \lambda((0,j),\tau)\mathbf{1}(\bar{\mu}(0,j) = \tau)].
\end{aligned} \tag{7.5.6}$$

In (7.5.6) $\mathbf{1}(\cdot)$ is the indicator function and

$$\begin{aligned}
\lambda((i,0),(k,0)) &= -\ln L(d(i,k)) - \ln \gamma(i,0) - \ln \alpha \\
\lambda((i,0),(0,m)) &= -\ln L(d(i,m)) - \ln \gamma(i,0) - \ln \alpha \\
\lambda((i,0),\tau) &= -\ln \gamma(i,0) - \ln \beta \\
\lambda((0,j),(k,0)) &= -\ln L(d(j,k)) - \ln \gamma(0,j) - \ln \alpha \\
\lambda((0,j),(0,m)) &= -\ln L(d(j,m)) - \ln \gamma(0,j) - \ln \alpha \\
\lambda((0,j),\tau) &= -\ln \gamma(0,j) - \ln \beta
\end{aligned} \tag{7.5.7}$$

Like in the single lane case, (7.5.6) is a linear form, the combinatorial constraint is $\bar{\mu} \in \mathcal{M}_c$, and the weights are given in (7.5.7). The constraint can be represented using the graph $\mathcal{G}(I,J,K,M)$, with $(1+I) \times (1+J) \times (1+K) \times (1+M)$ nodes corresponding to a four-dimensional grid with nodes indexed (i,j,k,m).

From each node (i,j,k,m) there are four edges (labeled τ) connecting to 'adjacent' nodes $(i+1,j,k,m),(i,j+1,k,m),(i,j,k+1,m),(i,j,k,m+1)$ and four 'diagonal' edges connecting to node $(i+1,j,k+1,m),(i+1,j,k,m+1),(i,j+1,k+1,m),(i,j+1,k,m+1)$. The first four edges are interpreted to mean that vehicles $i+1,j+1,k+1,m+1$ respectively are unmatched. The last four edges are interpreted to mean that $i+1$ is matched to $k+1$, $i+1$ is matched to $m+1$, and so on.

$\mathcal{G}(I,J,K,M)$ may be called the *edit graph* in analogy with the single input-single output sequence case. It is an important result that constrained matchings are in 1-1 correspondence to paths in $\mathcal{G}$ from $(0,0,0,0)$ to (I,J,K,M). We can then identify (7.5.1) with the sum of the weights of the edges along the path, assigning edge weights according to

$$
\begin{aligned}
w((i-1,j-1,k-1,m-1) &\to (i,j-1,k,m-1)\\
&= \lambda((i,0),(k,0))\\
w((i-1,j-1,k-1,m-1) &\to (i,j-1,k-1,m)\\
&= \lambda((i,0),(0,m))\\
w((i-1,j-1,k-1,m-1) &\to (i-1,j,k,m-1)\\
&= \lambda((0,j),(k,0))\\
w((i-1,j-1,k-1,m-1) &\to (i-1,j,k-1,m)\\
&= \lambda((0,j),(0.m))\\
w((i-1,j-1,k-1,m-1) &\to (i,j-1,k-1,m-1)\\
&= \lambda((i,0),\tau)\\
w((i-1,j-1,k-1,m-1) &\to (i-1,j,k-1,m-1)\\
&= \lambda((0,j),\tau)\\
w((i-1,j-1,k-1,m-1) &\to (i-1,j-1,k,m-1)\\
&= 0\\
w((i-1,j-1,k-1,m-1) &\to (i-1,j-1,k-1,m)\\
&= 0
\end{aligned}
\tag{7.5.8}
$$

In (7.5.8) $w(n' \to n)$ is the weight assigned to the edge $n' \to n$. It is easy to check the next result.

Theorem 7.5.1. *For any path from $(0,0,0,0)$ to (I,J,K,M) the weight of the path, calculated as the sum of the edge weights given by (7.5.8), is equal to the sum (7.5.1). Hence μ_c^* is given by the minimum weight path.*

Let $\mathcal{N}$ be the nodes of $\mathcal{G}(I,J,K,M)$. For each $(i,j,k,m) \in \mathcal{N}$ let $Pr(i,j,k,m)$ be the eight *predecessor* nodes of (i,j,k,m) from which there is an edge to (i,j,k,l):

$$
\begin{aligned}
&Pr(i,j,k,m) = \{(i-1,j,k-1,m),(i-1,j,k,m-1),\\
&(i,j-1,k-1,m),(i,j-1,k,m-1),(i-1,j,k,m),\\
&(i,j-1,k,m),(i,j,k-1,m),(i,j,k,m-1)\}.
\end{aligned}
$$

Let $\prec$ be a total ordering of $\mathcal{N}$ which respects the Pr relation, i.e., for $n=(i,j,k,l)$

$$n' = (i',j',k',l') \in Pr(n) \Rightarrow n' \prec n.$$

The previous minimum weight path algorithm applies without change except that W is a four-dimensional array, initialized with $W(0,0,0,0)=0$. $W(I,J,K,M)$ is the minimum weight of paths connecting $(0,0,0,0)$ to (I,J,K,M). The minimum weight path can be constructed by backtracking through $pr(n)$. The algorithm requires $I \times J \times K \times M$ iterations.

7.6 Estimating the Model

The statistical model (7.4.2)-(7.4.4) parameterizes the probability density of the observation matrix D as $p(D \mid \bar{\mu}, \mu_f, \sigma_f, \mu_g, \sigma_g)$, with parameter $\bar{\mu}$ for the true matching, and $(\mu_f, \sigma_f, \mu_g, \sigma_g)$ for the four parameters of the Gaussian distributions of f, g.

Ideally the optimum matching and the parameters of f, g should be jointly determined as the maximum likelihood estimate

$$(\hat{\bar{\mu}}, \hat{\mu_f}, \hat{\sigma_f}, \hat{\mu_g}, \hat{\sigma_g}) = \arg \max_{\bar{\mu}, \mu_f, \sigma_f, \mu_g, \sigma_g} p(D \mid \bar{\mu}, \mu_f, \sigma_f, \mu_g, \sigma_g).$$

While this is a well-defined optimization problem, it is computationally very expensive to solve because of the combinatorial nature of the variable $\bar{\mu}$. Instead we resort to coordinate-wise optimization:

1. Begin with an initial estimate $\bar{\mu}^0$ (for which we use a matching algorithm based solely on the distance).

2. At step i we have the estimate $\bar{\mu}^i$. Find

$$(\mu_f^i, \sigma_f^i, \mu_g^i, \sigma_g^i) = \arg \max_{\mu_f, \sigma_f, \mu_g, \sigma_g} p(D \mid \bar{\mu}^i, \mu_f, \sigma_f, \mu_g, \sigma_g)).$$

   This is easy because $\bar{\mu}^i$ partitions elements of the observation matrix D into distances of pairs of matched and unmatched vehicles, so (μ_f^i, σ_f^i) are the empirical moments of the matched pairs and (μ_g^i, σ_g^i) are corresponding values for the unmatched pairs.

3. Use the optimal matching algorithm to find

$$\bar{\mu}^{i+1} = \arg \max_{\bar{\mu}} p(D \mid \bar{\mu}, \mu_f^i, \sigma_f^i, \mu_g^i, \sigma_g^i),$$

   and return to 2) with $i+1$.

Since the likelihood $p(D \mid \bar{\mu}^i, \mu_f^i, \sigma_f^i, \mu_g^i, \sigma_g^i)$ increases with i, the iteration must converge to a local maximum of the likelihood. (The iteration is reminiscent of the EM algorithm [Dempster et al., 1977].) For the test results the iteration converged in three to four steps.

7.7 Real-Time Matching

The edit graph of Section 7.5 grows with the observation time interval, and with each new upstream or downstream vehicle, one must calculate the distance of its signature from all previous signatures. The effort to compute these distances for each new vehicle grows linearly with the observation interval. For real-time implementation, we must restrict the growth of the edit graph. One way of doing this is as follows. For each new upstream vehicle i (with signature X_i) that arrives at time s_i compute the distance $d(i,k)$ as

$$d(i,k) = \begin{cases} \delta(X_i, Y_k) & \text{if } 0 \leq t_k - s_i \leq T_M \\ \infty & \text{else} \end{cases} \tag{7.7.1}$$

For each new downstream vehicle j (with signature Y_j) that arrives at time t_j compute the distance $d(m,j)$ as

$$d(m,j) = \begin{cases} \delta(X_m, Y_j) & \text{if } 0 \leq t_j - s_m \leq T_M \\ \infty & \text{else} \end{cases} \tag{7.7.2}$$

If T_M is an upper bound on the travel time, (7.7.1)-(7.7.2) must hold. The number of distances $\delta(X_i, Y_j)$ to be calculated is thus bounded, and the computational burden for each new upstream or downstream vehicle is essentially constant.

A simple choice for T_M would be to assume a minimum speed and take T_M to be the corresponding travel time, which requires knowledge of the link length, signal cycle times, etc. A better idea is this. Let μ be the minimum weight path matching until the current time. Pick an integer M. Let $i_1, \cdots, i_M$ be the most recently matched M downstream vehicles, and set

$$T_M = 2 \times \max\{t_{\mu(i_m)} - s_{i_m},\ 1 \leq m \leq M\}.$$

Thus T_M is (say) two times the maximum travel time experienced by these M vehicles. This choice will automatically adapt to changes in travel time.

7.8 Performance Analysis

In this section we evaluate the performance of two approaches for unconstrained matching, and discuss a simple heuristic that shows the benefits of adding constraints. We consider a single lane matching problem, but a similar line of reasoning can be extended for multiple input, multiple output matching. The first approach for unconstrained matching is a direct *minimum distance* matching. The matching estimate $\mu_{minD}(D)(i) = j^*$, if j^* is the minimizer of $\min_j D_{i,j}$ and $\min_j D_{i,j} \geq d^*$. If $\min_j D_{i,j} < d^*$, then $\mu_{minD}(D)(i) = \tau$. Next, we analyze unconstrained MAP matching, described in Section 7.4. To conclude the section we compute a heuristic analysis of incorporating constraints into the MAP matching formulation. The proofs for all theorems are presented in Section 7.11.

7.8.1 Minimum distance matching

Define the cumulative and complementary distribution functions

$$G(d) = \int_0^d g(x)dx; \quad \tilde{G}(d) = 1 - G(d).$$

Similarly define $F(d)$ and $\tilde{F}(d)$.

Theorem 7.8.1 gives the performance of the minimum distance matching function, μ_{minD}.

Theorem 7.8.1. *The probability of a correct match, $\mu_{minD}(i) = \bar{\mu}(i)$, is*

$$p(\mu_{minD}(D)(i) = j \mid \bar{\mu}(i) = j) = \int_0^{d^*} f(x)[\tilde{G}(x)]^{M-1} dx. \tag{7.8.1}$$

The probability of an incorrect match, $\mu_{minD}(i) \neq \bar{\mu}(i)$, is

$$p(\mu_{minD}(D)(i) \neq j \mid \bar{\mu}(i) = j) = (M-1) \int_0^{d^*} [\tilde{G}(x)]^{M-2} g(x) \tilde{F}(x) dx. \tag{7.8.2}$$

The probability that a vehicle is unmatched, $\mu_{minD}(i) = \tau$, is

$$p(\mu_{minD}(D)(i) = \tau \mid \bar{\mu}(i) = j) = \tilde{F}(d^*)[\tilde{G}(d^*)]^{M-1}. \tag{7.8.3}$$

The three probabilities (7.8.1)-(7.8.3) add up to 1.

Theorem 7.8.1 allows us to predict how the performance depends on the threshold d^* and the number M of potential vehicle matches. From (7.8.1)-(7.8.2), the probabilities of both correct and incorrect match increase as d^* increases. Thus, as in hypothesis testing generally, the proper choice of the threshold value must compromise between correct and incorrect re-identification.

Second, from (7.8.1)-(7.8.2), the probability of a correct match decreases and the probability of an incorrect match increases as M increases. This is intuitive: the larger is the number M of potential matches, the worse is the performance of the minimum distance matching function. So one way to improve the matching algorithm is to reduce M. A common way of reducing M is to place an upper bound T on the link travel time and limit the matching of an upstream vehicle i to those downstream vehicles j for which $t_j - s_i \leq T$, e.g., [Ndoye et al., 2008].

Third, the probability of a vehicle being unmatched decreases as d^* or M increases. This, too, is intuitive: a larger d^* implies a less stringent condition on matching, while a larger M increases the chance of finding a potential match.

Formulas (7.8.1)-(7.8.3) help determine the range of values of d^* and M for which the re-identification scheme gives satisfactory performance. Figure 7.7 plots the probabilities of correct and incorrect matches using the Gaussian approximations for the distributions f, g in Figure 7.3 in (7.8.1), (7.8.2) for link $A \to B$. For a per lane flow of 500 vph, $M = 50$ corresponds to a time interval of 6 minutes, which is the travel time window over a three-mile long link at an average speed of 30 mph. For $d^* = 0.15$, the minimum distance matching function is predicted to give 45% correctly matched, 25% incorrectly matched, and 30% unmatched vehicles.

7.8.2 Unconstrained MAP matching μ_{uMAP}

We shall evaluate the performance of any matching function μ by its (normalized) *reward* $\rho(\mu)$:

$$\rho(\mu) = \frac{1}{N} \mathbb{E} \sum_{i=1}^{N} \mathbf{1}(\mu(D)(i) = \bar{\mu}(i)), \tag{7.8.4}$$

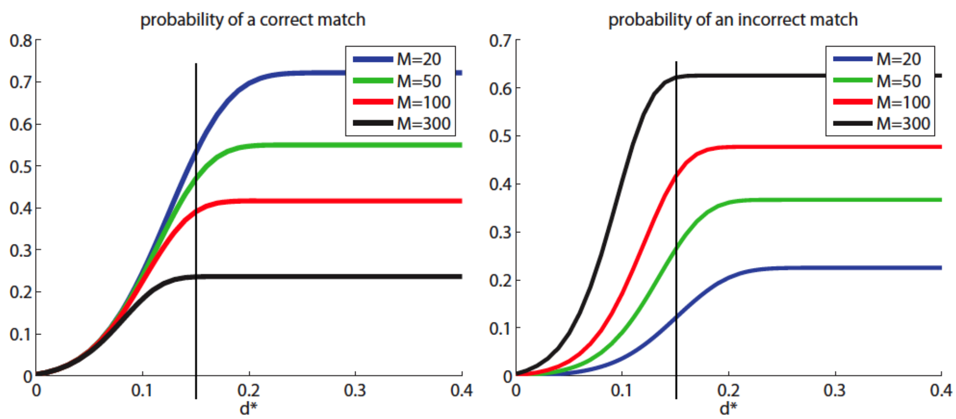


Figure 7.7. Probabilities of correct and incorrect matches of μ_{minD} for different values of d^*, M.

in which $\mathbf{1}(\cdot)$ is the indicator function: $\mathbf{1}(\mu(D)(i) = \bar{\mu}(i)) = 1$ if $\mu(D)(i) = \bar{\mu}(i)$, and $= 0$ otherwise. Thus $\rho(\mu)$ is the correct matching rate, the fraction of correctly matched vehicles on average.

To evaluate the expectation operator $\mathbb{E}$ in (7.8.4) we need the joint probability distribution on $(D, \bar{\mu})$. Since $p(D \mid \bar{\mu})$ is already specified by (7.4.2)-(7.4.4) we only need the (prior) distribution of $\bar{\mu}$. We assume that the number of upstream vehicles $N = (1+\beta)M$ of which M vehicles cross the downstream sensor and βM vehicles turn before reaching the downstream sensor. Thus β is the turning probability. Subject to this assumption, we impose a uniform distribution on $\bar{\mu}$:

$$p(\bar{\mu}(i) = j) = \alpha,\ j = 1, \cdots, M;\ \ p(\bar{\mu}(i) = \tau) = \beta, \tag{7.8.5}$$

with $M\alpha + \beta = 1$.

Let μ^* denote the *optimal* or reward-maximizing matching function.

Theorem 7.8.2. *μ^* is given by*

$$\mu^*(D)(i) = \begin{cases} j & \text{if } L(d(i,j)) \geq L(d(i,k))\ \forall k;\ L(d(i,j)) \geq \beta/\alpha \\ \tau & \text{if } L(d(i,k)) < \beta/\alpha\ \forall k \end{cases}. \tag{7.8.6}$$

To implement (7.8.6) we need β or α, since the $d(i,j)$ and M are known from the data. In the present context, β is the turning probability, which may be determined from field observations or experience. But another consideration may govern the choice of β. From (7.8.6) one sees that the larger is β, the more stringent is the requirement of a match, and lower is the probability of an incorrect match. So, depending on the application, one should choose a larger value for β, if the 'cost' of an incorrect match is high.

Observe that $\mu^*(D)$ maximizes the posterior probability

$$p(\bar{\mu} \mid D) = \frac{p(D \mid \bar{\mu})(p(\bar{\mu})}{p(D)}, \tag{7.8.7}$$

with prior probability $p(\bar{\mu})$ given by (7.8.5). So $\mu^* = \mu_{uMAP}$ is also the *(unconstrained) maximum a posteriori (MAP)* matching function.

The minimum distance matching μ_{minD} and unconstrained MAP matching μ_{uMAP} have a similar structure. One calculates a statistic for each pair (i,j) of downstream and upstream vehicles—$d(i,j)$ in the μ_{minD} case and the likelihood ratio $L(d(i,j))$ in the μ_{uMAP} case—and matches i to the best j in terms of this statistic, provided that it meets a threshold. However, μ_{minD} does not take into account the uncertainty in the distance measurements, whereas μ_{uMAP} does.

Intuitively, correct matching of a downstream vehicle i to one of the upstream vehicles $1, \cdots, M$ should be more difficult as M increases. The next result shows this is indeed the case. Corollary 7.8.1 can be compared with (7.8.1).

Define

$$f_L(l) = p(L(d(i,j)) = l \mid \bar{\mu}(i) = j) \text{ and } G_L(l) = p(L(d(i,k)) \leq l \mid \bar{\mu}(i) \neq k), \qquad (7.8.8)$$

the pdf of $L(d(i,j))$ and the cumulative distribution function (cdf) of $L(d(i,k))$, conditional on $\bar{\mu}(i) = j \neq k$. That is, $f_L(l)$ is the pdf of $L(d)$ when d has pdf f, and $G_L(l)$ is the cdf of $L(d)$ when d has pdf g.

Corollary 7.8.1. *The maximum reward is given by the explicit formula:*

$$\rho(\mu_{uMAP}) = M\alpha \int_{\beta/\alpha}^{\infty} f_L(l)[G_L(l)]^{M-1} dl + \beta[G_L(l)]^M. \qquad (7.8.9)$$

Moreover, $\rho(\mu_{uMAP})$ decreases as M increases, keeping $M\alpha$ and β constant ($M\alpha + \beta = 1$).

