Efficient Sensor Fault Diagnosis in Wireless Sensor Networks

by

Chun Lo

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2014

Doctoral Committee:

Professor Mingyan Liu, Co-Chair
Associate Professor Jerome P. Lynch, Co-Chair
Professor Anna C. Gilbert
Professor Jun Ni

To my mother

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to the many people who have helped me complete my Ph.D education. Foremost, I would like to express my sincere gratitude to my advisors Prof. Mingyan Liu (EECS) and Prof. Jerome Lynch (CEE/EECS) for their continuous support of my Ph.D research. They patiently provided encouragement, vision, and advice that allowed me to proceed through the Doctorial program and to complete my dissertation.

Besides my advisors, I would like to also thank the rest of my thesis committee: Prof. Anna Gilbert (MATH/EECS) and Prof. Jun Ni (ME), for their insightful comments and support. I also got important inspiration and guidance on part of my research from Prof. Anna Gilbert's class in compressive sensing.

I would like to give sincere thanks to Prof. Demosthenis Teneketzis (EECS) and Prof. Clayton Scott (EECS) for their insightful and interesting courses; and to Ms. Becky Turanski (EECS) for her assistance and advice throughout my graduate program.

I am thankful for my many friends in the US. I appreciate the joy and support they have provided and for burning the midnight oil together. Special thanks to Yelin Kim, Sammy Lee, Shao-Yuan Chen, Sunjoon Park, Nellie Kim, Yao Wang, Victor Chan, Wonhyung Lee, Eugene Wu, Sean O'Conner, the entire Li family, Henry Fan, AACCC, Miku Kawakami, Yumi Hasegawa, Seow Yuen Yee, Bhargav Avasarala, Takanori Watanabe and Michael Allison. Special thanks also goes to the many supporters I have from Hong Kong including my undergraduate advisor, Prof. Roger

Cheng, and my friends: Fanco Tsang, Derrick Liu, Roger Chan, Michelle Ng, Kathy Hon and Cheuk Fung Ling.

Last but not least, I would like to thank my family, especially my mother, Lai Hung Ng, for her support and love throughout my life

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Efficient Sensor Fault Diagnosis in Wireless Sensor Networks

by

Chun Lo

Co-Chair: Mingyan Liu

Co-Chair: Jerome P. Lynch

Nodes in wireless sensor networks (WSN) are prone to faults due to their inexpensive components and due to the harsh environments in which they are deployed. Therefore, automated fault diagnosis algorithms are necessary to ensure network functionality and measurement quality. Because wireless sensor networks have limited energy resources and consist of a large number of sensors, there is a need for fast and power efficient sensor fault diagnosis algorithms. This thesis proposes two frameworks of efficient sensor fault diagnosis. The first is a distributed model-based fault diagnosis framework for embedment in the WSN nodes. Fault specific algorithms are designed under this framework for detecting and identifying spike and non-linearity faults without the use of reference sensors. These algorithms fill the gap between existing centralized model-based and distributed model-free frameworks. In addition, they have the benefit of being scalable, power efficient and highly accurate. In the second framework, group testing-based fault diagnosis algorithms are proposed for situations where the number of faulty sensors is much smaller than the number of sensors in the network. These group testing algorithms evaluate sensors on a collective basis instead of on the traditional individual basis. This study designs a Kalman-filtering based method for evaluating a group of sensors to determine if faulty sensors exist in the group. This method, together with the combinatorial group testing technique, is able to detect faulty sensors in $\mathcal{O}(d^2 \log(N))$ tests, where d is the number of faulty sensors and N is the size of the network. This study also develops a Bayesian adaptive group testing algorithm in which test pools are designed based on previous group test results. This adaptive method further reduces the required number of tests and is suitable for noisy group test systems. Algorithms of both frameworks are evaluated by simulated and real sensor data with faults present. Results show that the distributed algorithms are able to achieve a detection rate of 85% or higher while keeping the false alarm rate low ($\sim$ 1%) under typical faulty signals. The group testing algorithms are able to reduce the required number of tests significantly while achieving similar accuracy as the traditional fault detection methods.

CHAPTER I

Introduction

Wireless sensor networks (WSNs) consist of a large number of distributed autonomous sensors developed to monitor physical and environmental conditions. Each sensor is capable of sensing, signal processing, and communicating wirelessly. WSNs can also be installed at low cost because wire installation is no longer necessary. This provides a great incentive for WSN adoption in large-scale systems in which monitoring is traditionally accomplished by wired sensors, such as in structural health monitoring (*Lynch*, 2007) and environmental monitoring (*Wark et al.*, 2007) systems. For example, wire installation in structural health monitoring system accounts for about 25% of the total system cost and 75% of the installation time (*Straser and Kiremidjian*, 1998). The high mobility and random deployment ability of WSNs are opening a range of new applications that had been impractical in the past such as animal tracking (*Sikka et al.*, 2004) and battlefield surveillance (*Bokareva et al.*, 2006). The ubiquitous computing paradigm (*Weiser*, 1993; *Estrin et al.*, 2002) adoption is accelerating with the help of wireless sensor technologies, whereby the physical world instrumented with pervasive networks of sensor-rich embedded systems. These systems will respond to our actions quickly and autonomously while blending into the background without people noticing them. This vision of the future has recently been termed the internet-of-things, or IoT.

To maintain small sizes and low costs, the electronics and enclosures of wireless sensors tend to be low cost thereby leaving them vulnerable to experiencing faults and failures. Moreover, the harsh environments in which WSN are usually deployed also increase the chances of abnormal sensor behavior (*Akyildiz et al.*, 2002). Therefore, wireless sensors in WSNs are more prone to error than traditional (wired) sensors. Incorrect measurements from malfunctioning sensors not only deteriorate the ability of the network to accurately take measurements but also can lead to wrong decisions which can be hazardous. Because WSNs consist of numerous sensors and these sensors are usually difficult to access after deployment, automated malfunction diagnosis capability would be an indispensable tool in maintaining the functionality and quality of the WSN.

System malfunction diagnosis has long been an important research topic in the field of control theory (*Chen and Patton*, 2012; *Frank*, 1990; *Borner et al.*, 2001; *Liu et al.*, 2009; *Venkatasubramanian et al.*, 2003). The sensor malfunction diagnosis for WSNs, however, has unique challenges and thus requires further study. The two major new challenges of sensor malfunction diagnosis in WSNs are (*Paradis and Han*, 2007):

- A WSN usually contains a large number of sensors. These sensors can generate huge amounts of data that can be difficult for diagnosis algorithms to manage and process. Consequently, malfunction diagnosis algorithms must be both efficient and scalable for large WSNs.

- Each sensor usually has a limited amount of energy, which can constrain its operation. Aggregating data from wireless sensors and transmitting data sets to powerful base stations for malfunction diagnosis is not desired because of the significant communication energy required to do so. Therefore, diagnosis algorithms are required to be energy efficient by minimizing communications and distributing computing workload across the network.

This thesis defines sensor malfunctions to be the unexpected deviation of the actual sensor measurement from the true value. The malfunction of sensors is divided into two main types: failure and faults. Sensor failure refers to the situation in which the sensor is not responding or giving measurements. This type of malfunction is often due to major system problems such as the depletion of energy or communication errors. Sensor fault refers to the situation in which the sensor is still responding and monitoring the environment. The observations of these sensors, however, are corrupted such that their measurements do not reflect the real situation. Sensor faults are less severe and are due to less critical problems, such as bad installations and poor shielding of sensors. Sensor failures are usually detected by query-and-listen methods (*Ramanathan et al.*, 2005; *Staddon et al.*, 2002; *Goodrich and Hirschberg*, 2006; *Marti et al.*, 2000). In contrast, sensor faults are more difficult to detect and characterize since the validity of the data must be analyzed. This thesis focuses exclusively on sensor fault diagnosis.

In the past decade, many studies have developed sensor fault diagnosis methods. However, only a few of them are tailor-made for WSNs and take WSN energy constraints into account. Sensor fault diagnosis contains three main stages. The first stage, malfunction detection, refers to the detection of whether malfunctioning sensors exist in the network. This stage may not necessarily point out which sensors are abnormal. The second stage, malfunction isolation and malfunction type identification, refers to the isolation of abnormal sensors and the identification of the malfunction type that is occurring in the identified sensor. The third stage, malfunction recovery, refers to the automatic correction of the detected errors. Each stage is increasingly difficult to achieve. Hence, not all malfunction diagnosis methods achieve all three stages of analysis.