Remark 3. *In the Gaussian case,*

$$f(d) = \frac{1}{\sqrt{2\pi\sigma_f^2}} \exp -\frac{(d-\mu_f)^2}{2\sigma_f^2}, \quad g(d) = \frac{1}{\sqrt{2\pi\sigma_g^2}} \exp -\frac{(d-\mu_g)^2}{2\sigma_g^2}.$$

Here μ_f and σ_f denote the mean and standard deviation of the Gaussian pdf f while μ_g and σ_g denote analogous quantities for g. To characterize μ^* it is more convenient to maximize the 'log-likelihood' $l(d)$ instead of the 'likelihood' $L(d)$:

$$l(d) = \ln L(d) = \ln \frac{\sigma_g}{\sigma_f} - \frac{(d-\mu_f)^2}{2\sigma_f^2} + \frac{(d-\mu_g)^2}{2\sigma_g^2}.$$

It is easy to check by differentiating this quadratic expression that $l(d)$ is decreasing *in d for $0 \leq d \leq \mu_g$ for the estimated parameters of Figure 7.3. Thus in this range maximizing $l(d)$ is equivalent to minimizing d. Since $[0, \mu_g]$ is likely to include the range $L(d(i,j)) \geq \beta/\alpha$ (in (7.8.6)), this means that μ^* coincides with μ_{minD} (with an appropriate threshold).*

7.8.3 Constrained matching heuristic

Define the thresholded score $W_{ij}^t = L(d(i,j))\mathbb{I}(L(d(i,j)) \geq b)$. To compute a heuristic approximation for the probability of error for constrained matching, assume that W_{ij}^t are independent with variance σ_1^2, and mean q_1 if $\bar{\mu}^*(i) = j$ and mean $-q_0$, variance σ_0^2 else, with $q_1, q_0 \geq 0$. Furthermore if $\bar{\mu}^*(i) = \tau$, then the score is $W_{i\tau} = b$. Finally, we assume that a large deviation bound holds, where μ is the mean and σ^2 the variance of the variable:

$$\mathbb{P}\left(W_{ij}^t > x\right) \leq \exp\left\{-\frac{(x-\mu)^2}{2\,\sigma^2}\right\}$$

Define the sets $\mathcal{M}_c^i = \mathcal{M}_c \cap \{\bar{\mu}(i) = j\}$ and $\mathcal{M}_c^{-i} = \mathcal{M}_c / \mathcal{M}_c^i$. Moreover, let $\mathcal{M}_c^{-r}$ be the set of matches with at most r correct matches, and $\mathcal{M}_c^r$ be the set with at least r correct matches. Extend the set of M downstream vehicles to a set of $M+N$, where the last N correspond to assignments to τ. Notice that once N elements out of the $M+N$ are selected, there is only one assignment possible (unlike the unconstrained case where $N!$ assignments are available), so the number of possible constrained matchings is:

$$|\mathcal{M}_c| = \begin{pmatrix} M+N \\ N \end{pmatrix} \leq \left(\frac{e(M+N)}{N}\right)^N = e^{\log((s+1)e)\,N},$$

where $s = M/N$. Notice that the unconstrained matching space has size $(M+1)^N$, which is higher than the above quantity, when $N \to \infty$ and $M = s\,N$. The probability of $N-r$ incorrect matches for the matching is

$$\begin{aligned}
p_e(N-r) &= \mathbb{P}\left(\max_{\mu \in \mathcal{M}_c^{-r}} \sum_{k=1}^N W_{k\mu(k)} > \max_{\tilde{\mu} \in \mathcal{M}_c^r} \sum_{k=1}^N W_{k\tilde{\mu}(k)}\right) \\
&\leq \mathbb{P}\left(\max_{\mu \in \mathcal{M}_c^{-r}} \sum_{k=1}^N W_{k\mu(k)}^t > \sum_{k=1}^N W_{k\bar{\mu}(k)}^t\right) \\
&\leq |\mathcal{M}_c| \max_{\mu \in \mathcal{M}_c^{-r}} \mathbb{P}\left(\sum_{k=1}^N W_{k\mu(k)}^t > \sum_{k=1}^N W_{k\bar{\mu}(k)}^t\right) \\
&\leq \left(\frac{e(M+N)}{N}\right)^N \exp\left\{-\frac{[(N-r)\,(q_1+q_0)]^2}{2\,N\,(\sigma_1^2+\sigma_0^2)}\right\} \\
&= \exp\left\{\log((1+s)e) - \mathrm{SNR}^2\frac{(N-r)^2}{N^2}\right\} N,
\end{aligned}$$

where $\mathrm{SNR}^2 = (q_1+q_0)^2/2(\sigma_1^2+\sigma_0^2)$. For this probability to become small for N large, the maximum r is

$$\begin{aligned}
\log((1+s)e) - \mathrm{SNR}^2\frac{(N-r)^2}{N^2} \leq 0 \Rightarrow r \leq r^*, \\
r^* = N\left(1 - \frac{\sqrt{\log((1+s)e)}}{\mathrm{SNR}}\right),
\end{aligned}$$

suggesting that for large N, at least r^* correct matches will be made by the algorithm, as long as $\mathrm{SNR} > \sqrt{\log((1+s)e)}$. Contrast this to the similar quantity obtained for uncon-

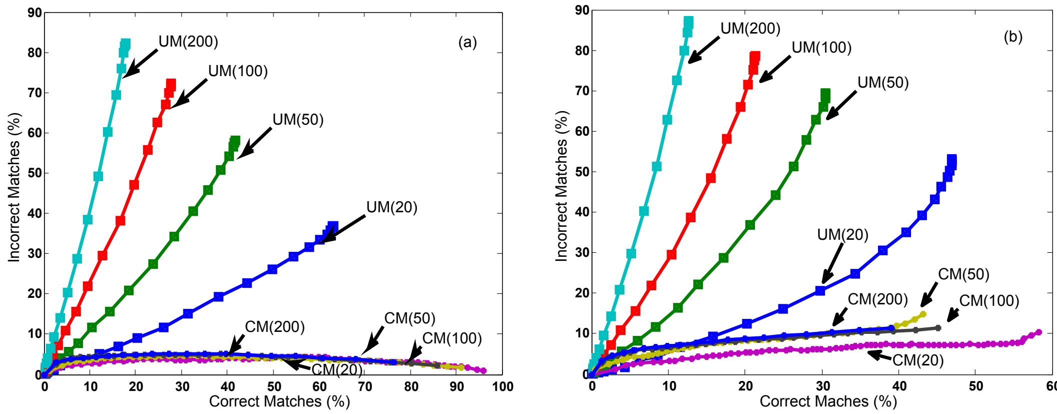


Figure 7.8. Error curve for unconstrained (UM) and constrained (CM) matching for various choices of number of vehicles (M) and (a) no overtaking or turns and (b) 10% of vehicles overtake and 25% turn.

strained matching:

$$r^* = N\,\frac{\left(SNR^2 - \log(s\,N+1)\right)}{\log(N\,e/(N\,s+1))},$$

and it suggests the advantages of the constraint, since in this case SNR needs to increase with N for the fraction of errors to remain constant.

Notice these are not true error calculations but simple heuristic analysis. A tight and useful error bound for this case requires exploring mean field analysis on a growing matching graph.

7.9 Experimental Results

We evaluate system performance using both synthetic and field test data. Synthetic 'data' are obtained by randomly generating the observation distance matrix D from the distributions in Figure 7.3(c), conditional on a randomly generated true matching $\bar{\mu}$. We can thereby vary the number of observations, turns, and overtakings or fifo violations, and compare the results of the optimal constrained and unconstrained matchings, μ_c^*, μ_u^* with $\bar{\mu}$.

7.9.1 Synthetic Data

Performance is captured in the number C_M of correct matches when the estimated match is not a turn, and the number E_M of erroneous or incorrect matches when the estimated match is not a turn. $C_M + E_M$ is the number of 'diagonal' edges and $M - C_M - E_M$ is the number of 'horizontal' edges in the minimum weight path, which we denote by μ_c^*. If T is the number of true turns, i.e., the number of horizontal edges in $\bar{\mu}$, the number of erroneously estimated turns is $E_T = |(M - C_M - E_M) - T|$. (Incorrectly assigning matches as turns reduces the number of samples available for travel time estimates, but does not necessarily bias the travel time estimate.)

The minimum weight path μ_c^* is a function of the ground truth $\bar{\mu}$, the distance matrix D, and the turn probability β in (7.5.3) which determines the weight of the horizontal edges. Hence C_M, E_M are functions of $\bar{\mu}, D, \beta$. The smaller β is, the larger is the weight of the horizontal edges, and the smaller will be their number $M - C_M - E_M$ in μ_c^* (and in μ_u^*). Indeed $M - C_M - E_M \to 0$ as $\beta \to 0$. Conversely, as the turn probability $\beta \to 1$, only horizontal edges will be included, so $C_M, E_M \to 0$.

We simulate two distinct scenarios: in Figure 7.8(a) there are no turns and no overtakings or fifo violations; in Figure 7.8(b) there are 25% turns and 10% overtaking vehicles. The figures give the performance of both optimum constrained (CM) and optimum unconstrained matchings (UM), using normalized proportions $\bar{E}_M = E_M/M$ and $\bar{C}_M = C_M/M$ to compare across different numbers of vehicles M. Also, $\bar{E}_T = |(1 - \bar{C}_M - \bar{E}_M) - T/M|$.

Figure 7.8(a) shows ROC curves for vehicle matching (correct match proportion vs. incorrect match proportion). The curves are parameterized by β: $\bar{C}_M + \bar{E}_M \to 0$ as $\beta \to 1$, and $\bar{C}_M + \bar{E}_M \to 1$ as $\beta \to 0$. When the fifo constraint is satisfied (a), the error in constrained matching is negligible (nearly 100% correct matches with 0.5% incorrect matches). Unconstrained matching is much worse, with incorrect match percentage increasing with M for a fixed correct match percentage, exactly as the analysis in Kwong et al. [2008] predicts. (The matchings in Sun et al. [1999]; Oh and Ritchie [2002]; Ndoye et al. [2008] are all unconstrained.)

Figure 7.8(b) shows that even with turns and fifo violations, constrained matching outperforms unconstrained matching, even though the latter permits overtaking. About 50% of vehicles are correctly matched, with an incorrect matching rate under 10%, independently of the number of vehicles. Including turns (25%), the maximum correct matching percentage for this scenario is 75%.

7.9.2 Field Data

Data at the test site have been continuously collected since mid-May, 2008; results for May 23 are presented here. Both lanes of the road have sensors. We split site A as 0 and 1; B as 2 and 3; C as 4 and 5; and D as 6 and 7. Link $A \to B$ comprises the fast or left lane $0 \to 2$, and the slow or right lane $1 \to 3$; the fast (even-numbered) and slow (odd-numbered) lanes of the three other links are labeled similarly.

Single lane matching refers to matching only fast lane signatures. Figure 7.9 displays the travel time distributions for single lane matching at links $A \to B$, $B \to C$ and $C \to D$ for a 30 minute time interval. The legend 'A to B 99/211' means that 99 vehicles out of a possible 211 were matched on $A \to B$, for a matching rate of 47%; the other rates are 59% and 51%. Also shown is the travel time distribution for the entire 0.9 mile segment $A \to D$, with a matching rate of 41% between vehicle signatures at A and D. (Note that there are six signals between A and D, Figure 7.1.) These matching rates, together with the results with simulated data, indicate turning rates near 25%.

The measurement system allows construction of the box plots of Figure 7.10. Each box corresponds to a 30 minute time interval that starts at the time where the box is placed. In

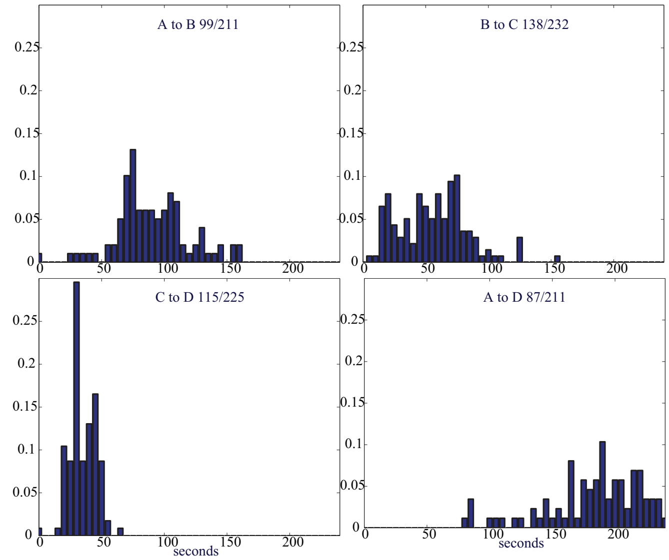


Figure 7.9: Travel time distributions for May 23, 2008, 1-1:30PM.

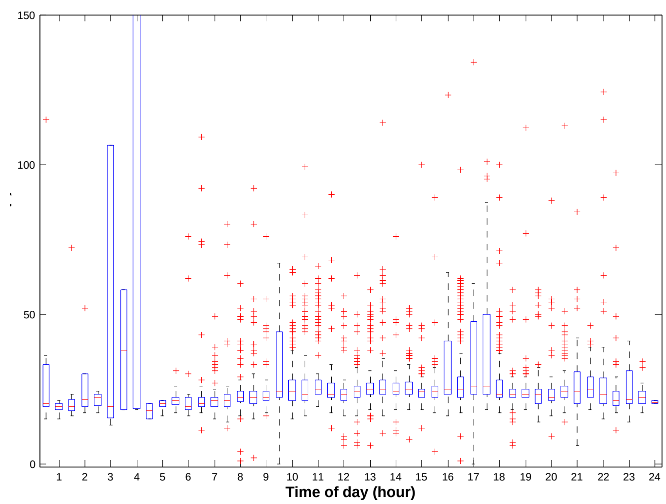


Figure 7.10. Box plot of 30 min blocks of travel time samples (in sec) for May 23, 2008, 24 hours.

the early AM hours, travel times experience immense variability due to vehicles that briefly stop. Congestion forms during 9:30-10AM and 4-6PM.

Figure 7.11 plots vehicle volume and travel time statistics (median, 20th and 70th percentiles) every 15 minutes, on the fast lane of link $C \rightarrow D$. The lane carries 440 vehicles/hour during the peak. The large variability in the 70th percentile between 9-10AM and 4-6PM is due not to a large vehicle volume but to the large number of turns.

Consider now multiple lane matching for the link $B \rightarrow C$. Figure 7.12 shows the result of applying the single lane matching algorithm *separately* for lanes $2 \rightarrow 4$ (50% matching rate), and $3 \rightarrow 5$ (38% matching rate). This matching misses vehicles that change lanes.

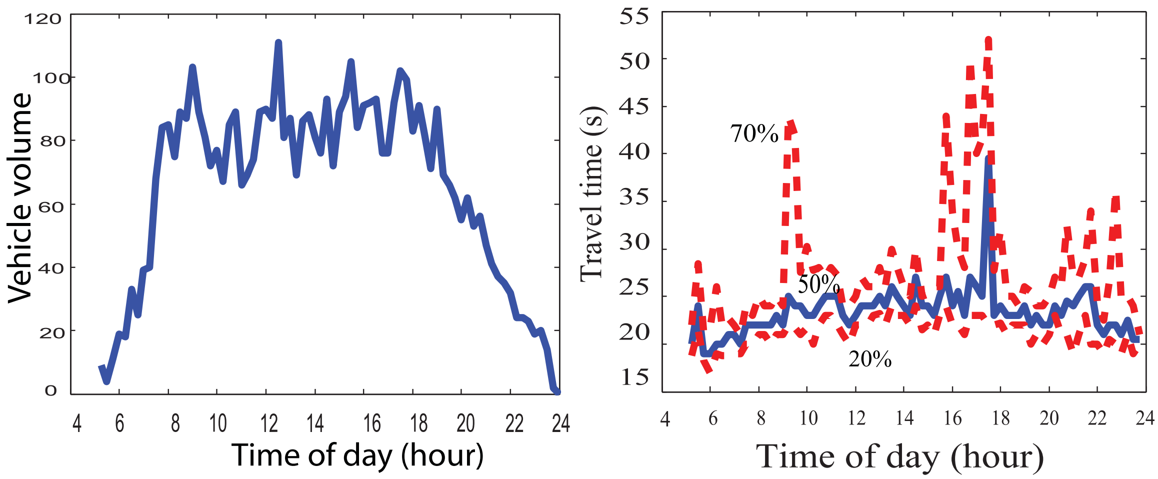


Figure 7.11. Vehicle volumes and travel time statistics for 15 minute blocks for May 23, 2008

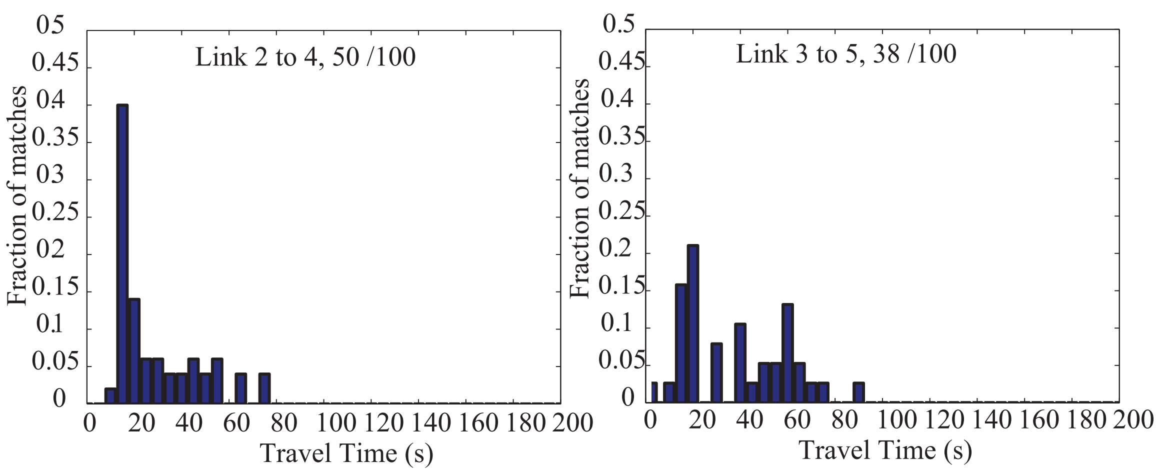


Figure 7.12: Single lane matching for lanes 2 to 4 and 3 to 5 for May 23, 7:30-8:00AM

Figure 7.13 displays the results of *multiple* lane matching for the same data. The number of vehicles matched for lane $2 \to 4$ is 46 and for lane $3 \to 5$ is 35; 21 vehicles are matched for lane $2 \to 5$ and 17 vehicles are matched for lane $3 \to 4$.

Comparison of Figures 7.12 and 7.13 shows that the distribution of travel times for $2 \to 4$ and $3 \to 5$ remains very close for both single and multiple lane matching, confirming that the estimation of travel time distributions is not affected by accounting for each lane separately. However, in the multiple lane matching a total of 119 out of 200 vehicles are matched as compared with 88 out of 200 from single lane matching.

Figure 7.14 shows the travel time distributions for the four (2×2) matches between C and D: 36 matches for $4 \to 6$, 15 matches for $4 \to 7$, 30 matches for $5 \to 7$ and 7 matches for $5 \to 6$, for a total of 88 matches, compared with 75 matches (not shown) using two separate single lane matchings.

Figures 7.13 and 7.14 indicate that vehicles that change from the ‘fast’ to the ‘slow’

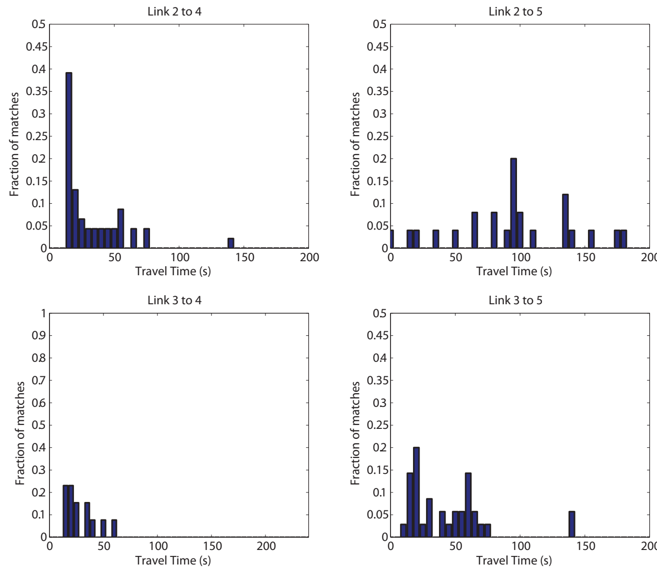


Figure 7.13: 2×2 matchings for link $B \rightarrow C$ for May 23, 7:30-8:00AM

lane ($2 \rightarrow 5, 4 \rightarrow 7$) experience a longer travel time that those that remain in the fast lane ($2 \rightarrow 4, 4 \rightarrow 6$). On the other hand, vehicles that change from the 'slow' to the 'fast' lane ($3 \rightarrow 4, 5 \rightarrow 6$) travel faster than those that remain in the slow lane ($3 \rightarrow 5, 5 \rightarrow 7$).

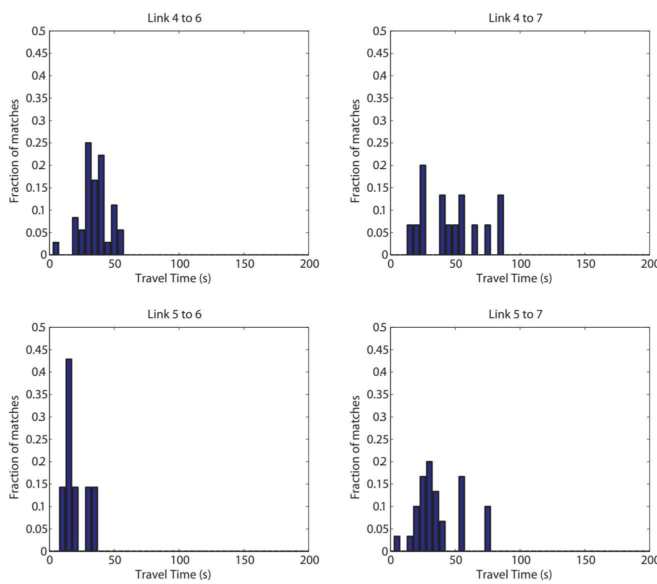


Figure 7.14: 2×2 matchings for link $C \rightarrow D$ for May 23, 07:30-8:00 AM

Figure 7.15 shows the use of the test system for traveler information. The figure plots the median travel time every half-hour over the 0.9 mile segment from midnight of 10/20/2008 to midnight of 10/24/2008. (The median travel time is set to 0 if 10 or fewer vehicles traversed the link during the half-hour.) On 10/22/2008 an accident caused the I-880 freeway to be shut down from 6:30AM until 7:25PM. Although the accident was several miles away from San Pablo Avenue, one can observe its impact in the tripling of the travel time median during the afternoon of 10/22 as southbound drivers diverted to San Pablo to avoid the freeway.

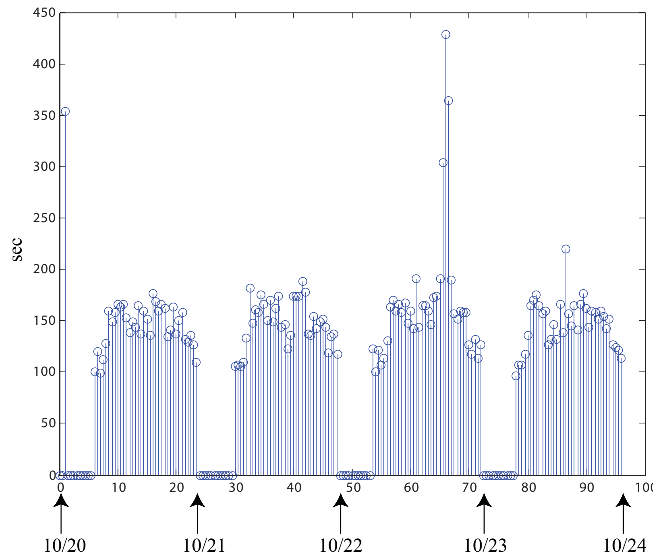


Figure 7.15. Median travel time every 30 min from 10/20/2008 to 10/24/2008. Travel time is in sec, and time is in hours beginning midnight of 10/20/2008.

7.10 Discussion

We described a system for measuring the state of a road network in real time. This appears to be the first cost-effective, scalable system that provides real-time measurements of vehicle count and individual vehicle travel times in the links of a road network. The system requires deployment of wireless magnetic sensors at locations that demarcate each link. The deployment is flexible in the way links are defined. A link may span several intersections and not all lanes in a link may be sensed. Deployment can be incremental. The system is based on anonymous matching of vehicle magnetic signatures recorded by sensors at the ends of a link.

The optimum matching algorithm relies crucially on the first in-first out (fifo) or non-overtaking constraint. Violation of this constraint reduces the number of matches, but does not bias the travel time or vehicle count estimates. The algorithm is very efficient and permits real time computation. The matching algorithm works for single-lane as well as multi-lane links. The multi-lane matching algorithm has several applications, including intersections with turns, multi-lane ramps, work zones, and weaving sections.

There are two promising directions of future work. First, suitable modifications of the matching algorithm appear to have use in several situations of mobile sensing. Second, the real-time measurement of the state can improve the performance of several traffic control algorithms. For example, real-time measurement of ramp queues can improve ramp metering control, and measurement of vehicle count in certain freeway sections can improve variable speed limit control.

7.11 Proofs

7.11.1 Proof of Theorem 7.8.1

Applying the definition (7.4.1) and using (7.4.2), the probability of a correct match, $\mu_{minD}(i) = \bar{\mu}(i)$, is

$$\begin{aligned} p(\mu_{minD}(D)(i) = j \mid \bar{\mu}(i) = j) &= p\left(d(i,j) \leq d(i,k) \forall k;\;\; d(i,j) \leq d^* \mid \bar{\mu}(i) = j\right) \\ &= \int_0^{d^*} p(d(i,j) = x | \bar{\mu}(i) = j) \prod_{k \neq j} \mathbb{P}(d(i,k) \geq x | \bar{\mu}(i) = j) dx \\ &= \int_0^{d^*} f(x) [\tilde{G}(x)]^{M-1} dx. \end{aligned}$$

The probability of an incorrect match is

$$\begin{aligned} p(\mu_{minD}(D) \neq j \mid \bar{\mu}(i) = j) &= p(d(i,j) > \min_{k \neq j} d(i,k);\;\; \min_{k \neq j} d(i,k) \leq d^* \mid \bar{\mu}(i) = j) \\ &= \int_0^{d^*} \mathbb{P}(d(i,j) > x | \bar{\mu}(i) = j) p(\min_{k \neq j} d(i,k) = x | \bar{\mu}(i) = j) dx \\ &= (M-1) \int_0^{d^*} [\tilde{G}(x)]^{M-2} g(x) \tilde{F}(x) dx. \end{aligned}$$

To arrive at the last equality, we use the facts that $\mathbb{P}(d(i,j) > x \mid \bar{\mu}(i) = j) = \tilde{F}(x)$ and

$$\mathbb{P}(\min_{k \neq j} d(i,k) \leq x \mid \bar{\mu}(i) = j) = 1 - \mathbb{P}(d(i,k) \leq x, k \neq j \mid \bar{\mu}(i) = j) = 1 - [\tilde{G}(x)]^{M-1},$$

which, upon differentiating with respect to x, gives the density

$$p(\min_{k \neq j} d(i,k) = x \mid \bar{\mu}(i) = j) = (M-1)[\tilde{G}(x)]^{M-2} g(x).$$

Lastly, the probability that $\mu_{minD}(i) = \tau$ is

$$\begin{aligned} p(\mu_{minD}(D)(i) = \tau \mid \bar{\mu}(i) = j) &= p\left(d(i,j) \geq d^*;\;\; d(i,k) \geq d^*, k \neq j \mid \bar{\mu}(i) = j\right) \\ &= \tilde{F}(d^*)[\tilde{G}(d^*)]^{M-1}. \end{aligned}$$

This proves (7.8.1)-(7.8.3). □

7.11.2 Proof of Theorem 7.8.2

For any matching function μ, we can express $\rho(\mu)$ as

$$\begin{aligned} N\rho(\mu) &= \sum_{i=1}^{N}\big[\sum_{j=1}^{M} p(\mu(D)(i)=j \mid \bar{\mu}(i)=j)p(\bar{\mu}(i)=j) \\ &\quad + p(\mu(D)(i)=\tau \mid \bar{\mu}(i)=\tau)p(\bar{\mu}(i)=\tau)\big] \qquad (7.11.1)\\ &= \sum_{i=1}^{N}\big[\alpha\sum_{j=1}^{M}\int p(\mu(D)(i)=j \mid D)p(D \mid \bar{\mu}(i)=j)dD \\ &\quad + \beta\int p(\mu(D)(i)=\tau \mid D)p(D \mid \bar{\mu}(i)=\tau)dD\big]. \qquad (7.11.2)\end{aligned}$$

From (7.4.2)-(7.4.4) and (7.8.5)

$$\begin{aligned} &p(D \mid \bar{\mu}(i)=j) = p(D_i \mid \bar{\mu}(i)=j)p(D_{-i}) = L(d(i,j))\gamma(D_i)p(D_{-i}),\\ &p(D \mid \bar{\mu}(i)=\tau) = \gamma(D_i)p(D_{-i}),\end{aligned}$$

in which $p(D_{-i}) = \prod_{l\neq i} p(D_l)$. Substitution into (7.11.2) yields

$$\begin{aligned} N\rho(\mu) &= \sum_{i=1}^{N}\big[\alpha\sum_{j=1}^{M}\int p(\mu(D)(i)=j \mid D)L(d(i,j))\gamma(D_i)p(D_{-i})dD \\ &\quad +\beta\int p(\mu(D)(i)=\tau \mid D)\gamma(D_i)p(D_{-i})dD\big], \qquad (7.11.3)\end{aligned}$$

We see from (7.11.3) that $\mu^*(D)(i)$ is given by that j which maximizes the integrand. This selection leads to (7.8.6). □

7.11.3 Proof of Corollary 7.8.1

From (7.11.1)

$$N\rho(\mu^*) = \sum_{i=1}^{N}\big[\alpha\sum_{j=1}^{M} p(\mu^*(D)(i)=j \mid \bar{\mu}(i)=j) + \beta p(\mu(D)(i)=\tau \mid \bar{\mu}(i)=\tau)\big]. \qquad (7.11.4)$$

From (7.8.6)

$$\begin{aligned} p^{\alpha}(M) &= p(\mu^*(D)(i)=j \mid \bar{\mu}(i)=j),\\ &= p(L(d(i,j)) \geq \max\{L(d(i,k)), k\neq j, \beta/\alpha\} \mid \bar{\mu}(i)=j), \qquad (7.11.5)\\ p^{\beta}(M) &= p(\mu(D)(i)=\tau|\bar{\mu}(i)=\tau) = p(\beta/\alpha \geq \max\{L(d(i,k))\}|\bar{\mu}(i)=\tau). \qquad (7.11.6)\end{aligned}$$

In (7.11.5)-(7.11.6), the random variables $d(i,j)$ and $d(i,k)$ are independent; $d(i,j)$ is distributed according to f and the $d(i,k)$ are all distributed according to g. The random variable on the right hand side of the inequalities in (7.11.5)-(7.11.6) increases with M, because it is the maximum of more random variables, whereas the random variable on the left hand side does not change with M. Thus both probabilities $p^{\alpha}(M)$ and $p^{\beta}(M)$ decrease with M. So, from (7.11.4) $\rho(\mu^*) = M\alpha p^{\alpha}(M) + \beta p^{\beta}(M)$ decreases with M.

We can use (7.11.5), (7.11.6) to calculate $\rho(\mu^*)$. From (7.11.5),

$$\begin{aligned} p^{\alpha}(M) &= p(L(d(i,j) \geq \max_{k \neq j} L(d(i,k)), L(d(i,j) \geq \beta/\alpha \mid \bar{\mu}(i) = j), \\ &= \int_{\beta/\alpha}^{\infty} f_L(l)[G_L(l)]^{M-1} dl, \\ p^{\beta}(M) &= p(\max_{k} L(d(i,k) \leq \beta/a \mid \bar{\mu}(i) = \tau) = [G_L(l)]^M. \end{aligned}$$

This gives (7.8.9). □

Chapter 8

Monitoring Load Impact in Roads

8.1 Introduction

Weighing stations along the highway are used to check truck weights. These stations require separate areas along the highway where trucks stop be to weighed. Due to their high cost and also operational issues, such as requiring trucks to reduce speed and queue up for some time, there are few such stations. In Weigh-In-Motion stations (WIM) trucks can be weighed as they slowly move along [Cebon, 1999]. This technology is deployed in roadside weighing stations as a replacement to traditional weigh-in stations.

Traditional stations use bending plate, piezoelectric, or load cell sensors to measure the vertical forces applied by axles to sensors [Cebon, 1999]. The stations require a controlled environment and continuous calibration to reliably estimate static axle loads. Additional calculations are then performed to transform the static axle load estimates into the dynamic load that the pavement actually experiences. The latter calculations are based on models of vehicle-pavement interactions. These interaction models are rarely if ever calibrated for individual WIM stations [Gonzalez et al., 2003; Stergioulas et al., 2000].

This chapter explores a very different approach. The system comprises a network of sensor nodes (SN) and an access point (AP). Each SN assembles a single- or double-axis MeMS accelerometer, a microprocessor, flash memory, a radio, and an electronic PC board that interconnects these components. A pair of AA batteries powers the assembly. The SN is encased in a 3” cube embedded in the pavement. The processed data are sent by the SN radio to the AP, situated on the side of the road. The AP may record the data locally or forward them to a remote site.

The SNs directly measure the vibration (acceleration) of the pavement under them. It may also be possible to process the SN data to estimate the truck axle weight and spacing, classification, and speed. The installed cost of SN and AP are a fraction of the cost of current WIM stations. Figure 8.1 shows a possible deployment.

Moses [1979] and Leming and Stalford [2003] deal with a similar problem. But the application is restricted to bridges, and the model does not consider transient pavement effects. The models are much simpler since they rely on modal estimation, and give no accuracy guarantees. In bridges the responses have higher amplitudes as well as the decay is slower, making it possible for a modal estimation procedure to work. But among the

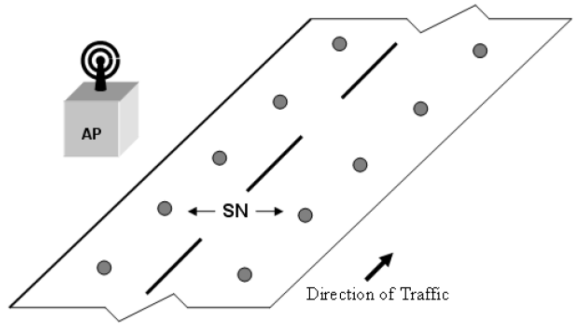


Figure 8.1. Deployment of proposed WIM system on a multi-lane freeway or bridge location. The sensor nodes are only 3” in diameter; the access point is a 5” cube. Data from sensors nodes are sent to the access point via radio. The sensor nodes and access points are drawn at an exaggerated scale relative to lane width.

many constraints, only a single truck at a time can pass through the bridge, which makes it an impractical solution.

We in turn, develop a different approach. We start our study with the analysis of a literature validated PDE model of the pavement [Sousa et al., 1988; Hardy and Cebon, 1993; Cebon, 1999]. We compute a closed form solution of the pavement response under truck motion. We then design optimal weight estimation algorithms using the closed form solution. Along the way we discuss issues such as required precision for SN, energy consumption for stand alone operation and communication requirements, as well as efficient algorithmic implementations.

The chapter also holds independent interest due to the closed form solution derivation presented. To our knowledge no similar derivations exist in the literature for the situation presented. One advantage of the closed solution in the present case is that for usual parameters the system is stiff, and numerical integration poses serious difficulties. We attempted using some popular PDE solvers for computing the solution and obtained poor approximations.

Furthermore, our estimation problem aims at estimating a finite parameter, from infinite measurements or point measurements of an distributed dimensional system, contributing to the literature on estimation in systems described by partial differential equations [Gerdin et al., 2007; Ewing et al., 1999; Baumeister et al., 1997; Ljung, 1999].

The chapter is organized as follows. Section 8.2 states the pavement model and the estimation problem of interest. Section 8.3 develops an analysis of the model, including a closed form approximation that is of independent interest. In Section 8.4 we present methods for estimating the load under various setups. The presented method is optimal and can be used to gauge other methods used in practice. Section 8.5 introduces some system design considerations, regarding sensor placement and estimation methods. We discuss simulation results using real world pavement parameters in Section 8.6. The proofs of all theorems of the chapter are presented in Section 8.7. Concluding remarks are presented in Section 8.8.

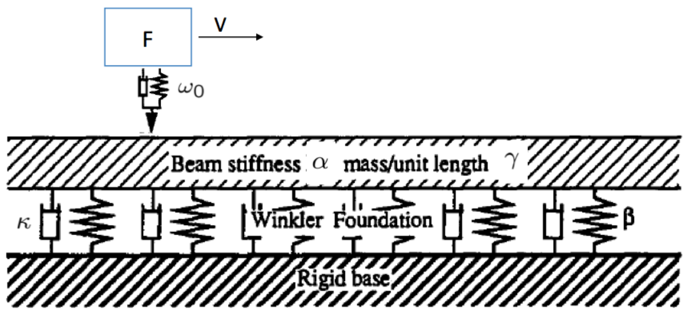


Figure 8.2: Euler beam model for a roadway and a quarter-car axle model.

8.2 Problem statement

We consider the model of a road as an Euler beam with elastic foundation with a moving load. The vertical non-stationary force acting on the the road (beam) is due to transient dynamic loads applied through tires of moving vehicles [Hardy and Cebon, 1994; Markow et al., 1988; Sousa et al., 1988].

The conventional *model of the equation of motion* of a one-dimensional damped beam is (see [Cebon, 1999; Fryba, 1972; Hardy and Cebon, 1993; Rao, 2007] and Figure 8.2)

$$EI\frac{\partial^4 y}{\partial x^4} + \gamma\frac{\partial^2 y}{\partial t^2} + \kappa\frac{\partial y}{\partial t} + \beta y = F(x,t). \tag{8.2.1}$$

Here x and $y(x,t)$ are the horizontal position (along the road) and the vertical displacement (of the pavement), and $F(x,t)$ is the applied force at position x and time t. The displacement y varies in the domain $y \in R$, and the position x varies within the interval $[0, L]$. The standard road beam model makes the assumption $\beta > \kappa$, which results in the road pavement having natural frequencies [Sun and Kennedy, 2002].

The basic force resulting from a truck moving at velocity V is modeled as the *moving excitation* [Hardy and Cebon, 1994]

$$F(x,t) = F\cos(\omega_0 t) \times \delta(x - Vt), \tag{8.2.2}$$

where V is the velocity of the point of application of the force $F\cos(\omega_0 t)$ with magnitude F and frequency ω_0. The magnitude and frequency are determined by the vehicle's suspension system. Typical values are $F = 50000\,N$ and $\omega_0 = 2\pi\, f_0$, where f_0 is between 1 Hz and 3 Hz [Fu and Cebon, 2002; Chen et al., 2002b]. Real trucks have force excitations composed of a linear combination of basic components

$$\begin{aligned} F(x,t) &= P(t) \times \delta(x - Vt), \\ P(t) &= F(\sum_{r=0}^{W} P_r \cos(\omega_r t)), \end{aligned} \tag{8.2.3}$$

where the number of components W and the frequencies ω_i depend on the truck suspension system type. For quarter car models, $W = 2$, $\omega_0 = 0$, ω_1 is in the given range [Chen et al., 2002b; Stergioulas et al., 2000]. For walking beam models, $W = 3$, with $\omega_0 = 0$ [Stergioulas et al., 2000]. The values of P_r are usually assumed to be equal or have a fixed proportion.

We also consider a *fixed excitation* applied at a point x_0 at time t_0

$$F(x,t) = F\delta(t - t_0)\delta(x - x_0). \tag{8.2.4}$$

The point of application of the force at time t is $x_a = Vt$. It starts to move at time $t_0 = 0$, from position $x_a(0) = 0$. This model is an approximation of the standard quarter car model [Stergioulas et al., 2000; Hardy and Cebon, 1994; Chatti and Yun, 1996]. We opt for the approximation since in real applications, the quarter car model has too many parameters compared to the expected uncertainty [Cebon, 1999; Hardy and Cebon, 1994].

We consider two types of boundary and initial condition sets for solving the equation of motion: **Model I** and **Model II**. In **Model I** we consider equation (8.2.1) for the elastic beam taking it to be finite of length L with its ends freely hinged at $x = 0$ and at $x = L$. In **Model II** we consider equation (8.2.1) for the elastic beam taking it to be semiinfinite, with its end freely hinged at $x = 0$. In both cases the beam is initially at rest.

The *observation* is given by the measurement equation

(A) *Pointwise displacement sensor measurement* of $y(x^*, t),\ t \in [0, \tau]$

$$z(t) = y(x^*, t) + \xi(t). \tag{8.2.5}$$

(B) *Pointwise acceleration sensor measurement* of $\ddot{y}(x^*, t),\ t \in [0, \tau]$

$$z(t) = \frac{\partial^2 y(x^*, t)}{\partial t^2} + \xi(t) = \ddot{y}(x^*, t) + \xi(t). \tag{8.2.6}$$

Throughout the text y' denotes the spatial derivative $\partial y/\partial x$ in x and $\dot{y}$ – the time derivative $\frac{\partial y}{\partial t}$ in t.

In (8.2.5) and (8.2.6), x^* is the point of measurement and $\xi(t)$ is the *measurement noise*, with $\xi(t)$ white noise with variance σ_ξ^2 (**White Noise Model**) [Grimmett and Stirzaker, 1992; Hayes, 1996] or $\xi(t) = \eta(t) + u(t)$, $|u(t)| \leq \mu$, $\mu > 0$, and $\eta(t)$ white noise with variance σ_η^2 (**Bounded Noise Model**). White noise arises in applications due to electrical and transducer noise in typical sensors used for measurements [Ljung, 1999]. Bounded noise arises due to drift observed in some sensor modalities. Typically, we also observe white noise together with the bounded noise.

In general, continuous time measurements are not available. But we sample at a high sampling rate, therefore the performance loss due to discretization is small. Also, we allow measurements to be made at several points along the highway, at $x_1, ..., x_N$. The vector of observed functions is denoted by $\mathbf{z}(t)$.

Based on this model we identify three problems.

Problem 1[Force estimation] Estimate the value F on the basis of the available measurement $\mathbf{z}(t), t \in [0, \tau]$. The parameters $EI, \gamma, \kappa, \beta$, in (8.2.1) are all taken as known.

Problem 2[Class detection] Suppose there are m nonintersecting intervals $\mathcal{F}_k \subset \mathbb{R}_+$:

$$\{\mathcal{F}_j \cap \mathcal{F}_k \mid j, k = 1, ..., m; k \neq j\} = \emptyset.$$

On the basis of measurements $z(t), t \in [0, \tau]$ identify to which interval $\mathcal{F}_k$ does F belong.

Problem 3[Calibration] Given available measurements $\mathbf{z}(t)$ and an input with known dynamic force (F, ω_0), estimate the parameters of the road model.

Observe that these problems deal with the identification of a finite number (F) through measurement of an infinite–dimensional process [Ljung, 1999]. Also notice that the bounded noise model is more naturally related to **Problem 2** and the white noise model is better related to **Problem 1**. In this chapter we focus on **Problems 1** and **2**. **Problem 3** will be addressed separately.

8.3 System analysis

In this section we explore the behavior of the system given in equation (8.2.1). First an analytic solution of the response of the system is computed under the assumption the beam is finite. Next an extension for the semi-infinite beam is presented, and the solution can be reduced to a particular setting of the finite beam solution. We also consider an analytic approximation to the complete solution.