In order to reveal the faulty sensors, a reference system is required. A simple method is based on hardware redundancy. This method uses duplicate sensors to perform the same monitoring task. Faulty sensors can be detected by comparing measurements between sensors (*e.g.*, (*Broen*, 1974)). This method is simple but its chief weakness is the high cost in hardware. The major approach currently in use for sensor fault diagnosis is based on the concept of analytical redundancies. The basis of this approach is that for sensors that are attached to the same physical system, correlation exists between the observations of the sensors. This correlation can be exploited to monitor the behavior of an unknown sensor. Any detected abnormal behavior is subject to further fault investigation. The general flow of faulty sensors diagnosis by analytical redundancy (*Isermann*, 1997) is summarized in three stages. First, Sensor measurements are processed to extract features that are useful for fault diagnosis. Second, these extracted features will be verified by other sensors based on the redundancies that exist in the sensor network. Examples of redundancies are correlation between sensors, system models with model parameters of the underlying system, etc. Third, any discrepancies found in the validation process is denoted as residuals. The residuals are further analyzed to detect whether there is a fault. If a fault exists, the diagnosis algorithm further isolates and identifies the fault types, if possible.

Analytical redundancy based detection methods can be classified as centralized or distributed methods and model-based or model-free methods. Centralized methods collect all the required data to a base-station and then carry out the fault detection by the base-station. In contrast, distributed methods carry out the fault detection locally, *i.e.*, near where the data is observed. The fault detection process of distributed methods usually only has local data available but it usually requires less communication energy as data only are transmitted over a short distance. Model-based methods use a mathematical model to emulate the dynamics of the system that is under monitoring. Analytical redundancies are then obtained directly from this model. The system model can be acquired by the physical properties of the system (*i.e.*, physics-based) or learned from the historical data of the system (*i.e.*, data-driven). Data-driven methods are specially valuable when the system being monitored is too complex to be modeled analytically. The acquired system model then acts as a fundamental reference system for sensor fault diagnosis. Model-free methods do not have a sophisticated mathematical model and the analytical redundancies are usually based on simple assumptions of the system. Therefore, model-based methods usually have higher accuracy (if the model is accurate) the model-free methods while model-free methods usually have lower computation requirement than the model-based methods.

Most of the model-based methods require a complete observation of the system, thereby they are usually implemented as a centralized method. On the other hand, the simple assumptions used in model-free methods can be easily implemented distributively. As a result, most of the existing methods are either centralized model-based methods (e.g., *Li et al.* (2007); *Kobayashi and Simon* (2003); *Da and Lin* (1995); *Ricquebourg et al.* (1991); *Xu et al.* (1999); *Dunia et al.* (2004); *Jiang* (2011)) or distributed model-free methods (e.g., *Ding et al.* (2005); *Chen et al.* (2006); *Luo et al.* (2006)).

Recent studies provide illustrative examples. Da *et al.* (*Da and Lin*, 1995) and Kobayashi *et al.* (*Kobayashi and Simon*, 2003) each proposed a centralized sensor fault diagnosis method that uses a bank of Kalman filters. Both methods assume that the system model is linear, and they formulate the model as a state space model. For a network of N sensors, an N-Kalman filter system is established such that the i^{th} Kalman filter is based on all but the i^{th} sensor. Assuming that there is only one faulty sensor in the network, exactly one Kalman filter (out of the N Kalman filters) behaves differently from the others. Therefore, the faulty sensor can be detected and isolated. The difference between Da's method and Kobayashi's method is the way they measure the discrepancies between the Kalman filters. When the state-space model is accurate, both methods give accurate detection results. The main constraint of both methods is at most one faulty sensor is allowed in the system.