Kenney [1954], Sun and Kennedy [2002] and Chan et al. [1999] propose approximations of the beam response to moving loads. These approximations are different, in the sense that no guarantees on the error size of the approximation are computed, as well as the applied loads have different characteristics. Furthermore, the modulated moving characterization of the system response is not as clearly identifiable in some of these approximations. In some sense, the work in this section complements and extends previous approximation methodologies.

Chatti and Yun [1996] also proposes a numerical approximation methodology to compute pavement responses, based on a state-space model [Oppenheim et al., 1997]. The main difficulty with this approach for our purposes is that calculating the numerical responses in real-time is much more computationally intensive than the formulas derived in this section.

8.3.1 Finite beam

Let us now consider equation (8.2.1) for the elastic beam taking it to be *of finite length* L, with both ends freely hinged at $x = 0$ and $x = L$ [Rao, 2007]. Then we have

$$y(0,t) = y(L,0) = 0, \; y''(0,t) = y''(L,0) = 0, \; t \geq 0. \tag{8.3.1}$$

We assume the beam to be originally at rest, in its equilibrium position:

$$y(x,0) = 0, \dot{y}(x,0) = 0, \quad x \geq 0. \tag{8.3.2}$$

Therefore, the motion of the beam will arise only due to the external force $F(x,t)$. For the moving excitation, we have the next result.

Theorem 8.3.1. *Consider the system in equation (8.2.1) with the boundary conditions (8.3.1) and (8.3.2). The response of the system excited by $F(x,t) = F\cos(\omega_0 t) \times \delta(x - Vt)$ is as follows.*

(a) The exact solution is given by:

$$y(x,t) = \frac{2}{L}\sum_{m=0}^{\infty} Y_m(t)\sin\left(\frac{\pi m x}{L}\right), \tag{8.3.3}$$

where $Y_m(t)$ is composed of two parts: $Y_{tr,m}(t)$, the transient natural beam response, and $Y_{ss,m}(t)$, the "steady-state" component, corresponding to the response of the beam to the excitation,

$$\begin{aligned}
Y_m(t) &= Y_{tr,m}(t) + Y_{ss,m}(t), \qquad (8.3.4)\\
Y_{ss,m}(t) &= \frac{F_0}{2}\left\{|F_{a,m}|\sin(\omega_{a,m}t + \angle|F_{a,m}|) + |F_{b,m}|\sin(\omega_{b,m}t + \angle|F_{b,m}|)\right\},\\
Y_{tr,m}(t) &= \frac{F_0}{2\Omega_m}e^{-kt}\left(|C_{a,m}|\sin(\Omega_m t + \angle C_{a,m}) + |C_{b,m}|\sin(\Omega_m t + \angle C_{b,m})\right),\\
F_0 &= \frac{F}{\gamma},\quad k = \frac{\kappa}{\gamma},\quad \omega_m^2 = (\alpha(\pi m/L)^4 + \beta)/\gamma,\quad \Omega_m^2 = \omega_m^2 - k^2\\
\omega_{a,m} &= \frac{\pi m}{L}V + \omega_0,\quad \omega_{b,m} = \frac{\pi m}{L}V - \omega_0,\\
C_{a,m} &= \frac{1}{k^2 - 2k\Omega_m i - \Omega_m^2 + \omega_{a,m}^2},\\
C_{b,m} &= \frac{1}{k^2 - 2k\Omega_m i - \Omega_m^2 + \omega_{b,m}^2},\\
F(s,m) &= s^2 + 2ks + \omega_m^2,\\
F_{a,m} &= F(i\omega_{a,m}, m)^{-1},\\
F_{b,m} &= F(i\omega_{b,m}, m)^{-1} = F^*(i\omega_{a,m}, -m)^{-1}.
\end{aligned}$$

(b) We have

$$\lim_{L\to\infty} y(x,t) = F_0 Re[\psi^*(Vt - x)e^{j\omega_0 t}] + O\big(e^{-k\,t}\big),$$

where

$$\begin{aligned}
\psi^*(t) &= \frac{1}{2\pi\, i}\int_{-\infty}^{\infty}\Omega(s)^{-1}e^{s\,t}ds,\\
\Omega(s) &= \alpha/\gamma s^4 + V^2 s^2 + (2\omega_0 V i + 2kV)s + (\beta/\gamma - \omega_0^2 + 2k\omega_0 i). \qquad (8.3.5)
\end{aligned}$$

(c) The response of system (8.2.1) to the fixed excitation $F(x,t) = F\delta(t - t_0)\delta(x - x_0)$ is

given by

$$
\begin{aligned}
y(x,t) &= \frac{2}{L}\sum_{m=0}^{\infty}\tilde{Y}_m(t-t_0)\cos\left(\frac{\pi m(x-x_0)}{L}\right)-\frac{2}{L}\sum_{m=0}^{\infty}\tilde{Y}_m(t-t_0)\cos\left(\frac{\pi m(x+x_0)}{L}\right),\\
\tilde{Y}_m(t) &= F_0\Omega_m^{-1}e^{-kt}\sin(\Omega_m t)u(t),
\end{aligned}
$$

where the Heaveside function $u(t)=1$ for $t\geq 0$ and $u(t)=0$ for $t<0$.

The solution in Theorem 8.3.1 does not solve for the truck forcing term (equation (8.2.2)), but since the PDE is linear, the result is easily extended.

Corollary 8.3.1. *Let*

$$h(x,t|\omega_0,V)=\frac{1}{\gamma}Re[\psi^*(Vt-x)e^{j\omega_0 t}], \tag{8.3.6}$$

where ψ^* is computed according to equation (8.3.5) with parameters ω_0 and V. Then the response of system (8.2.1) to the truck forcing term (equation (8.2.2)) is given by

$$\lim_{L\to\infty} y(x,t)=F\left(h(x,t|0,V)+h(x,t|\omega_0,V)\right)+O\big(e^{-k\,t}\big). \tag{8.3.7}$$

The qualitative behavior of the system can be explored using Theorem 8.3.1(c). The closed form solution for the displacement $y(x,t)$ can be obtained by computing the inverse Laplace transform [Oppenheim et al., 1997] of $\Omega(s)^{-1}$ as shown. Inverting Laplace transforms requires the specification of the region of convergence (ROC) of the integral [Oppenheim et al., 1997]. Since the system we are dealing is a physical system, the solution obtained from the inversion computation should be a solution with bounded energy.

The standard inversion procedure starts by computing the roots of the rational transfer function to be inverted. In the present case this corresponds to finding the values λ_i such that $\Omega(\lambda_i)=0$, which amounts to solving for the roots of a fourth order polynomial. Then we can use the partial fraction expansion, and assuming no repeated roots, obtain the decomposition

$$\Omega(s)^{-1}=\sum_{i=1}^{4}\frac{A_i}{s-\lambda_i},$$

where A_i are the partial fraction expansion coefficients. Using the bounded energy condition as the region of convergence rule, the inverse Laplace transform states:

$$\frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{A_i}{s-\lambda_i}e^{s\,t}ds=\begin{cases} A_ie^{\lambda_i t}\,u(t) & Re[\lambda_i]\leq 0\\ -A_ie^{\lambda_i t}\,u(-t) & Re[\lambda_i]>0\end{cases}.$$

Since the coefficient of s^3 in the polynomial $\Omega(s)$ is zero, we must have $\lambda_1+\lambda_2+\lambda_3+\lambda_4=0$, which implies that either $Re[\lambda_i]=0$ for all the roots, or else, there are roots with $Re[\lambda_i]>0$ and with $Re[\lambda_i]<0$. The beam is damped, therefore not all roots can have $Re[\lambda_i]=0$. Without loss of generality, let us assume that $Re[\lambda_1]>0$, $Re[\lambda_2]>0$, $Re[\lambda_3]<0$ and

$Re[\lambda_4] < 0$. Then the function $\psi^*(t)$ in Theorem 8.3.1(c) can be computed as

$$\psi^*(t) = -A_1 e^{\lambda_1 t} u(-t) - A_2 e^{\lambda_2 t} u(-t) + A_3 e^{\lambda_3 t} u(t) + A_4 e^{\lambda_4 t} u(t).$$

The beam deflection response is essentially a traveling wave shaped by $\psi^*(t)$. The shape of ψ^* implies that there is a decaying behavior for large $t > 0$ and for small $t < 0$. Moreover, the time t^* at which the truck goes over the location x is $t^* = x/V$. At this time, the value of the wave shape is $\psi^*(0)$. This also implies that at location x the beam experiences some displacement even before the truck arrives at that location, since $\psi^*(t) \neq 0$ for $t < 0$. This displacement is caused by the sum of the excitations just prior to the truck arriving at that location. The whole response is modulated by the truck's suspension system frequency. This accurately captures the important phenomena observed in the more complex quarter car model [Cebon, 1999; Hardy and Cebon, 1994].

A better comprehension of the behavior of the roots can be gained by looking at the system response for large truck speeds. Consider the transform $s' = i\omega_0 + V s$. Then, the polynomial can be written as:

$$\begin{aligned} \Omega(s') &= s'^2 + 2k\, s' + \frac{\beta}{\gamma} + \frac{\alpha}{\gamma V^4}(s' - i\,\omega_0)^4, \\ &\approx s'^2 + 2ks' + \frac{\beta}{\gamma}. \end{aligned}$$

Thus the roots of the original $\Omega(s)$ at high speed are given by

$$\lambda_{1,2} = \frac{-k \pm \sqrt{\beta/\gamma - k^2}\,i - \omega_0 i}{V}.$$

The displacement can be computed as

$$y(x,t) \approx \frac{F}{\gamma V \sqrt{\beta/\gamma - k^2}}\, e^{-k\,(t - \frac{x}{V})} \sin\left(\sqrt{\beta/\gamma - k^2}\left(t - \frac{x}{V}\right)\right) \cos\left(\frac{\omega_0\, x}{V}\right). \tag{8.3.8}$$

The exponential decay of the solution is at a rate $-k$ (notice the normalization by V) independent of speed, and the fundamental frequencies of the system is $\sqrt{\beta/\gamma - k^2}$. At high speeds, the suspension system modulation frequency ω_0 only affects the amplitude of the response spatially.

To conclude the discussion, the solution for a fixed excitation (Theorem 8.3.1 (d)) can be related to the moving excitation solution. Let $t_0 = x/V$ and $x_0 = Vt$ in the fixed excitation. This is similar to having an unmodulated moving impulse without iterating through the physical system. Then:

$$y(x,t) \;=\; \frac{2}{L}\sum_{m=0}^{\infty} \tilde{Y}_m(Vt - x)\cos\left(\frac{\pi m(Vt - x)}{L}\right) - \frac{2}{L}\sum_{m=0}^{\infty} \tilde{Y}_m(Vt - x)\cos\left(\frac{\pi m(x + Vt)}{L}\right),$$

which is the solution for the moving excitation when $\omega_0 = 0$.

8.3.2 Semi-infinite beam

For completeness, we consider system (8.2.1) for the elastic beam taking it to be semi-infinite, with its end freely hinged at $x = 0$ [Fryba, 1972; Rao, 2007]. The important observation is that the obtained solution is equivalent to the solution obtained for a finite beam of length L by letting $L \to \infty$, confirming the validity of our approximation. The computation of the current solution, though, relies on a continuous Fourier transform decomposition [Fryba, 1972; Rao, 2007].

Since the end is freely hinged, the boundary conditions are given by

$$y(0,t) = 0,\ y''(0,t) = 0,\ t \geq 0. \tag{8.3.9}$$

We assume the beam to be originally at rest, in its equilibrium position:

$$y(x,0) = 0, \dot{y}(x,0) = 0,\ \ x \geq 0. \tag{8.3.10}$$

Furthermore, also assume that the derivatives $y^{(k)}(x,t),\ k = 1,\ldots,3$, vanish at $x = \infty$, which is equivalent to the limit of the condition we used at $x = L$ for a finite beam.

Theorem 8.3.2. *The exact solution for the moving excitation hinged semi-infinite beam problem is given by*

$$y(x,t) = (2/\pi)^{\frac{1}{2}} \int_0^\infty Y_\xi(t) \sin(\xi x) d\xi, \tag{8.3.11}$$

where

$$Y_\xi(t) = Y_{\frac{\xi L}{\pi}}(t), \tag{8.3.12}$$

and Y_m is given in Equation (8.3.4).

8.4 Estimating the load

Problem 1 concerns the estimation of the weight under the **White Noise Model**, given that the parameters of the highway are known [Chan et al., 1999]. The input to the highway system is a truck, whose corresponding forcing model is given by

$$F(x,t) = F(1 + \cos(\omega_0[t - t_0] + \phi)) \times \delta(x - V[t - t_0]), \tag{8.4.1}$$

where we have included the phase term ϕ to account for the uncertainty in the initial conditions of the suspension system for the truck and t_0 to account for the unknown initial starting time of the truck.

We assume two types of situations. In the coherent estimation problem, we assume that the truck parameters t_0, ϕ, ω_0 and V are known, and we have to estimate the value of F. In a certain sense, this is the best possible situation, since the whole parametrization of the problem is known.

The conditions are progressively relaxed, assuming first that t_0 is unknown, both ϕ and t_0 are unknown, and finally t_0, ω_0 and ϕ are unknown. We assume that the speed V can be measured, but at the end of the section we study the sensitivity of our problem towards this parameter. The problems with less information are categorized as non-coherent estimation problems, and as we will see there are considerable noise tradeoffs in these cases. One feature of non-coherent estimation is that the identifiability of the force F depends on the information set. Denote the information set as $\mathcal{I}$, such as in $\mathcal{I} = [V, \omega_0, \phi]$.

The first important observation concerns the role of the measurement equation. For the methods presented here, the fact that displacement is being measured (equation (8.2.5)) or acceleration is being measured (equation (8.2.6)) does not change the methodology. The error rates of the proposed methods though could be different since they depend on the amount of energy measured by the transducer relative to the amount of noise. To normalize our error computations, we define the signal-to-noise ration (SNR) of the measurement system as [Hayes, 1996; Ljung, 1999]

$$\begin{aligned} \text{SNR} &= \frac{P_s - P_\xi}{P_\xi}, \\ P_\xi &= \mathbb{E}\left[\int_0^\tau \xi(t)^2\,dt\right],\ P_s = \mathbb{E}\left[\int_0^\tau z(t)^2\,dt\right], \end{aligned} \tag{8.4.2}$$

which is a surrogate measure of the relative amount of information provided by the sensor. We assume without loss of generality that the measurement is displacement. Also, for the remainder of the section, denote by $h(t, x|V, \omega_0, \phi)$, the response to the forcing equation (8.4.1):

$$h(x, t|\omega_0, V, \phi, t_0) = \frac{1}{\gamma} Re[\psi_1^*(V[t-t_0]-x)e^{j(\omega_0(t-t_0)+\phi)}] + \frac{1}{\gamma} Re[\psi_0^*(V[t-t_0]-x)e^{j\phi}], \tag{8.4.3}$$

where ψ_1^* is computed according to equation (8.3.5) with parameters ω_0 and V, and ψ_0^* computed with parameters $\omega_0 = 0$ and V. This result can be demonstrated with a minor modification in the proof of Theorem 8.3.1.

Furthermore, we allow the observation to be a scalar function $z(t)$, at a single point in space x^*, or more generally, $z_i(t)$, at points in space x_i^* for $i = 1, ..., I$, implying measurements with I sensors. Procedures for different information sets are shown in Theorem 8.4.1. Notice that as more parameters become unknown, the complexity of the procedure increases.

Theorem 8.4.1. *Given the complete information set $\mathcal{I}_0 = [V, \omega_0, \phi, t_0]$, the optimal mean square estimate of the parameter F is*

(a) For a single observation at x^*

$$\hat{F} = \int_0^\infty \frac{z(t)h(t, x^*|\omega_0, V, \phi, t_0)}{||h(t, x^*|\omega_0, V, \phi, t_0)||^2}dt, \tag{8.4.4}$$

and the Mean Square Error (MSE) is given by

$$\mathbb{E}[(\hat{F}-F)^2] = \frac{\sigma^2}{||h(t,x^*|\omega_0,V,\phi,t_0)||^2}, \tag{8.4.5}$$

$$= \frac{1}{SNR(x^*)} \tag{8.4.6}$$

(b) For multiple observations x_i^*, $i=1,...,I$:

$$\hat{F} = \frac{\sum_{i=1}^{I}\int_0^\infty z_i(t)h(t,x_i^*|\omega_0,V,\phi,t_0)dt}{\sum_{i=1}^{I}||h(t,x_i^*|\omega_0,V,\phi,t_0)||^2}, \tag{8.4.7}$$

The MSE is

$$\mathbb{E}[(\hat{F}-F)^2] = \frac{1}{\sum_{i=1}^{I} SNR(x_i^*)}, \tag{8.4.8}$$

Given the information set $\mathcal{I}_1 = [V,\omega_0,\phi]$,the energy estimate of the parameter F is

(c) For a single observation at x^*

$$\hat{F} = \left[\frac{\int_0^\tau z(t)^2\,dt}{||h(t,x^*|\omega_0,V,\phi,0)||_\tau^2}\right]^{\frac{1}{2}}, \tag{8.4.9}$$

(d) For multiple observations x_i^*, $i=1,...,I$:

$$\hat{F} = \frac{\sum_{i=1}^{I}\int_0^\infty z_i(t)^2\,dt}{\sum_{i=1}^{I}||h(t,x_i^*|\omega_0,V,\phi,0)||_\tau^2}, \tag{8.4.10}$$

Given the information set $\mathcal{I}_i$, where $\mathcal{I}_0$ represents the complete information set, denote by $\mathcal{I} = \mathcal{I}_0 - \mathcal{I}_i$ the set of unknown parameters. Then

(e) For a single observation at x^*, the least-squares estimator is

$$\hat{\mathcal{I}} = \operatorname*{argmax}_{\mathcal{I}} \frac{\left(\int_0^\infty z(t)h(t,x^*|\omega_0,V,\phi,t_0)dt\right)^2}{||h(t,x^*|\omega_0,V,\phi,t_0)||^2} \tag{8.4.11}$$

$$\hat{F} = \frac{\left|\int_0^\infty z(t)h(t,x^*|\mathcal{I}_i\cup\hat{\mathcal{I}})\,dt\right|}{||h(t,x^*|\mathcal{I}_i\cup\hat{\mathcal{I}})||^2} \tag{8.4.12}$$

(f) For multiple observations x_i^*, $i = 1, ..., I$:

$$\hat{\mathcal{I}} = \underset{\mathcal{I}}{\operatorname{argmax}} \frac{\left(\sum_{i=1}^{I} \int_0^\infty z(t) h(t, x_i^* | \omega_0, V, \phi, t_0) dt\right)^2}{\sum_{i=1}^{I} ||h(t, x_i^* | \omega_0, V, \phi, t_0)||^2} \tag{8.4.13}$$

$$\hat{F} = \frac{\left|\sum_{i=1}^{I} \int_0^\infty z(t) h(t, x_i^* | \mathcal{I}_i \cup \hat{\mathcal{I}})\, dt\right|}{\sum_{i=1}^{I} ||h(t, x_i^* | \mathcal{I}_i \cup \hat{\mathcal{I}})||^2} \tag{8.4.14}$$

The first insight that Theorem 8.4.1 gives is that in the full information case, the optimal estimator guarantees that the mean squared error decreases as $O(1/I)$, where I is the number of sensors. So in theory increased precision in the force estimation can be obtained by adding additional sensors to the system. In practice the limits are the uncertainties about the speed over a longer stretch of pavement might limit this performance.

The second observation is that as the information set becomes smaller, the complexity of the optimization needed to be carried out increases. For example, for the information set $\mathcal{I}_4 = \{V\}$, an optimization over the three remaining parameters ω_0, t_0 and ϕ needs to be carried out. The optimization itself is not convex, but the domain is bounded in ω_0 and ϕ. This fact can be used to devise a more efficient optimization methodology.

To conclude the section, we note that the result for the **Bounded Noise Model** is identical to the **White Noise Model**.

8.5 System design

In this section we address some important considerations when building a practical system for axle dynamic force computation. The first important consideration is the spatial placement of the acceleration sensors, which can result in improved estimation of the force. The next consideration is how to implement a computation system for the axle force, based on the methodology suggested in section 8.4. Some considerations about the most efficient approaches to compute the optimization should be made. Both issues are addressed in this section.

8.5.1 Sensor placement and design

Natural constraints on the placement of the sensor arise from observing the system response function to the moving load (Theorem 8.3.1(c)). The constraints are driven by observability requirements of the output of the system. The first constraint arises from the observation that taking samples of sensors at different locations x_i, at times $t_i = x_i/V + \delta$, for some constant δ, we obtain the response function

$$y(t_i, x_i) = F_0 Re[\psi^*(\delta)] \cos(\omega_0/V\, x_i + \omega_0\delta) + F_0 Im[\psi^*(\delta)] \sin(\omega_0/V\, x_i + \omega_0\delta) + O\big(e^{-k(\frac{x_i}{V}+\delta)}\big).$$

The Nyquist condition [Oppenheim et al., 1999a] implies that the sampling rate has to be less than twice the highest frequency of the signal, for uniformly sampled spatial signals. If we assume that sensors are placed uniformly according to $x_i = i\,\Delta x$, and the bandwidth of

$y(t_i, x_i)$ is $\Delta\omega_0$, the condition becomes

$$\frac{2\,\pi}{\Delta x} \geq 2\Delta\omega_0.$$

The truck suspension system parameter ω_0 is in the range $\omega_0 \in 2\pi[1,3]$, therefore $\Delta\omega_0 = 2\pi(3-1)/V$ for the signal of interest, and we can conclude the following requirement on the sensor placement:

$$\Delta x \leq \frac{V}{4} \text{ (meters)}.$$

Interestingly, the minimum speed of the truck in the system is the limitation on how closes sensors must be. If we assume that the minimum speed is 30 mph, the sensors must be at most 3.35 meters apart for observability of the measurement.

Similarly, the fundamental frequencies in the function ψ^* play a role as well. For each root λ_i of the system function $\Omega(s)$ in (Theorem 8.3.1(c)), the Nyquist criterion applies, therefore

$$\Delta x \leq \frac{2\pi}{2\max_i \lambda_i} \text{ (meters)}.$$

For high speeds V of the truck, Equation 8.3.8 shows that both conditions can be simplified to the condition

$$\Delta x \leq \frac{2\,\pi\,V}{2(\Delta\omega_0 + \sqrt{\beta/\gamma - k^2})} \text{ (meters)}.$$

8.5.2 Distributed data computation

The optimization in Equation (8.4.11) is complex when the information set is small. The optimization is not convex, but is in a bounded domain, which facilitates a simple approach. We consider here the smallest information set, $\mathcal{I} = \{V\}$. The procedure can be adjusted for other information sets in a straightforward manner.

The parameter t_0 is a time shift parameter, and can be optimized separately. One choice is to compute a cross correlation [Hayes, 1996; Ljung, 1999], which consists in calculating the objective function for a series of values of t_0 in some window of interest $[T_1, T_2]$ where the energy of the signal $z(t)^*$ is greater than the noise floor. Another choice is to compute Fourier transforms of both $z(t)^*$ and the normalized signal

$$\tilde{h}(t, x_i^*) = \frac{h(t, x_i^*|\omega_0, V, \phi, 0)}{\sqrt{\sum_{i=1}^{I} ||h(t, x_i^*|\omega_0, V, \phi, 0)||^2}} \tag{8.5.1}$$

and use Parseval's relation to obtain:

$$\hat{\mathcal{I}} = \underset{\mathcal{I}-\{t_0\}}{\operatorname{argmax}} \left(\sum_{i=1}^{I} \int_{-\infty}^{\infty} Z(\omega) \tilde{H}(\omega, x_i^* | \omega_0, V, \phi, 0)^* d\omega \right)^2$$

$$\hat{F} = \left| \sum_{i=1}^{I} \int_{-\infty}^{\infty} Z(\omega) \tilde{H}(\omega, x_i^* | \omega_0, V, \phi, 0)^* d\omega \right| \Bigg/ \sqrt{\sum_{i=1}^{I} \int_{-\infty}^{\infty} |H(\omega, x_i^* | \omega_0, V, \phi, 0)|^2 \, d\omega} \, .$$

If smart sensors are used, each scalar product and normalization constant can be computed separately and transmitted to the fusion center. The fusion center then implements the equation above.

8.5.3 Applications

The measurement of a pavement displacement when excited by an external impulse has several interesting applications:

Weigh-in-Motion. The acceleration measurement can be converted into an estimate of the (dynamic) weight of the truck, using techniques similar to the one presented in this chapter.

Axle Counting. The number of axles in a truck can be detected from the acceleration measurements converted to displacement. This is an important application and currently there are a very limited number sensors for this purpose.

Pavement Damage Meter. The measurement can be converted to an estimate of how much the pavement is damaged as well. Some existing methods associate the average observed weights to damage [Cebon, 1999], but direct response analysis could potentially be used to evaluate damage.

FWD Replacement. Falling weight deflectometers are currently used to test pavement response. These equipment drop a known weight on the pavement and measure the response. The costs of transporting the equipment to the test location and calibration can be quite high. Instead, permanently embedded sensors in the pavement could record pavement response to trucks that regularly use those roads. The pavement response can be inferred from this measurement.

Structure Monitoring. Embedding accelerometers in concrete beams such as the support or roadway of a bridge, allow for permanent monitoring of structural integrity. In such cases, usually the response to impulsive loads is a fundamental quantity.

Impact Monitoring. Airport runways require permanent monitoring for early intervention in the case of runway damage. Furthermore, impacts of airplanes at landing are measured to determine optimal parameters for landing procedures and runway construction.

8.6 Experimental Results

In this section we examine the behavior of the pavement-truck system and the quality of the proposed weight estimation schemes. We consider two types of concrete pavements: rough pavements and smooth pavements.

The rough pavement model is the general model considered in this chapter. Pavement roughness excites the truck suspension system resulting in a modulation of the pavement displacement measurements [Cebon, 1999]. Not accounting for this excitation while solving for the weight of the truck can result in large errors.

Smooth pavements, as deployed in various highways, do not excite the truck suspension system, resulting in a system with the behavior equivalent to setting $\omega_0 = 0$ in the proposed model. Our experimental setup has been built in a smooth pavement so qualitative results from the simulation and experimental measurements can be made under this assumption.

First we validate the approximation methodology proposed in Section 8.3. Next, we analyze the response for the rough pavement model, and compare the behavior for a range of truck speeds and suspension system frequencies. In the following section we investigate smooth pavements and the variation of the response with respect to various system parameters. To conclude the section we contrast the experimental data obtained from a pavement embedded accelerometer and evaluate the estimation quality of truck weight estimators based on the observed behavior and show that the proposed method is quite robust to noise in the acceleration measurements.

8.6.1 Pavement response analysis

We use the more general model to evaluate the quality of the approximation proposed in Section 8.3. We undertake a more flexible concrete model, using the following parameter values [Cebon, 1999][1]:

$$EI = 1.38 \times 10^6 Nm^2; \beta = 170 \times 10^6 N/m^2; \gamma = 353 \times 10^3 kg/m; \kappa = 10^6 Ns/m^2, \quad (8.6.1)$$

where we have assumed $M = 10^6; g = 10m/s^2$. We take the axle weight of the truck to be 5000 Kg, therefore $F_0 = 50000N$. Whenever unspecified, we take the suspension system fundamental frequency to be $\omega_0 = 1.23Hz$.

The main difficulty in simulating the distributed system (Equation (8.2.1)) is its stiffness with respect to parameters in Equation (8.6.1) [Chatti and Yun, 1996]. Stiffness means that small variations on the forcing function $F(x, t)$ cause large variations in the output $y(x, t)$. This is the case for any pavement model, given that the material structure itself is not very flexible and therefore the system will exhibit a stiff response [Hardy and Cebon, 1994].

The pavement response to a moving load can be computed exactly using Theorem 8.3.1 (Equation (8.3.3)). The solution is a convergent infinite summation. The summation cannot be computed exactly, but can be approximated by truncating at a predetermined number of terms. Lemma 8.7.1 shows that the truncation has exponential decay so the ignored part will only contribute a finite amount to the error. Unfortunately, such solution does not give much insight on the behavior of the system. Furthermore, for the parameters

[1] In the book these parameters are: $EI = 1.38MNm^2; \beta = 170MN/m^2; \gamma = 353Mg/m; \kappa = 1MNs/m^2$.

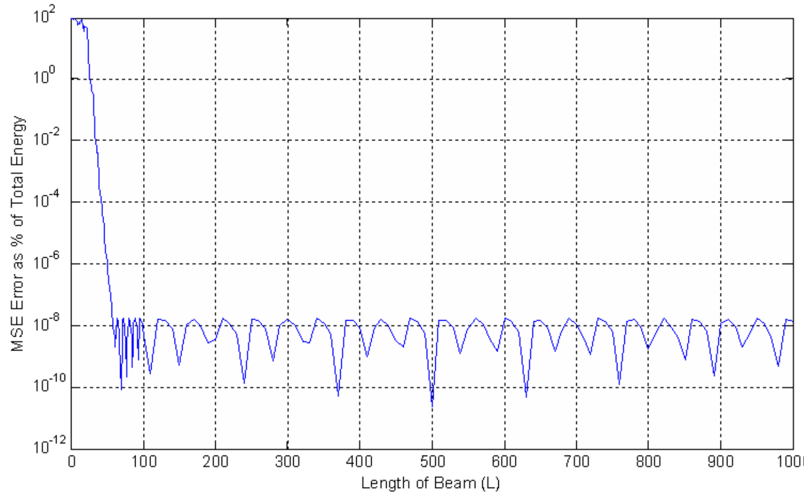


Figure 8.3. Relative Mean Squared Error (%) between ground truth displacement and asymptotic approximation at $L/2$ for $V = 10$ m/s ($y(L/2, t)$).

in Equation (8.6.1), the pavement exhibits a very stiff behavior, and the number of terms required is quite large. For the reported parameters, we observed that at least 5,000 elements were required before the norm of the additional terms being added is a small fraction of the sum at that point.

An alternative approach is to compute a direct numerical solution to the original PDE (Equation (8.2.1)). A Finite Element Method is indicated for this problem. The publicly available state-of-the-art FlexPDE solver can be used. Due to the high degree of stiffness of the PDE, the solver has difficulties finding acceptable numerical approximations to the response since it has to handle very poorly conditioned matrix inversions. In our experiments, the numerical approximation resulted in solutions qualitatively correct but with severe kinks, which are not physically valid.

The closed form approximation in Theorem 8.3.1(b) is easy to compute. The solution is exact as the length $L \to \infty$. It is important then to evaluate the quality of the approximation for finite values of L. As the gold standard we choose the truncated solution based on Theorem 8.3.1(b), with a large number of coefficients, $N = 10,000$, where it is observed computationally that the summation has converged to a degree. The relative mean squared error is used for comparison purposes:

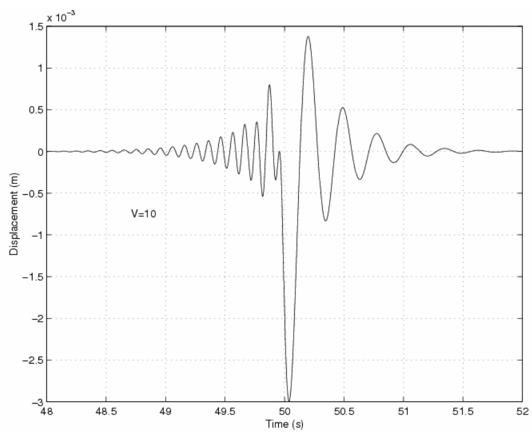
$$\mathrm{Err}(r(t), y(t)) = \frac{\int_0^\tau (r(t) - y(t))^2 dt}{\int_0^\tau r(t)^2 dt} \tag{8.6.2}$$

Figure 8.3 shows the relative mean squared difference between the ground truth solution and the asymptotic approximation, in percentages. A fixed position $x = L/2$ was chosen. Of course the solution is accurate away from the boundaries, and in our highway problem we are only interested in the behavior away from the virtual boundaries as well. Notice that very quickly the error becomes negligible. It is safe to say that for $L > 50$ m, we have an accurate solution for the given parameters choice.

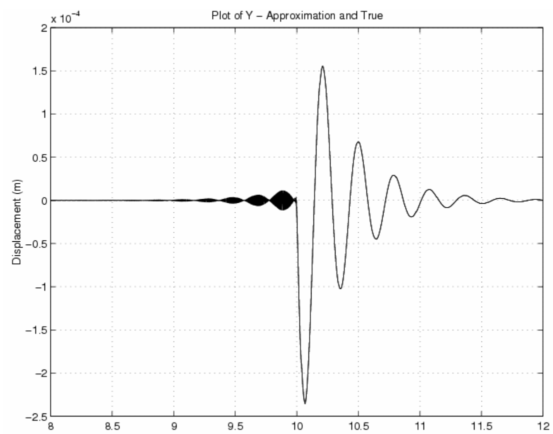
8.6.2 Rough pavement model

To start our analysis, we compute the responses of the system to a variety of changes in the parameters representing the truck such as its speed V and suspension frequency ω_0.

Figure 8.4 shows the displacement $y(x,t)$ at $x = 500$ m. The peak of the response happens at $t = 50$ s, when the truck is above the sensor, as expected. Furthermore, even before the truck arrives at $x = 500$ m, there is a response the signal being generated. This is a typical characteristic of a distributed parameter wave system. We also computed the signal frequencies before and after the arrival of the truck at $x = 500$ m. A single frequency before and a single frequency after are responsible for most of the response. As we saw in the theoretical section, the frequencies before the arrival of the truck at $x = 500$ m consist of the imaginary parts of the anti-causal poles of the response transfer function and the frequencies after correspond to the imaginary part of the causal poles.



(a)



(b)

Figure 8.4. (a) Displacement at $x = 500$ m, for $L = 1000$ m, $V = 10$ m/s and $\omega_0/2\pi = 1.23$ Hz. Fundamental frequencies (amplitudes) for the signal before $t = 50$ s are 4.7 Hz (14.2) and 9.8 Hz (2.8). After $t = 50$ s they are 3.7 Hz (13.8) and 3.5 Hz (3.1). (b) Displacement at $x = 500$ m, for $L = 1000$ m, $V = 50$ m/s and $\omega_0/2\pi = 1.23$ Hz.

One interesting feature is that the signal after the truck arrival vibrates at a smaller frequency than the signal before. In physical terms it can be understood as a doppler type phenomenon, but the waves being propagated are vibrations and the propagation medium is the pavement.

Figure 8.5 shows the wave behavior of the displacement response. The response is approximately localized in space and time, i.e., at any given fixed measurement point, there is a small time window of useful data. Furthermore the figure also shows more clearly the effects of modulating the typical response. In summary, the displacement response at any point in space is an appropriately shifted and modulated version of the response at any other point, with fix modulation frequency but variable phase.

In Section 8.3 we computed the asymptotic pole locations as the velocity of the truck become high. The fourth order polynomial reduced to a second order polynomial with causal complex roots. That is, the response of the pavement before the truck arrives at the

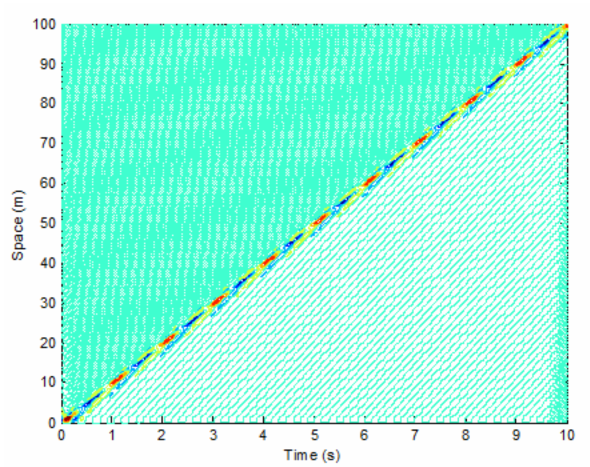
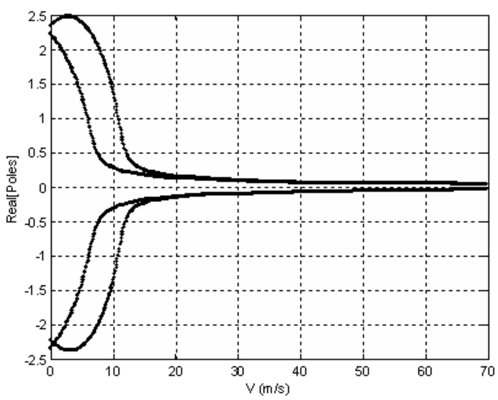
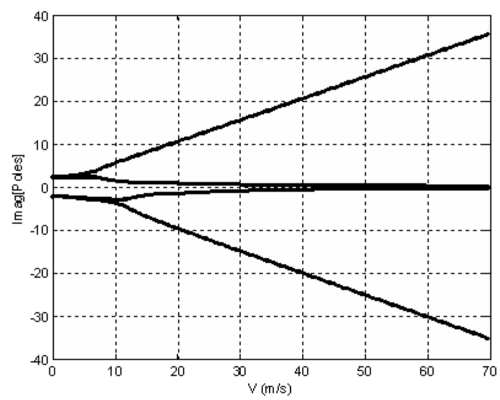


Figure 8.5. Contour plot of displacement $y(x,t)$, for $L = 1000$ m, $V = 10$ m/s and $\omega_0/2\pi = 1.23$ Hz.



(a)

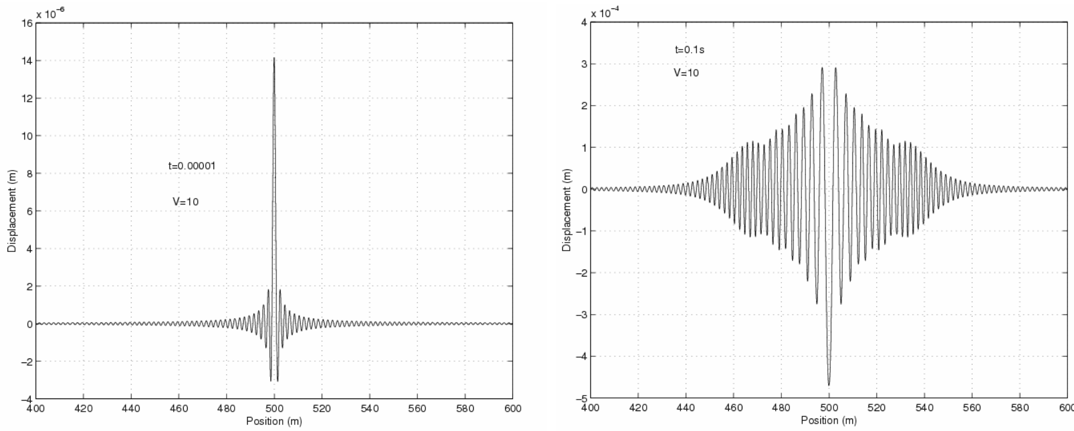


(b)

Figure 8.6. (a) Real and (b) imaginary parts of the poles of the system for $L = 1000$ m and $\omega_0/2\pi = 1.23$ Hz.

measurement location is negligible compared to the response after. Figure 8.4 shows the response with the truck at a higher speed, confirming this asymptotic viewpoint.

Figure 8.6 shows the variation of the magnitudes of the real and imaginary parts of the poles of the response with respect to the speed. As the speed becomes higher, we see that a pair of the imaginary frequencies tend to a small value. Furthermore, the remaining pair increases linearly with speed and becomes approximately conjugate. This behavior also means that the expansion coefficients for the small value imaginary frequency poles become small, as they are directly proportional to the product of the magnitude of the remaining poles. Thus, as speed increases, the 4 pole system collapses to a two pole system approximately. This observation will be useful to calibrate the model PDE. The Figure also shows the real part of the poles, and they confirm the notion that as the speed becomes higher we end up with a pair of causal complex conjugate poles and possibly a pair of anti-causal complex conjugate poles.



(a) (b)

Figure 8.7. Displacement impulse response along the highway, for $L = 1000$ m, at (a) $t = 0.00001$ s and (b) $t = 0.1$ s.

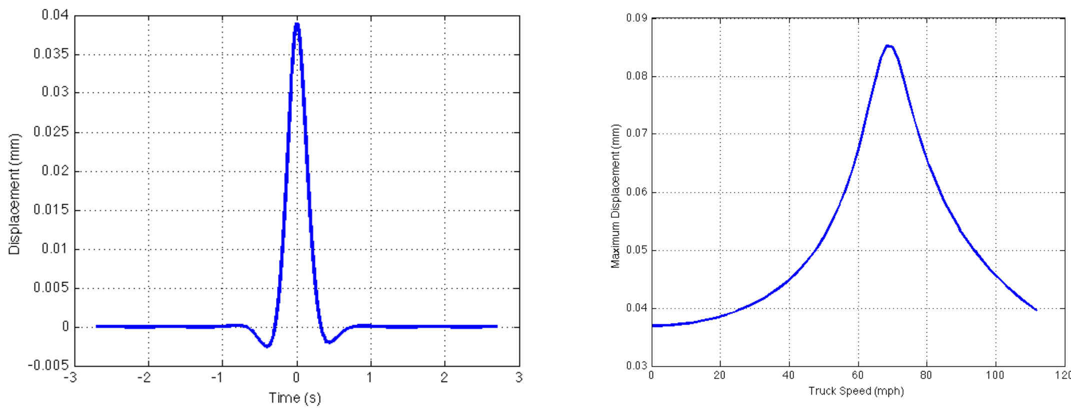
Finally, Figure 8.7 shows the impulse response for an impulse force (δ function) located at $x = 500$ m. Notice that now the oscillation is symmetric.