Li et al. (2007) relaxes the at-most-one-faulty-sensor constraint by introducing reference sensors. Reference sensors refer to sensors that are known to be functioning normally during the fault detection process. Li's method requires the number of reference sensors to be more than the number of uncertain sensors (*i.e.*, those in unknown fault state). A relationship between each uncertain sensor and the reference sensors can be found based on the state-space-model of the underlying system. (In order for the relationships to be exist, the method also has a special restriction on the feed-through matrix **D** in the state-space model.) With measurements of reference sensors, the output of an uncertain sensor can be predicted. If the observed measurements of the uncertain sensor do not agree with the predicted value, the uncertain sensor will be diagnosed as faulty. Although this method allows more than one faulty sensors exist in the system, it also increases the operation cost as reference sensors are expensive to maintain.

The previous three methods use physics-based system model to establish analytical redundancies. In the following, methods that use data-driven-based system model are introduced. For example, a faulty sensor diagnosis algorithm based on a probabilistic model is presented by *Ricquebourg et al.* (1991). This study based on the result of a previous study (*Ricquebourg et al.*, 2007) which suggested the sensor data could be converted to finite symbolic states. As a result, the paper models the sensor dynamics by a Markov Chain model (*Bremaud*, 1999). A Markov chain model is a discrete time stochastic model which a state at any time is related to the previous state only. Consequently, a list of states and the transition probability between states are enough to describe a Markov chain model. The method suggested by *Ricquebourg et al.* (1991) first learns the dynamics of the system under the framework of Markov chain model based on the sensor historical data. This method is different from other fault diagnosis methods by its probabilistic nature. The sensor symbolic state estimated by the Markov model may not agree with the observed sensor state in a short time even the sensor is normal. Therefore, the diagnosis is done by evaluate the discrepancies over a long period of time. The study further constructed a 3- level decision criterion and tried to isolate the sensors which have a significant discrepancies in a short time.

Jiang et al. (2007) suggested a model-based method which distinguishes sensor faults from monitored system faults and detect abnormal changes in system time constants (*Lipták*, 2005) and gains in both sensor system and monitored system. This method assumes the normal state-space model of the system and sensor dynamics is known and the monitored system has much smaller time constant than the sensor system's. During the fault detection process, a window of the sensor measurements and system inputs are used to identify system dynamics by subspace method. If the identified model shows discrepancies from the normal model in slow dynamics, a degradation of the monitored system is concluded. Similarly, if the identified model shows discrepancies in fast dynamics region, a degradation of the sensor is concluded. The identified system model is also used to calculate the changes in the gains of the system and the sensor such that the correct gain of the sensor measurements can be corrected.

Xu et al. (1999) proposed a centralized sensor diagnosis method based on a Neural Network model. In general, Neural Networks contains an input variable layers, a output variable layers and at least one hidden variable layers. The variables in a layer affect the next layer's variables by weighting functions. These weighting functions can be non-linear. Therefore, a Neural Network model is able to capture the dynamics of complex non-linear systems. In the study, the input layer and output layer of a Neural Network are equal to the measurements of the same group of sensor. The weighting function is trained by the historical sensor measurements. When a sensor or a small number of sensors become faulty, their outputs will be detected to be inconsistent with other sensors' output. Moreover, the correct outputs can be estimated by other normal sensors based on the captured correlations. The advantage of this method is its ability to model complex and/or non-linear systems. However, the corresponding disadvantage is the high computation complexity. Also, the accuracy decreases when the number of fault sensors increase.

Dunia et al. (2004) proposed a fault detection method based on the Principal Component Analysis (PCA) model. The study assumed that the sensors are highly correlated and that therefore their output dynamics can be captured in a much smaller dimension space than the dimension of the original observation space. When the dynamics of a sensor's observation are not concentrated on the principal components, the sensor is regarded as faulty. The high dynamics outside the principal components, which are regarded as having large residuals, indicate that the sensor has lost its original correlation with other sensors, and imply sensor fault based on the system assumption. The study further analyzed how different types of faults affect the residuals differently and therefore provides a tool to identify the fault types occurring in the faulty sensors.

All of the aforementioned methods are centralized methods. A base-station has the knowledge of the system model and collects observations from the sensors for performing fault diagnosis. The data collection process inevitably consumes a large amount of energy for communication, especially for large scale networks. Therefore, centralized methods are usually not preferred for energy limited wireless sensor networks. As a result, researchers have started to develop distributive fault diagnosis methods, which generally require less transmission of data (*Chen et al.*, 2006; *Ding et al.*, 2005; *Luo et al.*, 2006).