8.6.3 Smooth pavement model

The smooth pavement is modeled by taking $\omega_0 = 0$. Furthermore, in agreement with the observed values in real concrete pavement deployments, we consider a stiffer material, with $EI = 1.38 \times 120\,\mathrm{MNm}^2$. We set the length of the road to $L = 1000$m. Figure 8.8(a) shows the response at $x = 500$m for a truck moving at 10 m/s with $F = 50,000$N. Notice how the pavement stiffness reduces the ringing effect in the response.

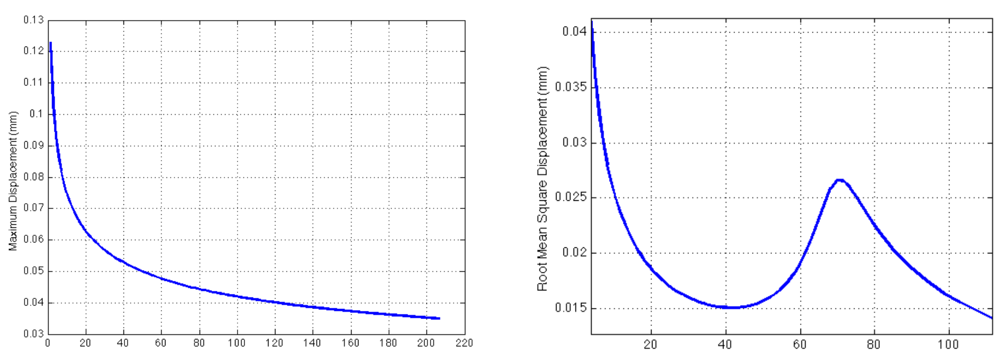
The response to the truck can be characterized by the value at the maximum. Figure 8.8(b) shows the variation of the peak value with truck speed. For speeds up to highway speeds, a fourth order relationship holds. Notice that the maximum deviation increases with speed, as the same energy is transferred to the road in a shorter amount of time. One method for converting a displacement measurement to truck force is to use a peak measurement of displacement and normalize it for speed effects using Figure 8.8(b), resulting in an estimate.

Figure 8.9(a) shows the peak variation with EI, the beam stiffness. Notice that beyond a certain level of stiffness, the qualitative behavior is the same. In fact, a careful inspection of simulation results reveals that for every EI value, there is a speed after which the qualitative behavior is the same as one for a lower EI and lower speed. Figure 8.9(b) shows the variation of the average energy of the displacement signal with truck speed. The advantage of this signal in an estimation procedure is the smaller variance of the uncertainty with respect to noise, when compared to the peak alone.



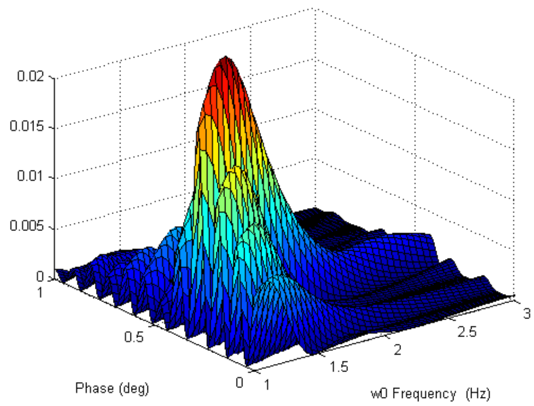
(a) (b)

Figure 8.8. (a) Displacement at $x = 500$ m, for $L = 1000$ m, $V = 10$ m/s. (b) Maximum displacement for varying truck speeds.

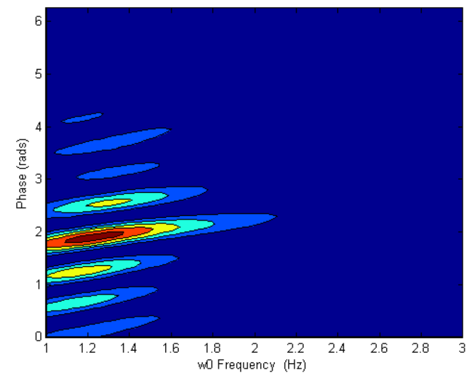


(a) (b)

Figure 8.9. (a) Maximum displacement for varying stiffness constant magnitudes. (b) Average energy (mean sum of squared values) of displacement in mm^2.

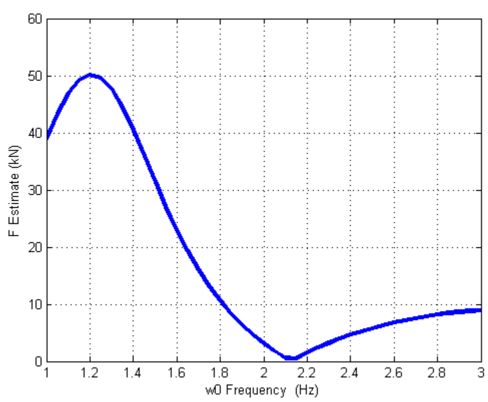


(a)

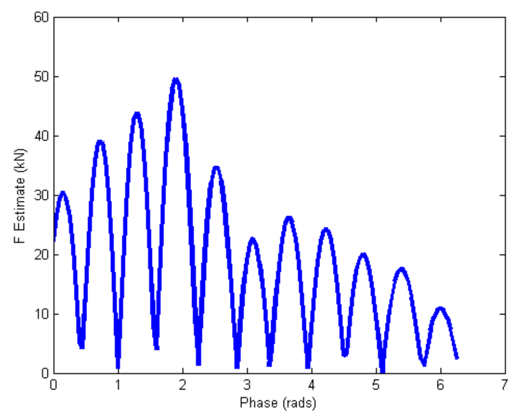


(b)

Figure 8.10. (a) Score function for Rough Pavement parameters. (b) Contour of score function.



(a)



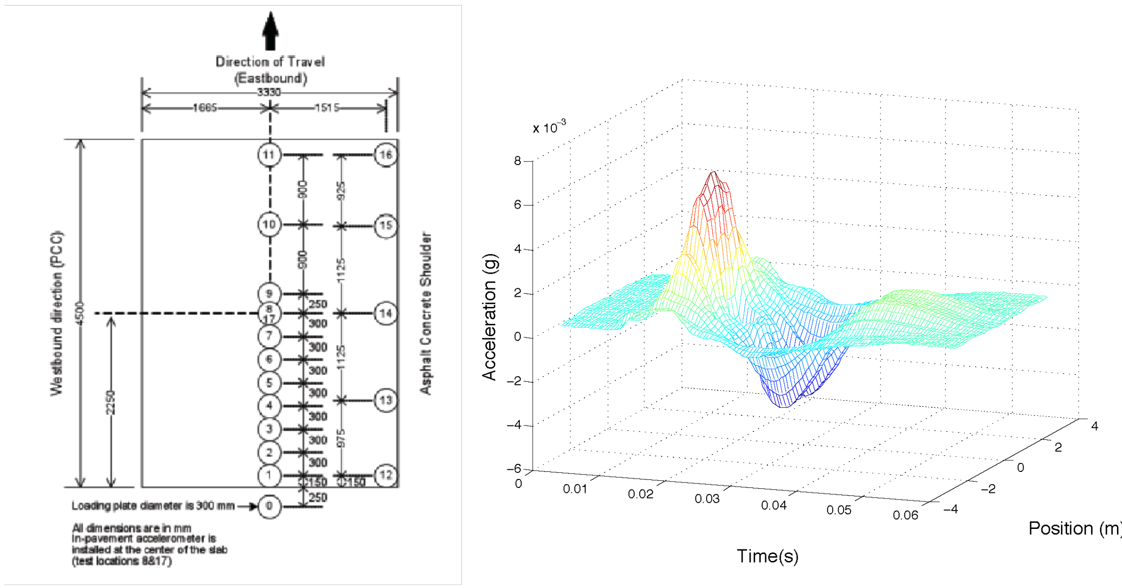
(b)

Figure 8.11. (a) Force estimate for a given ω_0 estimate assuming phase estimated correctly. True value is 50,000N. (b) Force estimate for a given ϕ estimate, assuming ω_0 estimated correctly.

8.6.4 Truck parameter estimation

There are three parameters of importance for the estimation of the force F to be accurate: knowing the shift t_0 (or random phase ϕ), the speed V and suspension system frequency ω_0. In the case of smooth pavements $\omega_0 = 0$, reducing the complexity of estimation. We assume that speed is measured by independent and accurate sensors.

Consider the experimental setup for the rough pavement model. Figure 8.10(a) and 8.10(b) show how the error correlation varies with chosen ω_0 and ϕ. Figure 8.11(a) and 8.11(b) shows a projection onto each of these axis. The sensitivity is not very high, and the estimation procedure can be quite robust.



(a) (b)

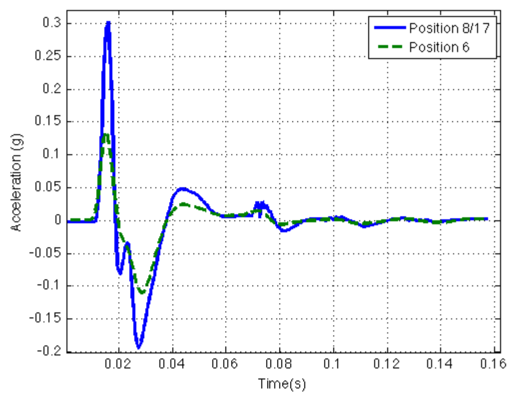
Figure 8.12. (a) Experimental setup for testing the pavement response with an embedded accelerometer (set at position 8/17). Weights are dropped at the numbered locations. (b) Measured acceleration map for the Falling Weight Deflectometer experiment (see text).

8.6.5 Field Data

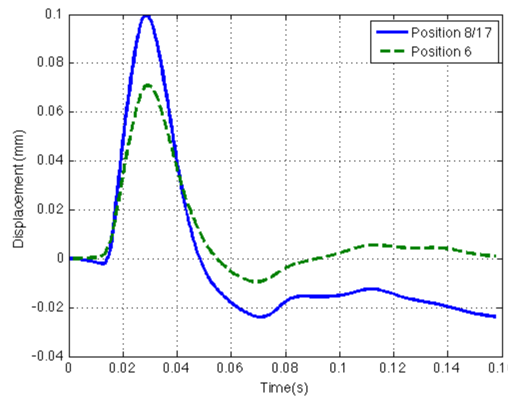
We performed a validation of the proposed approach, deploying an accelerometer sensor in a concrete pavement road (Road 32A in Yolo County, CA). Figure 8.12(a) shows the experimental deployment. The sensor is placed in the middle of a single lane. Two experiments were performed: one using a Falling Weight Deflectometer (FWD) to measure the pavement impulse response, and another using a real truck to measure the typical truck response. The noise power of the sensor after deployment was measured as $\sigma_w = 120\,\mu g$. The noise was not white due to a low pass filter used in the sensor.

In the first experiment, a known weight is dropped from a preset height at the numbered locations. The response of the pavement is recorded by the sensor. The equipment used for this experiment is a and the weight dropped simulates an impulse input to the concrete. Notice that since response is only recorded at a single location $x^* = 0$, we do not obtain the full impulse $h(x,t)$, which is the response observed by the whole concrete block to an impulse at $x = 0$. Instead we repeat the experiment at $x = x_k$, for $k = 1, ..., K$, (i.e. drop the weight at that location and measure at $x = 0$), to obtain $y(t|x_k)$. Assuming the pavement response is isotropic, we assign $h(x_k,t) = y(t|x_k)$. Due to the isotropic nature of the material, we also assume that $h(-x_k,t) = h(x_k,t)$. A smoothed version of the empirical measured acceleration map $h(x,t)$ is shown in Figure 8.12(b). The map shown interpolates the observed values.

The qualitative behavior of the displacement can be confirmed by plotting a numerical double integration of the observed $h(x_k,t)$. We chose locations 0 and 2, and show the corresponding output in Figure 8.13(b). Figure 8.13(a) shows the measured acceleration.

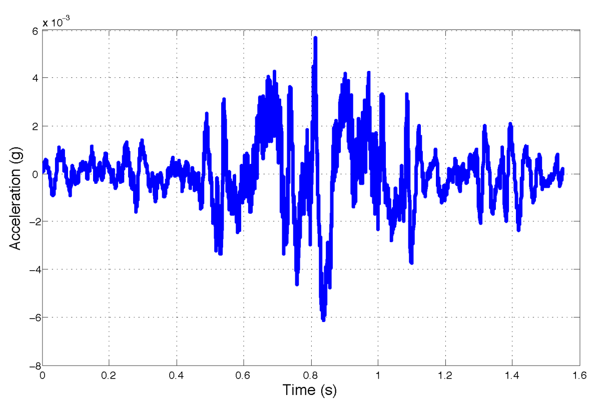


(a)

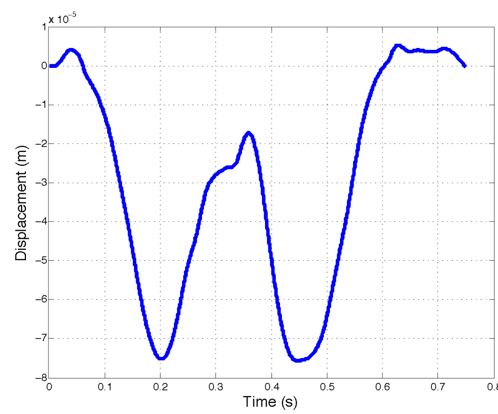


(b)

Figure 8.13. (a) Acceleration measurement for FWD experiment. Normalized to 50,000N. Drop positions are shown in the legend. (b) Displacement measurement for FWD experiment from double integrating acceleration.



(a)



(b)

Figure 8.14. (a) Acceleration measurement for truck experiment multiplied by -1 (to align orientation). 6,000N per axle truck moving at 35 MPH. (b) Displacement measurement for same experiment.

The behavior is close to that expected for a smooth pavement (e.g. Figure 8.7). Double integration of the correlated measurement noise creates the non-zero settling signal towards the end of the time window.

For the second experiment, we measure the response of the pavement to an excitation by a truck traveling at different speeds. Figure 8.14(a) shows a typical measured acceleration. Figure 8.14(b) shows an estimated displacement using the measurement. The estimation is not a straightforward double integration due to measurement noise. The response is very similar to those observed for smooth pavements. Multiple components are added together since the truck has multiple axles.

8.7 Proofs

8.7.1 Theorem 8.3.1

(a) A Fourier expansion should be used to solve the equation [Rao, 2007]. The basis choice is constrained by the boundary condition [Fryba, 1972; Rao, 2007]. There are four available basis $\sin\frac{\pi m x}{L}$, $\cos\frac{\pi m x}{L}$, $\sinh\frac{\pi m x}{L}$ and $\cosh\frac{\pi m x}{L}$. The four boundary conditions are used to determine the right expansion to use. Using a Fourier type expansion on the basis $\{\sin\frac{\pi m x}{L}\}$, integrating from 0 to L by parts and taking into account the boundary conditions (8.3.1) at $x = 0$ and $x = L$, we obtain the relations

$$\int_0^L y''''(x,t)\sin(\pi m x/L)dx = -(\pi m/L)^2\int_0^L y''(x,t)\sin(\pi m x/L)dx \tag{8.7.1}$$

$$= (\pi m/L)^4\int_0^L y(x,t)\sin(\pi m x/L)dx.$$

We now proceed as follows. We multiply both sides of equation (8.2.1) by $\sin(\pi m x/L)$ and integrate them from 0 to L in x. Further on, denoting $EI = \alpha$ and dividing both sides by γ, we come to equation

$$\ddot{Y_m} + 2\kappa\gamma^{-1}\dot{Y_m} + (\alpha(\pi m/L)^4 + \beta)\gamma^{-1}Y_m = F\gamma^{-1}\cos\omega_0 t\sin(\pi m V t/L) \tag{8.7.2}$$

with initial condition $Y_m(0) = \dot{Y}_m(0) = 0$. Here

$$Y_m(t) = \int_0^L y(x,t)\sin(\pi m x/L)dx$$

is the *finite Fourier sine coefficient* of function $y(x,t)$. The right-hand side arrived through formula

$$\int_0^L F\cos\omega_0(t)\sin(\pi m x/L)\delta(x - Vt)dx = F\cos\omega_0 t\sin(\pi m V t/L).$$

Using the definitions for k, F_0, ω_m^2 and Ω_m^2 we come to equation

$$\ddot{Y_m} + 2k\dot{Y_m} + \omega_m^2 Y_m = F_0\cos\omega_0 t\sin(\pi m V t/L) \tag{8.7.3}$$

with zero initial conditions for each m. We can now solve this equation by using the following simplification,

$$\ddot{Y_m} + 2k\dot{Y_m} + \omega_m^2 Y_m = \frac{F_0}{2}\sin((\pi mV/L + \omega_0)t) + \frac{F_0}{2}\sin((\pi mV/L - \omega_0)t) \tag{8.7.4}$$

Noticing that the roots of the differential equation are $k \pm \Omega_m i$ since $\beta > \kappa$ (implying $\omega_m^2 > k$), we can write the full response to the above ODE as shown in Equation (8.3.4) [Oppenheim et al., 1997].

(b) First we show that the transient part of the complete response to a moving excitation is $O(1)$ for the total transient response

$$Y_{tr}(x,t) = \frac{2}{L}\sum_{m=0}^{\infty} Y_{tr,m}(t)\sin\left(\frac{\pi m x}{L}\right). \tag{8.7.5}$$

Lemma 8.7.1. *$Y_{tr}(x,t) = O\big(e^{-k\,t}\big)$, uniformly in x. Furthermore, $\lim_{L\to\infty} Y_{tr}(x,t) = O\big(e^{-k\,t}\big)$, uniformly in x.*

Proof. From the definitions:

$$\begin{aligned}
|Y_{tr}(x,t)| &= \left|\frac{2}{L}\sum_{m=0}^{\infty} Y_{tr,m}(t)\sin\left(\frac{\pi m x}{L}\right)\right| \\
&\le \frac{2}{L}\sum_{m=1}^{\infty} |Y_{tr,m}| \\
&\le F_0 e^{-k\,t}\frac{2}{L}\sum_{m=1}^{\infty}\frac{1}{\Omega_m^3} \\
&\le F_0 e^{-k\,t}\frac{2}{L}\sum_{m=1}^{\infty}\frac{\gamma^{\frac{3}{2}}}{\alpha^{\frac{3}{2}}\,\pi^6 (m/L)^6} \\
&= \frac{2\,F\,\gamma^{\frac{1}{2}}}{\alpha^{\frac{3}{2}}\,\pi^6} e^{-kt}\,K(L),
\end{aligned} \tag{8.7.6}$$

where

$$K(L) = \frac{1}{L}\sum_{m=1}^{\infty}\frac{1}{(m/L)^6}.$$

For each finite L it is clear that $K(L) < \infty$. Moreover:

$$\begin{aligned}
\lim_{L\to\infty} K(L) &< \int_1^{\infty}\frac{1}{s^6}ds \\
&= 1/7.
\end{aligned}$$

□

We can now consider $Y_{ss,m}(t)$. Isolating the modulation of the forcing term:

$$\begin{aligned}Y_{ss,m}(t) = \quad & \frac{F_0}{2}\left\{|F_{a,m}|\sin\left(\frac{\pi m V}{L}t + \angle|F_{a,m}|\right) + |F_{b,m}|\sin\left(\frac{\pi m V}{L}t + \angle|F_{b,m}|\right)\right\}\cos(\omega_0 t) + \\ + \quad & \frac{F_0}{2}\left\{|F_{a,m}|\cos\left(\frac{\pi m V}{L}t + \angle|F_{a,m}|\right) - |F_{b,m}|\cos\left(\frac{\pi m V}{L}t + \angle|F_{b,m}|\right)\right\}\sin(\omega_0 t)\end{aligned} \tag{8.7.7}$$

We can now incorporate the sine term in Equation (8.3.3), using sine and cosine identities and moving constants around

$$f(r) = \frac{1}{2L}\sum_{m=0}^{\infty}|F_{a,m}|\cos\left(\frac{\pi m}{L}r + \angle|F_{a,m}|\right) + |F_{b,m}|\cos\left(\frac{\pi m}{L}r + \angle|F_{b,m}|\right) \tag{8.7.8}$$

$$g(r) = \frac{1}{2L}\sum_{m=0}^{\infty}|F_{a,m}|\sin\left(\frac{\pi m}{L}r + \angle|F_{a,m}|\right) - |F_{b,m}|\sin\left(\frac{\pi m}{L}r + \angle|F_{b,m}|\right) \tag{8.7.9}$$

$$y(x,t) = F_0\{(f(Vt-x) - f(Vt+x))\cos(\omega_0 t) + (g(Vt+x) - g(Vt-x))\sin(\omega_0 t)\} \tag{8.7.10}$$

Now we can compute the following quantity

$$f^*(r) = \lim_{L\to\infty}\frac{1}{2L}\sum_{m=0}^{\infty}|F_{a,m}|\cos\left(\frac{\pi m}{L}r + \angle|F_{a,m}|\right) + |F_{b,m}|\cos\left(\frac{\pi m}{L}r + \angle|F_{b,m}|\right). \tag{8.7.11}$$