These distributive fault diagnosis method usually have a simple assumption on the sensor relations, such as that sensors are closely deployed and thus observe similar signals. They do not require a formal system model of the underlying system and thus are regarded as model-free methods. Because of the simplicity, the diagnosis algorithm is usually more easily formulated in a distributive fashion. For instance, *Ding et al.* (2005) assumed that sensors in close proximity should have similar measurements. Therefore, the proposed method suggested that each sensor compares its output with the mean output value of the neighbor sensors. If the result of the comparison of a sensor is bigger than the results obtained by its neighbor sensors by a predefined threshold, this sensor is regarded as faulty. This method has low complexity requirement but accuracy is sensitive to the threshold value.

Chen et al. (2006) proposed an iterative fault diagnosis technique which also assumes the measurements of neighbor sensors are similar. In order to detect fault sensors, each sensor first calculates a "tendency state". This state measures how likely of the sensor being normal or faulty by comparing its own measurements with the neighbor sensors. If the comparison is larger than a threshold for more than half of the neighboring sensors, the "tendency state" is set to likely faulty, otherwise is likely normal. For a likely normal sensor, if there are more than a quarter of its neighbors are also being likely normal, the sensor will be regarded as normal and this result will be used to diagnose other sensors. Although the detection process occurs locally, each sensor is required to compare its output with its neighbors several times; at least five neighbors are recommended to ensure the method has a high level of accuracy. This drives the communication cost higher and increases the processing time, and thus has higher complexity than the method introduced by *Ding et al.* (2005).

Another distributive method was proposed for event-detection WSNs (*Luo et al.*, 2006). In this study, each sensor reports the occurrence of an event . In an event-detection network, each sensor will make a binary decision on whether a target event happened based on its observation. Luo *et al.* assumed the sensor faults are stochastically unrelated, *i.e.*, each sensor has the same probability of being faulty. A final decision of an event state is made by a majority vote from the individual decisions of a group of sensors, and sensors that frequently give different results are considered to be faulty. By showing that the error probability of final decisions decreases exponentially with the number of sensors, the study determined the optimal number of sensors that participate in the majority voting and the optimal threshold for majority voting, based on the probability for a sensor to be faulty. One of the limitations of this method is that it does not consider the case when sensors are at the boundary of an event (*i.e.*, neighbor sensors are observing different events).

A model-less faulty sensor detection method which utilizes the geometric locations of sensors is presented by *Guo et al.* (2009). This method focuses on the monitoring networks where the source of signal is located within a small area and the measurement of sensors is dependent on their geometric location. For example, the sound intensity detected by a sensor from an audio source is monotonically decreasing with the distance of the sensor from the source. The method used by Guo *et al.* divided the sensing area into different faces of subareas. For a source in any location within a face, the deployed sensors will have the same order of observed signal magnitude. The total number of possible orders of sensors is much larger than the possible number of faces. If the sensors are ordered according to the detected signal strength, faulty sensors can corrupt the order arbitrarily. For each object detection, the method maps the detected order to the closest possible order. After several detections (with the signal source appears in different faces), the faulty sensor could be isolated by comparing the detected orders and the closest orders after mapping. The limitation of this method is that the location of the sensors is required, and the faces should be updated after each movement of the sensors. Also, as the detection of faulty sensors requires the information from the entire network, this model-less sensor fault detection method is a centralized method.

In this thesis, a model-based distributed sensor fault detection method is proposed (Chapter 2, 3 and 4). The proposed method fills the gap between model-based centralized methods and model-less distributed sensor fault methods, with the goal of reducing communication energy consumption while maintaining high detection accuracy. As can be seen, most of the published studies only focus on the detection and isolation steps. *Dunia et al.* (2004) is one study that also studied fault identification and recovery. Though fault identification is more difficult to achieve, being able to identify the type of fault is important. This is because knowing the type of faults allows us to find out the cause of the fault more easily and possibly also leads to better sensor output recovery, which therefore reduces the cost of the network maintenance. Therefore, fault identification algorithms are also presented for identifying non-linearity faults and spike faults. This family of detection methods is also implemented and validated by a field study on the Grove Street Bridge located on Ypsilanti, Michigan.