Defining the constants

$$\begin{aligned}\omega_{a,\xi} &= \pi\xi V + \omega_0, \quad \omega_{b,\xi} = \pi\xi V - \omega_0,\\ \omega_\xi^2 &= (\alpha(\pi\xi)^4 + \beta)/\gamma, \quad \hat{\omega}_{a,\xi} = \xi V + \omega_0,\\ F(s,\xi) &= s^2 + 2ks + \omega_\xi^2,\\ F_{a,\xi} &= F(i\omega_{a,\xi},\xi)^{-1},\\ F_{b,\xi} &= F(i\omega_{b,\xi},\xi)^{-1} = F^*(i\omega_{a,\xi},-\xi)^{-1},\end{aligned}$$

we can obtain

$$\begin{aligned}f^*(r) &= \frac{1}{2}\int_0^\infty \{|F_{a,\xi}|\cos(\pi\xi r + \angle|F_{a,\xi}|) + |F_{b,\xi}|\cos(\pi\xi r + \angle|F_{b,\xi}|)\}\,d\xi\\ &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left|F(i\hat{\omega}_{a,\xi},\xi)^{-1}\right|\cos\left(\xi r + \angle F(i\hat{\omega}_{a,\xi},\xi)^{-1}\right)d\xi\end{aligned} \tag{8.7.12}$$

Now define:

$$\psi^*(r) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\Omega_\xi^{-1}e^{i\xi r}d\xi,$$
$$\Omega_\xi = -(\xi V+\omega_0)^2+2k(\xi V+\omega_0)i+(\alpha\xi^4+\beta)/\gamma. \tag{8.7.13}$$

Using this definition we can see that

$$f^*(r) = Re[\psi^*(r)], \tag{8.7.14}$$
$$g^*(r) = \lim_{L\to\infty} g(r) = Im[\psi^*(r)]. \tag{8.7.15}$$

Notice that Equation (8.7.13) is just the definition of an inverse fourier transform of Ω_ξ^{-1}. We can study the zeros of this transfer function. For this purpose we can write the poles/zeros of the Fourier transform as

$$\begin{aligned}\Omega_\xi &= -(\xi V+\omega_0)^2+2k(\xi V+\omega_0)i+(\alpha\xi^4+\beta)/\gamma \qquad (8.7.16)\\ &= \alpha/\gamma\xi^4-V^2\xi^2+(-2\omega_0V+2kVi)\xi+(\beta/\gamma-\omega_0^2+2k\omega_0 i)\\ &= \alpha/\gamma(i\xi)^4+V^2(i\xi)^2+(2\omega_0Vi+2kV)(i\xi)+(\beta/\gamma-\omega_0^2+2k\omega_0 i)\\ &= \alpha/\gamma s^4+V^2s^2+(2\omega_0Vi+2kV)s+(\beta/\gamma-\omega_0^2+2k\omega_0 i).\end{aligned}$$

(c) For the fixed excitation case, we can start at equation (8.7.2) and compute the solution to the fixed excitation $F(x,t) = F\delta(t-t_0)\delta(x-x_0)$ applied at the point x_0 at time t_0.

$$\ddot{Y}_m+2\kappa\gamma^{-1}\dot{Y}_m+(\alpha(\pi m/L)^4+\beta)\gamma^{-1}Y_m = F_0\delta(t-t_0)\sin(\pi m x_0/L). \tag{8.7.17}$$

Using the same definitions and initial conditions as in the moving excitation case, we can solve the above ODE:

$$y(x,t) = \frac{2}{L}\sum_{m=0}^{\infty}Y_m(t)\sin\left(\frac{\pi m x}{L}\right) \tag{8.7.18}$$
$$Y_m(t) = F_0\Omega_m^{-1}e^{-k(t-t_0)}\sin(\Omega_m(t-t_0))\sin(\pi m x_0/L)u(t-t_0) \tag{8.7.19}$$

We can develop the previous result, obtaining equation (8.3.6).

8.7.2 Theorem 8.3.2

Integrating by parts and taking into account the boundary conditions (8.3.9) at $x=0$ and those at $x=\infty$, we get the relations

$$\int_0^\infty y''''(x,t)\sin(\xi x)dx = -\xi^2\int_0^\infty y''(x,t)\sin(\xi x)dx = \xi^4\int_0^\infty y(x,t)\sin(\xi x)dx. \tag{8.7.20}$$

These will be used as follows. We multiply both sides of equation (8.2.1) by $(2/\pi)^{1/2}\sin\xi x$ and integrate them from 0 to ∞ in x. Further, denoting $EI=\alpha$ and dividing both sides by γ, we come to equation

$$\ddot{Y}_\xi + 2\kappa\gamma^{-1}\dot{Y}_\xi + (\alpha\xi^4 + \beta)\gamma^{-1}Y_\xi = F\gamma^{-1}\cos\omega_0 t \sin\xi V t, \tag{8.7.21}$$

with initial condition $Y_\xi(0) = \dot{Y}_\xi(0) = 0$. Here

$$Y_\xi(t) = (2/\pi)^{1/2}\int_0^\infty y(x,t)\sin(\xi x)dx \tag{8.7.22}$$

is the *Fourier sine transformation* of function $y(x,t)$. The right-hand is side arrived through formula

$$\int_0^\infty F\cos\omega_0(t)\sin(\xi x)\delta(x - Vt)dx = F\cos\omega_0 t\sin\xi V t. \tag{8.7.23}$$

Denoting $\kappa/\gamma = k,\ (\alpha\xi^4 + \beta)/\gamma = \omega^2,\ \Omega^2 = \omega^2 - k^2$ we come to equation

$$\ddot{Y}_\xi + 2k\dot{Y}_\xi + \omega^2 Y_\xi = F_0\cos\omega_0 t\sin\xi V t \tag{8.7.24}$$

with zero initial conditions. Assuming $\omega^2 - k^2 > 0$, the roots of its characteristic equation are

$$\lambda = -k_\xi \pm i\Omega_\xi. \tag{8.7.25}$$

Now we can follow the steps of Theorem 8.3.1 and using the definition of inverse Sine Transform in 8.3.11, we obtain Theorem 8.3.2.

8.7.3 Theorem 8.4.1

Denote $h(t, x^*|\omega_0, V, \phi, t_0)$ by $h(t, x^*)$. To prove (a), notice

$$z(t) = F\,h(t, x^*) + \xi(t).$$

Now consider the the projection operator that projects $z(t)$ into two subspaces: the subspace defined by $\frac{h(t,x^*)}{||h(t,x^*)||}$ and the subspace orthogonal to it. It is clear that the projection to the orthogonal subspace will not contain any information about F, as the noise is white. Thus our infinite dimensional estimation problem is reduced to:

$$\int_0^\infty \frac{z(t)h(t,x^*)}{||h(t,x^*)||}dt = F||h(t,x^*)|| + \int_0^\infty \frac{\xi(t)h(t,x^*)}{||h(t,x^*)||}dt.$$

In this one dimensional problem, with a single measurement, it is clear that the optimal estimate of F is obtained by dividing both sides by $||h(t, x^*)||$. The MSE can be computed directly from the definition, using $\mathbb{E}\big[\int_0^\infty \frac{\xi(t)h(t,x^*)}{||h(t,x^*)||^2}dt\big] = 0$ and $\mathbb{E}\big[\int_0^\infty \frac{\xi(t)h(t,x^*)}{||h(t,x^*)||^2}dt\big]^2 = \sigma^2/||h(t,x^*)||^2$ since $\xi(t)$ is white noise with variance σ^2.

For (b) the proof follows from (a), noticing that the multiple sensor problem can be reduced to the single problem by considering a composite vector $\mathbf{z} = [z_1 z_2 ... z_I]^T$.

The proof for (c) uses the Parseval's relation [Oppenheim et al., 1997] for a function

$f(t)$. Let $F(w)$ be the Fourier transform of $f(t)$. Then, using the time shift property of the Fourier Transform, $f(t-t_0) \Leftrightarrow e^{-j\,w\,t_0}F(w)$, and Parseval's relation, we have the identity

$$\int_0^\infty f(t-t_0)^2\,dt = \frac{1}{2\pi}\int_0^\infty |e^{-j\,w\,t_0}\,F(w)|^2\,dw = \frac{1}{2\pi}\int_0^\infty |F(w)|^2\,dw$$
$$= \int_0^\infty f(t)^2\,dt.$$

Using the identity, it is clear that for the given information set the proposed estimator computes F exactly when the measurement is noise free.

For item (e), we start by writing the empirical mean squared error function,

$$E = \int_0^\infty (z(t) - F\,h(t, x^*|\omega_0, V, \phi, t_0))^2\,dt.$$

One of the optimality conditions for the estimator is that $\partial E/\partial F = 0$, which implies

$$F = \frac{\sum_{i=1}^I \int_0^\infty z(t)h(t, x^*||\omega_0, V, \phi, t_0)\,dt}{\sum_{i=1}^I ||h(t, x^*||\omega_0, V, \phi, t_0)||^2}.$$

Now use this equation in the definition of the error E, and ignoring the term that only depends on z, we obtain

$$E' = -\frac{\left(\int_0^\infty z(t)h(t, x^*|\omega_0, V, \phi, t_0)dt\right)^2}{||h(t, x^*|\omega_0, V, \phi, t_0)||^2},$$

and so maximizing $-E'$ is equivalent to minimizing E. Item (f) follows by using the same approach to the cost

$$E = \sum_{i=1}^I \int_0^\infty (z(t) - F\,h(t, x_i^*|\omega_0, V, \phi, t_0))^2\,dt.$$

8.8 Discussion

In this chapter we have developed a methodology for estimating the magnitude of a dynamic forcing function applied to a concrete roadway, using distributed measurements from acceleration sensors embedded in the pavement. We use an asymptotic approximation to the pavement model that can be efficiently computed as the basis of our estimator. The asymptotic model is accurate for concrete slabs starting at 20 meters.

We verified the behavior of the pavement response using some simulations and then developed a time synchronized estimator for the forcing parameter. We also calculate an error bound that shows the quality of our approximation, which is helpful to gauge the quality of responses in field experiments.

One important issue that is addressed is also the need for a maximum distance between the sensors, which we derive using principles from sampling theory. This distance is the only constraint in sensor placement for our problem.

An extension of the current model, left for future work, is to model the roadway as a 2-dimensional system. We also are in the process of validating our setup experimentally with a multiple sensor array deployed at a concrete roadway.

Chapter 9

Contributions and suggested directions

In this Chapter we summarize the main contributions of the dissertation and present some suggested future directions for research.

9.1 Contributions

In this dissertation we introduced a principled methodology for building *adaptive signal and information systems* for monitoring a large transportation network. The methodology consisted of four main components: the deployment and design of sensors to measure the system, statistical methods for guaranteeing data reliability and managing the sensor network, and three important applications for the designed and deployed system.

In Chapter 2 we identified the important dynamic variables that characterize a transportation network. We reviewed various forms of sensing, and concluded that existing sensing infrastructure is appropriate to measure specific variables that characterize highways. We also identified two promising sensing technologies to measure properties of the road infrastructure, proposing a wireless road embedded accelerometer, and urban street system dynamics, using a magnetic wireless sensor network.

In Chapter 3 we investigated the reliability of the existing sensing infrastructure. The main purpose of the study was to create statistical metrics that are useful for characterizing the sensor network, as well as identifying the main challenges for maintaining a widely deployed sensor network that has the minimum reliability to meet the requirements for various applications. The main conclusion was that system reliability can be decomposed into three components: the quality of the *transduction technology*, the reliability of the *communication network* and the design of the *communication and sensing protocols*. A network's usefulness is directly related to these three metrics.

One conclusion of the empirical study of Chapter 3 was that there was a large fraction of sensors that exhibited unstable behavior. For certain periods of time these sensors report incorrect measurements, although the values alone seem plausible. In Chapter 4 we proposed a new method for detecting such periods in a group of sensors measuring a spatial and temporal phenomena. We created a framework for sequentially identifying failed sensors in a sensor network measuring a non-stationary environment. The method relies on comparing the data generated by a sensor with the data from its neighboring sensors, and a fault is declared if there is strong disagreement. The performance of the approach was analyzed using *change point theory* from statistical sequential analysis. The method was shown to be optimal under some weak conditions on the strength of information available for each sensor. An important theoretical contribution was the identification of a technique to analyze the performance of sequential networked decision making, when each decision maker faces an independent hypothesis test, but whose observation is correlated to the outcomes of hypothesis of his neighbors. The important conclusion is that in such cases, *collective strength of information* plays a major role, so even if a decision maker has weak information about his own hypothesis, his neighbors strength of information can help overcome this weakness.

In Chapter 5 we study how to optimally deploy and schedule energy-limited sensors in an existing network. We formulate the question as a statistical optimization problem, where a spatial stochastic model is learnt from data available for existing sensors. The problem is then shown to be combinatorially hard, and we propose a new algorithm to solve it approximately. We prove an appropriate performance guarantee using tools from algorithmic approximation theory. To the best of our knowledge this is the first *guaranteed* algorithm to this problem. Using the proposed algorithm we are able to design networks about 100 times bigger than with existing methods, and in various comparison examples we show that the performance of the designed network outperforms designs using these algorithms. We also show how the same approach can be used to create a privacy-preserving mechanism for sampling from mobile monitoring devices in a traffic application context.

The state of an urban traffic network is characterized by the statistical distribution of traffic variables in each network link, as opposed to a small number of moments from this distribution. Distributions in turn, can be characterized by appropriately chosen quantiles. It becomes important then to identify how to compute such quantiles in a network where data communication is expensive in terms of power consumption. Chapter 6 proposes various methods for computing such quantiles in a communication-efficient and sequential way, without requiring any prior parameterization of the statistical distribution. The method is based on sequential *stochastic approximation* theory, and is simple to implement in practice. One side benefit is that accurate performance estimates can be computed. We propose the method for both power constrained two way networks, such as for embedded magnetometer sensors, and one way power constrained network, such as for a mobile user, who uses existing quantile estimates for his own decision making.

In Chapter 7 we create a statistical method for estimating link travel times from magnetic signatures collected by a magnetometer sensor network deployed in a multiple lane urban street. The travel time for an individual vehicle is obtained by matching signatures measured by consecutive sensor arrays. Existing methods attempt to directly match signatures without imposing any structure arising from the application domain. The observed performance is insufficient to reliably determine the link travel time distributions for applications of interest. We instead propose a method that incorporates the combinatorial constraint that in a single lane, only a small fraction of vehicles overtake each other and return to the same lane. The method substantially outperforms existing methods, and is able to capture more than enough vehicles to provide reliable estimates of link travel time distributions. We compute some heuristic analysis to justify the performance of the method.

Chapter 8 concludes the dissertation's contributions by developing the application of an accelerometer sensor network for measuring heavy loads that impact road infrastructure. Currently existing methods for measuring such impact are either very expensive to deploy, or cannot be used in a permanent and real time manner. Instead, our proposed method relies on accurate measurements of the vibration experienced by the road when impacted by a load, and careful statistical models to estimate this load from the measurements. We model the system as a spatial distributed parameter system, and develop a careful approximation of the solution for the given formulation. We then apply optimal estimation theory to identify the best algorithm for estimating the load and the design of the physical deployment of the sensing network. We evaluate the performance of the methodology in real and experimental data set, concluding with the observation that multiple sensors are required for the application, but the technique is extremely promising.

9.2 Suggested directions

We have shown the *necessity* and *benefits* of using a statistically principled approach for monitoring and design of large engineering systems. In such systems, behavior is determined by the individual choices of a large number of autonomous agents, and therefore difficult to characterize in a purely deterministic way. Uncertainty needs to be introduced in the model to account for imperfect knowledge of the state of all agents and their decisions. The increase of computational power and recent advances in statistical methods provide . We have created various general methods as well as specific solutions illustrating the power of the approach, resulting in state of the art performance. There are five cornerstones that enable the approach: novel sensing and hierarchical processing structures, where decisions from lower layers can be locally computed and passed to higher layers; design of sensing and communication infrastructure aware of decision making and estimation method requirements; novel data representation methods for heterogeneous data sources; robust statistical methods that are able to work under imperfect data situations; and model learning and optimization methodologies that seek *approximate* optimality in place of exact optimality, therefore reducing computation complexity. In the remainder of the section we address specific application areas and opportunities.

9.2.1 Sensing and hierarchical processing

The capacity to sense a large network is central to any engineering systems approach to monitor and optimize societal scale systems. Creating novel forms of sensing requires benefiting from advances in transduction technologies and designing appropriate hierarchical computing and communication abstractions. For example, in systems where dynamics happen in different time scales, local processing can be beneficial and the reduction in data transmission rates essential for proper operation.

Some important directions for further investigation are possible new sensors for mobile sensing of traffic, such as radar sensors for parking detection and cellphone based monitoring for travel time estimation, and the development of a more extensive deployment of the accelerometer sensor network for load impact measurement. Other systems, such as bridges and buildings can benefit from the accelerometer sensor we proposed in this dissertation. We have started pursuing some of these goals in upcoming research for mobile traffic sensing and estimation.

9.2.2 Nonparametric sequential statistical methods

There are three statistical ideas central to monitoring and control of large scale systems: spatial stochastic modeling, sequential decision making methods and computationally efficient stochastic optimization. Chapters 4 to 7 introduced different variations of these methods to address monitoring and optimization in a transportation network, but the ideas were more generally applicable. There are two central difficulties in such problems: the decentralized nature of decision making in the system, requiring possible multiple layers of abstraction and information aggregation, and the stochastic nature of the information, due to imprecision in the sensing methods, reliability issues in the data collection system and impossibility of measuring certain properties of the system.

Determining appropriate abstraction and aggregation layers, such as how to partition a distributed estimation problem among a fusion center and the local sensing nodes, should be done accounting for power and processing constraints and can be specified using an objective function for system performance. Usually it is impossible to determine optimal solutions for the partition that maximizes the objective function, and therefore approximate solutions are available. In general, *sequential theory*, such as stochastic approximation and sequential analysis, provide a framework to determine and analyze appropriate methodologies for optimization. The main challenge though, is to ensure that such methodologies satisfy processing and sensing constraints in real applications.

The stochastic nature of information requires that any estimation or detection problem to monitor the system account for uncertainties in an appropriate manner. Traditionally, a parametric form is assumed for uncertainty (Gaussian distribution) and accounted for in the problem formulation, resulting in specific algorithms and strategies. Unfortunately in many systems, such as urban traffic, parametric assumptions do not lead to satisfactory models for the observed dynamics. The development of *nonparametric methods* for modeling large engineering systems is one solution to this problem. Algorithms and methodologies from nonparametric statistics can be developed to handle the modeling of dynamic systems with a large state space.

One important direction for future research that combines both sequential theory and nonparametric statistics is to expand the change point detection approach to event detection in a large sensor network. In traffic applications, event detection is useful for efficient incident management in highways. Concretely, we are currently developing a methodology for event detection based on using correlations between sensor data. If the correlation window has enough samples, then it can be approximately modeled as arriving from a normal distribution, despite the fact that the data itself is from any distribution. Our methodology generalizes various existing methods for event detection currently in the literature.

9.2.3 Representing, identifying and analyzing interconnected systems

A large network of interconnected systems characterizes the infrastructure systems we study. In such scenarios, it is important to develop approaches for representing the interconnected behavior, identifying the appropriate model from data and analyzing system response based on observed changes on the input.

We are currently pursuing two main areas of inquiry: how to model the spatial and time behavior of a distributed parameter system, such as concrete pavement or a building, using a sparse network model, and how to learn such models from imperfect data obtained from sensor measurements. The impulse response of a linear distributed parameter system can be modeled in a sense as a spatial stochastic process, and a better understanding of the connection between spatial statistics, nonparametric methods and dynamic systems will allow us to easily model more complex systems where a Partial Differential Equation representation is not readily available.

The analysis of interconnected system behavior focuses on understanding the behavior of the system when the patterns of interconnection change or information sharing between interconnected subsystems is subject to errors. Some important questions that need to be addressed is *re-identification* of a large interconnected system after a small change in patterns. The usual approach is to completely re-identify the system, thus requiring a large set of observations. It has been our observation from various practical problems, such as those presented in Chapter 4 and 6, that re-identification should require a smaller data sample. For systems that exchange information subject to errors, one fruitful area of inquiry is the extension of the methods in Chapter 6 to a network without a fusion center.

One last problem we mention as an important theoretical line of inquiry in network systems science is understanding how to compute properties of inference problems in random graphs. Our first step in this direction is computing asymptotic error rates for the matching problem discussed in Chapter 7.

9.2.4 Transportation systems: large scale monitoring and closing the loop

In Chapter 1 we presented a systematic view on how to think about building solutions for transportation systems. The methodology went from monitoring to intervention, where by intervention we imply the use of monitored data to optimize system behavior according to a chosen metric. There are various important directions to be explored based on the methodologies we suggest.