A common characteristic of most of the proposed fault detection methods is that uncertain sensors are evaluated individually. We observe that while certain regional effects or disaster events may result in a large number of faulty sensors at the same time, faults occur randomly and sporadically during normal operation and in the absence of systemic problems. When faults are rare and sparse, evaluating sensors one by one seems inefficient, especially when the number of sensors in the network is large. Hence, this study develops another family of efficient fault detection methods which is based on group testing and Kalman filter. There are only a few studies that adopted group testing to detect malfunction sensors. Specifically, *Goodrich and Hirschberg* (2006) proposed a group testing based algorithm for detecting failure (dead) sensors. This algorithm uses broadcast-and-response technique to make requests to sensors that belong to a particular group, T, which then verify their existence. The responses are aggregated along the routing tree to the base station. Since $|T|$ is known, the base station is able to detect whether a group T contains any failure sensor. *Tosic and Frossard* (2012) proposed a distributive sensor fault detection algorithm that measures a smooth phenomenon (which implies neighboring sensors have similar measurements), while a group test is performed using an unspecified dissimilarity comparison of neighboring sensors' measurements. The proposed method is different from these two studies because it focuses on detecting faulty sensors which are still responsive to queries. Also, the proposed method does not have the constraint that sensors are highly correlated, *i.e.*, it does not assume neighboring sensors have similar measurements. Therefore, whether a group contains faulty sensor(s) is not as straight forward as counting the total responses from the sensors or calculating the variance of the measurements of a group of sensors. A Kalman filter-based group testing method is proposed for model-based sensor fault detection.

In the rest of this chapter, the problem statement and main structures of this thesis is presented. Consider a group of N monitoring sensors attached to a common system, as well as a base station, that together form a multi-hop wireless sensor network. Some of the sensors are able to communicate directly with the base station but most of them can communicate with the base station only through other sensors. The requirement that sensors be attached to a common system ensures the existence of correlation between sensors' observations. Each sensor periodically collects a noisy measurement of the environment. A sensor may fail at any time. This study focuses on sensor fault diagnosis, *i.e.*, diagnosis of faulty sensors that are still responding and monitoring the environment but the reporting measurements that are corrupted. We target that detecting the faulty sensors based solely on the reported measurements from the sensors. This thesis can be divided into two main parts; both focused on developing energy efficient algorithms and algorithms that are scalable for huge networks.

In the first part, we aim at developing a scalable model-based faulty diagnosis algorithm that is also able to identify fault types. We assume that the common system in which the sensors are attached is a linear system. Each sensor has a fixed location and periodically collects measurements of the system. We will develop a model-based distributive fault diagnosis algorithm. As mentioned in the previous chapter, the existing fault diagnosis algorithms are either model-based centralized methods or model-free distributive methods. Model-based centralized algorithms usually have higher accuracy but also have high communication energy consumption. Model-free distributive algorithms reduces communication significantly but it also sacrificing the diagnosis accuracy. Therefore the diagnosis algorithm has advantages from both type of algorithms, *i.e.*, energy efficiency and at the same time achieves high accuracy. Moreover, the algorithm will be able to identify and recover the type of faults, such as spike and non-linearity, in addition to performing the basic fault detection and isolation. We also chose not to use reference sensors (*i.e.*, a priori defined sensors that will perform normally during diagnosis process) of which are required in many other systems. This is a significant advantage as reference sensors usually have higher cost and/or require special maintenance.

In the second part, we focus on algorithm implementation and design issues in wireless sensor networks. As we aim at developing efficient sensor fault diagnosis algorithms that are suitable for energy-limited sensors, it is important to implement the algorithms on real sensors and evaluate the performance under real-world environments. This study can also confirm the complexity of the algorithm is suitable for low cost sensors (which equipped with 8-bit micro-processor). We implemented our algorithms on sensors that are deployed on a bridge for monitoring bridge vibrations. Sensor faults are created artificially for performance evaluation. As the proposed distributive model-based fault diagnosis algorithm partitions the network into sensor pairs. The partition affects both communication cost and fault detection accuracy. This study presents a edge covering based method for proper network partition. Moreover, this partitioning process could be carried out by the sensors, instead of a base-station, to further reduce the communication energy.

In the third part, we aim at developing fast sensor fault diagnosis algorithms for a situation in which the number of faulty sensors is much smaller than the total number of sensors in the sensor network. When most of the sensors are normal, evaluating sensors individually may not be an efficient way to perform sensor fault diagnosis, especially for applications that use thousands of sensors. A efficient fault detection method based on Combinatorial Group Testing (CGT) and Kalman filter is proposed. Each time, a group of sensor is tested and determine whether the group contains any faulty sensor. If the group members are chosen properly, the number of required group tests to detect all the faulty sensors is much smaller than the total number of sensors. In additional to the existing CGT method, we also proposed a novel Bayesian Group Testing (BGT) method. The BGT method is a adaptive method as for each group test, the group member is determined based on the previous group test results. Also, the BGT method is suitable for noisy group testing systems, *i.e.*, the result of group tests may be incorrect. Both the CGT method and BGT method are evaluated by simulations and real data collected by bridge monitoring sensors. The performance of these two methods are also compared to other Kalman filter based sensor fault detection methods.

Therefore, the contribution of the thesis can summarized as followings:

1. A distributed model-based fault detection algorithm is suggested to fill the gaps between centralized model-based methods and the distributed model-less methods.

2. Under the distributed model-based detection framework, fault type specific algorithms are proposed for detecting and identifying spike and non-linearity faults without using reference sensors.

3. A field experiment is conduct to evaluate the real-world performance of the distributed model-based fault detection algorithm. Network partition methods and insights are also given based on the field study.

4. An efficient sensor fault detection method based on Combinatorial Group Testing and Kalman filter is proposed for detecting rare faulty sensors.

5. A novel Bayesian group testing algorithm is proposed to further reduce the number of required group tests that is required by the Combinatorial Group Testing method. This Bayesian group testing method is also suitable for noisy group testing systems.

CHAPTER II

Distributed Model-based Sensor Fault Detection

This chapter proposes the framework of a Distributed Model-based sensor fault detection method. This method partitions the sensor network into sensor pairs and the dynamics of each pair of sensors is modeled by a simple linear model. This chapter lays a theoretical foundation for the sensor fault detection methodology to be described in Section III and Section IV.

Consider a set of wireless sensors attached to a time-invariant physical system. Since sensor responses all depend on the common physical system, a linear relationship exists between the system outputs measured by these sensors. This relationship can then be exploited to evaluate the "correctness" of the sensor measurements. Specifically, sensors can pair up and check whether their outputs are consistent with this linear relationship; inconsistencies can then be used to determine whether one or both of the sensors may be faulty. This pair-wise comparison can be performed between any pair of sensors and only the result of the comparison needs to be conveyed to the base station or a central processing node in the WSN. For the purpose of conserving energy in the WSN, sensors are generally grouped within close proximity of each other.

The structure of the proposed algorithm is illustrated in Fig. 2.1. The algorithm can be separated into a training (also called "model parameter identification") phase and a detection phase. During the training phase, each sensor node learns the relationship between itself and each of its neighbors. For example, in Fig. 2.1(a), sensor 2 broadcasts its measurement data to neighbouring sensors 1 and 3. After the data is received, sensors 1 and 3 calculate the relationship between their outputs and sensor 2's output (Fig. 2.1(b)). This model parameter identification processes is performed by each sensor one after another and the results are stored locally, as shown in Fig. 2.1(c). During the detection phase, the network is partitioned into pairs of neighboring sensors; this can be done centrally or in a distributed fashion. Each pair of sensors then performs a comparison according to their trained relationship. Fig. 2.1(d) shows a network partitioned into 3 pairs: $\{1,2\},\{2,3\}$ and $\{4,5\}$. For example, consider sensor pair $\{1,2\}$. Sensor 2 first transmits its output to its partner, sensor 1. Sensor 1 then checks whether its measured output agrees with the output predicted by the previously trained relationship (Fig. 2.1(e)). Finally, each sensor pair will report its results to the base station (Fig. 2.1(f)). In this architecture, fault detection is executed locally and only the diagnostic results need to be sent to the base station, thereby drastically reducing the communication cost in the multi-hop communication network. With computation and communication requirements distributed over the entire network, the WSN is more scalable to larger node counts while consuming less energy.