System performance can be optimized either by optimizing the behavior of system controls, such as traffic lights, or by shifting the behavior of drivers, by providing them with both incentives and mechanisms to do better decisions. The basis of methodologies for both are the monitored information obtained by deploying the sensing mechanisms we designed together with the integration of existing data sources. We are working in the first method by considering how detailed information obtained from a magnetic sensor network can be used to both perform optimization of traffic light sequencing and predictive routing for users.

Road infrastructure maintenance planning can benefit by developing methodologies based on the accelerometer sensor network proposed in this dissertation. The cost effectiveness of the sensor and the advanced modeling methodologies it enables due to the detailed measurements will enable various applications for optimizing the planning and construction of roads. Furthermore, the sensor can serve other purposes as well. In one application we are currently developing, the number of axels in a truck is counted using the deployed sensor.

We believe that the problems explored in this dissertation provide an avenue to a rich set of questions, both in the context of transportation and infrastructure networks, as well as for other applications.

Bibliography

Z. Abrams, A. Goel, and S. Plotkin. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *IPSN*, 2004.

S. Amari and T. S. Han. Statistical inference under multiterminal rate restrictions: A differential geometric approach. *IEEE Trans. Info. Theory*, 35(2):217–227, March 1989.

A. Asadpour and A. Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. In *STOC*, pages 114–121, New York, NY, USA, 2007. ACM Press. ISBN 978-1-59593-631-8. doi: http://doi.acm.org/10.1145/1250790.1250808.

E. Ayanoglu. On optimal quantization of noisy sources. *IEEE Trans. Info. Theory*, 36(6): 1450–1452, 1990.

X. Bai, S. Kumar, Z. Yun, D. Xuan, and T. H. Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *ACM MobiHoc*, Florence, Italy, 2006.

J. Baumeister, W. Scondo, M. A. Demetriou, and I. G. Rosen. On-line parameter estimation for infinite-dimensional dynamical systems. *SIAM Journal on Control and Optimization*, 35(2):678–713, 1997.

A. Benveniste, M. Metivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximations*. Springer-Verlag, New York, NY, 1990.

P. J. Bickel, C. Chen, J. Kwon, John Rice, E. van Zwet, and P. Varaiya. Measuring traffic. *Statistical Science*, 22(4):581–597, 2007.

R. S. Blum, S. A. Kassam, and H. V. Poor. Distributed detection with multiple sensors: Part ii—advanced topics. *Proceedings of the IEEE*, 85:64–79, January 1997.

A.A. Borovkov. Asymptotically optimal solutions in the change-point problem. *Theory of Probab. Appl.*, 43(4):539–561, 1999.

D. Cebon. *Handbook of Vehicle-Road Interaction*. Swets and Zeitlinger Publishers, 1999.

K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Stat. Sci.*, 10(3): 273–304, Aug. 1995. ISSN 08834237.

J. F. Chamberland and V. V. Veeravalli. Asymptotic results for decentralized detection in power constrained wireless sensor networks. *IEEE Journal on Selected Areas in Communication*, 22(6):1007–1015, August 2004.

T. H. T. Chan, S. S. Law, and T. H. Yung. Interpretive method for moving force identification. *Journal of Sound and Vibration*, 219(3):503–524, 1999.

K. Chatti and K. K. Yun. Sapsi-m: Computer program for analyzing asphalt concrete pavements under moving arbitrary loads. *Transportation Research Record*, 1539:88–95, 1996.

C. Chen, J. Kwon, J. Rice, A. Skabardonis, and P. Varaiya. Detecting errors and imputing missing data for single loop surveillance systems. *Transportation Research Record*, 1855: 160–167, 2002a.

C. Chen, J. Kwon, J. Rice, A. Sakabardonis, and P. Varaiya. Detecting errors and imputing missing data for single loop surveillance systems. *Transportation Research Record*, 1855: 160–167, 2003.

C. Chen, J. Kwon, and P. Varaiya. An empirical assessment of traffic operations. In H.S. Mahmassani, editor, *Proceedings of the 16th International Symposium on Transportation and Traffic Theory*, pages 105–124. Elsevier, 2005.

Y. Chen, C. A. Tanb, L. A. Bergmanc, and T. C. Tsaod. Smart suspension systems for bridge-friendly vehicles. In *Proceedings of the 2002 SPIE Annual International Symposium on Smart Structures and Materials; Smart Systems for Bridges, Structures, and Highways*, 2002b.

C. Chong and S. P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91:1247–1256, 2003.

B. Coifman. *Vehicle reidentification and travel time measurement using loop detector speed traps*. PhD thesis, University of California, Berkeley, Berkeley, CA 94720, 1999.

N. A. C. Cressie. *Statistics for Spatial Data*. Wiley, 1991.

C. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research, Part B*, 28(4):269–287, 1994.

A. Das and D. Kempe. Algorithms for subset selection in linear regression. In *STOC*, 2008.

A. Dempster, N. Laird, and D. Rubin. Likelihood from incomplete data via the EM algorithm. *J. Royal Statstical Society, B*, 39(1):1–38, 1977.

A. Deshpande, C. Guestrin, S. Madden, J. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *VLDB*, 2004.

A. Deshpande, S. Khuller, A. Malekian, and M. Toossi. Energy efficient monitoring in sensor networks. In *Latin*, 2008.

A. G. Dimakis, A. D. Sarwate, and M. J. Wainwright. Geographic gossip: efficient aggregation for sensor networks. In *Information Processing in Sensor Networks (IPSN)*, pages 69–76, 2006.

R. Durrett. *Probability: Theory and Examples*. Duxbury Press, New York, NY, 1995.

A. Duzdar and G. Kompa. Applications using a low-cost baseband pulsed microwave radar sensor. In *Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference*, volume 1, pages 239–243, Washington, DC, May 2001.

E. Elnahrawy and B. Nath. Context-aware sensors. *Lecture Notes in Computer Science (LNCS)*, 2920:77–93, 2004.

S. C. Ergen and P. Varaiya. Pedamacs: Power efficient and delay aware medium access protocol for sensor networks. *IEEE Transactions on Mobile Computing*, 5(7):920–930, 2006.

A. Ostfeld et al. The battle of the water sensor networks (BWSN): A design challenge for engineers and algorithms. *To appear in J. Wat. Res. Plan. Mgmt.*, 2008.

R. E. Ewing, T. Lin, and Y. Lin. A mixed least-squares method for an inverse problem of a nonlinear beam equation. *Inverse Problems*, pages 19–32, 1999.

M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions - ii. *Math. Prog. Study*, 8:73–87, 1978.

L. Fryba. *Vibration of Solids and Structures under Moving Loads*. Noordhoff International Publishing, Groningen, The Netherlands, 1972.

T. T. Fu and D. Cebon. Analysis of a truck suspension database. *Int. Journal of Heavy Vehicle Systems*, 19(4):281–297, 2002.

T. Fujito. Approximation algorithms for submodular set cover with applications. *TIEICE*, 2000.

M. Gerdin, T. B. Schon, T. Glad, F. Gustafsson, and L. Ljung. On parameter and state estimation for linear differential algebraic equations. *Automatica*, pages 416–425, 2007.

A. Gonzalez, A. T. Papagiannakis, and E. J. O'Brien. Evaluation of an artificial neural network technique applied to multiple-sensor weigh-in-motion systems. *Transportation Research Record*, 1855:151–159, 2003.

H. H. Gonzalez-Banos and J. Latombe. A randomized art-gallery algorithm for sensor placement. In *Proc. 17th ACM Symp. Comp. Geom.*, pages 232–240, 2001.

G.R. Grimmett and D.R. Stirzaker. *Probability and random processes*. Oxford Science Publications, Clarendon Press, Oxford, 1992.

J. A. Gubner. Decentralized estimation and quantization. *IEEE Trans. Info. Theory*, 39 (4):1456–1459, 1993.

J. Han, P. K. Varshney, and V. C. Vannicola. Some results on distributed nonparametric detection. In *Proc. 29th Conf. on Decision and Control*, pages 2698–2703, 1990.

T. S. Han and S. Amari. Statistical inference under multiterminal data compression. *IEEE Trans. Info. Theory*, 44(6):2300–2324, October 1998.

T. S. Han and K. Kobayashi. Exponential-type error probabilities for multiterminal hypothesis testing. *IEEE Trans. Info. Theory*, 35(1):2–14, January 1989.

A. Haoui, R. Kavaler, and P. Varaiya. Wireless magnetic sensors for traffic surveillance. *Transportation Research C*, 16(3):294–306, 2008a.

A. Haoui, R. Kavaler, and P. Varaiya. Wireless magnetic sensors for traffic surveillance. *Transportation Research Part C*, 16(3):294–306, 2008b.

M. S. A. Hardy and D. Cebon. Response of continuous pavements to moving dynamic loads. *Journal of Engineering Mechanics*, 119(9):1762–1780, 1993.

M. S. A. Hardy and D. Cebon. Importance of speed and frequency in flexible pavement response. *Journal of Engineering Mechanics*, 120(3):463–482, 1994.

M. H. Hayes. *Statistical Digital Signal Processing and Modeling*. John Wiley and Sons, 1996.

D. S. Hochbaum and W. Maas. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32:130–136, 1985.

S. R. Jefferey, G. Alonso, M. J. Franklin, W. Hong, and J. Widom. A pipelined framework for online cleaning of sensor data streams. In *ICDE*, 2006.

J. T. Kenney. Steady state vibrations of beam on elastic subgrade for moving loads. *Journal of Applied Mechanics*, 21(4), 1954.

Y. K. Ki and D.K. Baik. Model for accurate speed measurement using double-loop detectors. *IEEE Transactions on Vehicular Technology*, 55(4):1094–1101, 2006.

J. P. Klein and M. Moeschberger. *Survival Analysis Techniques for Censored and Truncated Data*. Springer-Verlag, 2nd edition, 2003.

L. A. Klein. *Data Requirements and Sensor Technologies for ITS*. Artech House, Norwood, MA, 2001.

F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli. On-line fault detection of sensor measurements. *Proc. IEEE Sensors*, pages 974–980, 2003.

F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli. Fault-tolerance in sensor networks. *Handbook of Sensor Networks*, 36, I. Mahgoub and M. Ilyas (eds.) 2004.

F. Koushanfar, N. Taft, and M. Potkonjak. Sleeping coordination for comprehensive sensing using isotonic regression and domatic partitions. In *Infocom*, 2006.

A. Krause and C. Guestrin. Nonmyopic active learning of gaussian processes – an exploration–exploitation approach. In *Proc. of 24th International Conference on Machine Learning (ICML)*, 2007.

A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. In *JMLR*, 2007.

A. Krause, E. Horvitz, A. Kansal, and F. Zhao. Towards community sensing. In *IPSN*, 2008a.

A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, and C. Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *J. Wat. Res. Plan. Mgmt.*, 136(6), 2008b.

A. Krause, B. McMahan, C. Guestrin, and A. Gupta. Selecting observations against adversarial objectives. In *NIPS*, 2008c.

H. J. Kushner and G. G. Yin. *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York, NY, 1997.

J. Kwon, P. Bickel, and J. Rice. Web of evidence models: Detecting sensor malfunctions in correlated sensor networks. Technical report, University of California Berkeley, 2003.

K. Kwong, R. Kavaler, R. Rajagopal, and P. Varaiya. Arterial travel time estimation based on vehicle re-identification using wireless sensors, 2008. Submitted to Transportation Research, Part C.

T. L. Lai. Sequential analysis: Some classical problems and new challenges (with discussion). *Statist. Sinica*, 11:303–408, 2001.

T. L. Lai. Information bounds and quick detection of parameter changes in stochastic systems. *IEEE Trans. on Info. Theory*, 44(7):2917–2929, 1998.

E. Lehmann. *Elements of Large-Sample Theory*. Springer, 1999.

S. K Leming and H. L. Stalford. Bridge weigh-in-motion system development using superposition of dynamic truck/ static bridge interaction. In *Proceedings of the American Control Conference*, 2003.

H. X. Liu and W. Ma. A virtual vehicle probe model for time-dependent arterial travel time estimation. Submitted for Publication, 2008.

L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, 2nd edition, 1999.

G. Lorden. Procedures for reacting to a change in distribution. *Ann. Math. Statist.*, 42: 1897–1908, 1971.

X. Luo, M. Dong, and Y. Huang. On distributed fault-tolerant detection in wireless sensor networks. *IEEE Transactions on Computers*, 55:58–70, 2006.

Z.Q. Luo. Universal decentralized estimation in a bandwidth-constrained sensor network. *IEEE Trans. Info. Theory*, 51(6):2210–2219, 2005.

C.A. MacCarley, S. Hockaday, D. Need, and S. Taff. Evaluation of video image processing systems for traffic detection. *Transportation Research Record*, 1360, 1992.

M. Markow, J. K. Hedric, B. D. Bradmeyer, and E. Abbo. Analyzing the interactions between vehicle loads and highway pavements. In *Proceedings of 67th Annual Meeting, Transportation Research Board*, 1988.

P. T. Martin, Y. Geng, and X. Wang. Detector technology evaluation. Technical Report 03-154, Department of Civil and Environmental Engineering, MPC, 2003.

K. Marzullo. Tolerating failures of continuous-valued sensors. *ACM Transactions on Computer Systems*, 8:284–304, 1990.

Y. Mei. Asymptotic optimality theory for decentralized sequential hypothesis testing in sensor networks. *IEEE Transactions in Information Theory*, 54(5):2072 – 2089, 2008.

P.G. Michalopoulos, R.D. Jacobson, C.A. Anderson, and J.C. Barbaresso. Integration of machine vision and adaptive control in the fast-trac ivhs program. In *72nd Annual Meeting of Transportation Research Board*, Washington, DC, January 1993.

L. E. Y. Mimbela and L. A. Klein. A summary of vehicle detection and surveillance technologies used in intelligent transportation systems. Technical report, Vehicle Detector Clearinghouse, New Mexico State University, 2000.

F. Moses. Weigh-in-motion system using instrumented bridges. *Transportation Engineering Journal*, 105(3):233–249, 1979.

E.W. Myers. An $O(ND)$ difference algorithm and its variations. *Algorithmica*, 1:251–266, 1986.

M. Ndoye, V. Totten, B. Carter, D.M. Bullock, and J.V. Krogmeier. Vehicle detector signature processing and vehicle reidentification for travel time estimation. In *Proceedings of 88th Transportation Research Board Annual Meeting*, Washington, D.C., January 2008.

X. Nguyen, M. J. Wainwright, and M. I. Jordan. Nonparametric decentralized detection using kernel methods. *IEEE Trans. Signal Processing*, 53(11):4053–4066, November 2005.

M. S. Nikulin. *Parametric and semiparametric models with applications to reliability, survival analysis, and quality of life*. Birkhuser, 2004.

C. Oh and S.G. Ritchie. Real-time inductive-signature-based level of service for signalized intersections. *Transportation Research Record*, 1802:97–104, 2002.

A. V . Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice-Hall, 2nd edition, 1999a.

A. V. Oppenheim, A. S. Willsky, and S. Hamid. *Signal and Systems*. Prentice-Hall, 2nd edition, 1997.

A.V. Oppenheim, R.W.Schafer, and J.R.Buck. *Discrete-time Signal Processing*. Prentice-Hall, Inc., New Jersey, 1999b.

E. Ould-Ahmed-Vall, G. F. Riley, and B. Heck. Distributed fault-tolerance for event detection using heterogeneous wireless sensor networks. Technical Report GIT-CERCS-06-09, Georgia Institute of Technology, 2007.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41:100–115, 1954.

S. Papadimitriou, J. Sun, and P. S. Yu. Local correlation tracking in time series. In *Sixth IEEE International Conference on Data Mining (ICDM)*, pages 456–465, 2006.

PeMS. California Performance Measurement System, http://pems.eecs.berkeley.edu, 2009.

M. Pollak. Average run lengths of an optimal method of detecting a change in distribution. *Annals of Statistics*, 15:749–779, 1987.

A.K. Ponnuswami and S. Khot. Approximation algorithms for the max-min allocation problem. In *APPROX*, 2007.

J.G. Proakis. *Digital Communications*. McGraw-Hill, New York,NY, 2000.

R. Rajagopal and P. Varaiya. Health of california's loop detector system. Technical Report UCB-ITS-PRR-2007-13, PATH, 2007.

S. S. Rao. *Vibration of Continuous Systems*. John Wiley and Sons, 2007.

S. W. Roberts. A comparison of some control chart procedures. *Technometrics*, 8:411–430, 1966.

L.A. Rossman. The epanet programmer's toolkit for analysis of water distribution systems. In *Annual Water Resources Planning and Management Conference*, 1999.

A. Safaai-Jazi, Siamak A. Ardekani, and Majid Mehdikhani. A low-cost fiber optic weigh-in-motion sensor. Technical Report SHRP-ID/UFR-90-002, Strategic Highway Research Program, National Research Council, Washington, DC, 1990.

D. Schrank and T. Lomax. The 2007 Annual Urban Mobility Report. Technical report, Texas Transportation Institute, 2007.

R. J. Serfling. *Approximation Theorems of Mathematical Statistics*. Wiley Series in Probability and Statistics. Wiley, 1980.

A. N. Shirayev. *Optimal Stopping Rules*. Springer-Verlag, 1978.

A. N. Shiryayev. On optimum methods in quickest detection problems. *Theory of Probability and Applications*, 8:22–46, 1963.

D. Siegmund. *Sequential Analysis: Tests and Confidence Intervals*. Springer-Verlag, 1985.

A. Skabardonis and N. Geroliminis. Real-time estimation of travel times along signalized arterials. In H.S. Mahmassani, editor, *Proceedings of the 16th International Symposium on Transportation and Traffic Theory*, pages 387–406. Elsevier, 2005.

S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *ICC*, 2001.

J. B. Sousa, J. Lysmer, S. S. Chen, and C. L. Monismith. Dynamic loads: effects on the performance of asphalt concrete pavements. In *Proceedings of 67th Annual Meeting, Transportation Research Board*, 1988.

L. K. Stergioulas, D. Cebon, and M. D. Macleod. Static weight estimation and system design for multiple-sensor weigh-in-motion. *Journal Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 214(8), 2000.

C. Sun, S.G. Ritchie, K. Tsai, and R. Jayakrishnan. Use of vehicle signature analysis and lexicographic optimization for vehicle reidentication on freeways. *Transportation Research, C*, 7:167–185, 1999.

C.C. Sun, G.S. Arr, R.P. Ramachandram, and S.G. Ritchie. Vehicle reidentification using multidetector fusion. *IEEE Transactions on Intelligent Transportation Systems*, 5(3): 155–164, September 2004.

L. Sun and T. W. Kennedy. Spectral analysis and parametric study of stochastic pavement loads. *Journal of Engineering Mechanics*, 128(3):318–327, 2002.

J. M. Sussman. *Introduction to Transportation Systems.* Artech House, 2000.

A.G. Tartakovsky and V.V. Veeravalli. General asymptotic bayesian theory of quickest change detection. *Theory of Probab. Appl.*, 49(3):458–497, 2005.

R. R. Tenney and N. R. Jr. Sandell. Detection with distributed sensors. *IEEE Trans. Aero. Electron. Sys.*, 17:501–510, 1981.

J. N. Tsitsiklis. Decentralized detection. In *Advances in Statistical Signal Processing*, pages 297–344. JAI Press, 1993.

D. Tulone and S. Madden. An energy-efficient querying framework in sensor networks for detecting node similarities. In *MSWiM*, pages 191–300, 2006.

A. W. van der Vaart. *Asymptotic Statistics.* Cambridge University Press, 1998.

V. V. Vazirani. *Approximation Algorithms.* Springer, 2001.

V. V. Veeravalli. Decentralized quickest change detection. *IEEE Transactions in Information Theory*, 47:1657–1665, 2001.

V. V. Veeravalli, T. Basar, and H. V. Poor. Decentralized sequential detection with a fusion center performing the sequential test. *IEEE Trans. Info. Theory*, 39(2):433–442, 1993.

R. Viswanathan and P. K. Varshney. Distributed detection with multiple sensors: Part i—fundamentals. *Proceedings of the IEEE*, 85:54–63, January 1997.

J. Vondrak. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, 2008.

N. A. Weber. Verification of radar vehicle detection equipment. Technical Report SD98-15-F, Southern Dakota Department of Transportation, Pierre, SD, 1999.

J.L. Williams, J.W. Fisher III, and A.S. Willsky. Performance guarantees for information theoretic active inference. In *AISTATS*, 2007.

Z. Zhang and T. Berger. Estimation via compressed information. *IEEE Trans. Info. Theory*, 34(2):198–211, 1988.

F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Processing*, 19(2):61–72, 2002.

Y. Zhao. Mobile phone location determination and its impact on intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 1(1):55–64, 2000.

Z. Zhou and J. Guo. On-line fault detection of sensor measurements. *Proc. SPIE*, 3374: 451–455, 1998.

R. Zielinski. Optimal quantile estimators: Small sample approach. Technical report, Inst. of Math. Pol. Academy of Sci., 2004.

L. Zou, J. M. Xu, and L. X. Zhu. Arterial speed studies with taxi equipped with global positioning receivers as probe vehicle. In *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, volume 2, pages 1343–1347, Washington, DC, September 2005.