The relationship between sensor outputs, $y_p = f(y_q)$ is now derived. To simplify the discussion, the relationship is established between two sensors but the derivation can be generalized to sets of sensors. Consider a physical system that is represented mathematically by the following *deterministic* state space model,

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k)\ , \end{aligned} \tag{2.1}$$

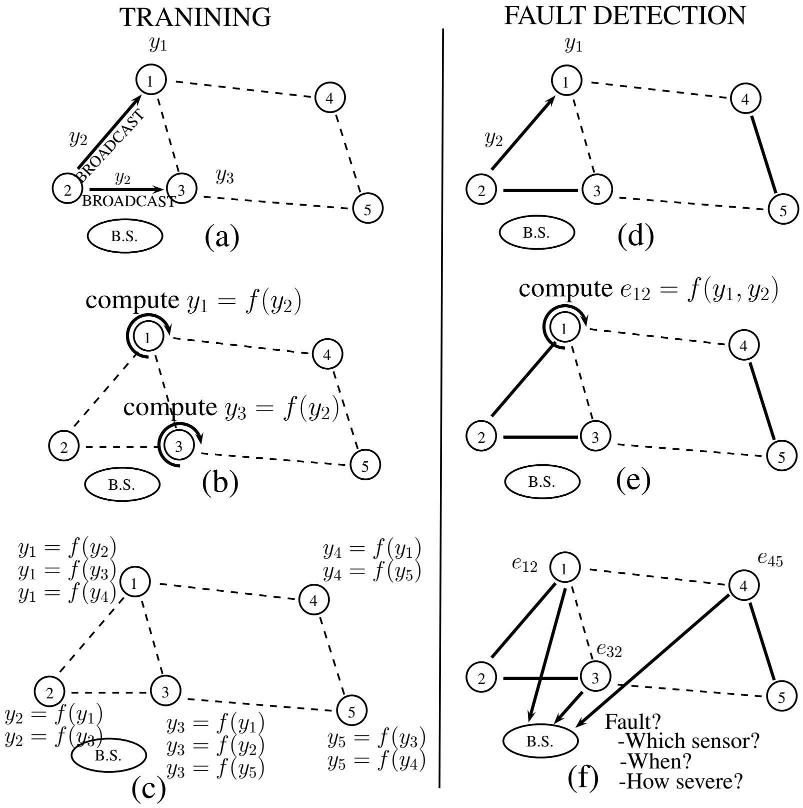


Figure 2.1: During training: (a) each sensor broadcasts its output, (b) a linear relationship between sensor pairs is calculated, and (c) finally pair-wise linear relationships of the network are constructed . For fault detection: (d) the base station divides the sensor network into pairs, (e) each pair performs the fault detection method, and (f) each pair sends their results, e, back to the base station (B.S.).

where $\boldsymbol{x}(k) \in \mathbb{R}^n$ is the state vector of the system, $\boldsymbol{u}(k) \in \mathbb{R}^l$ is the input vector, and $\boldsymbol{y}(k) \in \mathbb{R}^N$ is the output vector of the sensors. Furthermore, $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the state transition matrix which defines the transition of system states, $\boldsymbol{B} \in \mathbb{R}^{n \times l}$ is the input matrix which represents the relationship between the input and the system state, $\boldsymbol{C} \in \mathbb{R}^{N \times n}$ is the output matrix, and $\boldsymbol{D} \in \mathbb{R}^{N \times l}$ is the feed-through matrix. In the remainder of this paper, time-invariant is assumed. The system is also assumed to be stable, i.e., the output of the system, $\boldsymbol{y}$, is bounded when the system input, $\boldsymbol{u}$, is bounded.

Taking the Z-transform of Equation 6.2, the discrete-time frequency domain representation is derived:

$$\begin{aligned} z\boldsymbol{X}(z) &= \boldsymbol{A}\boldsymbol{X}(z) + \boldsymbol{B}\boldsymbol{U}(z) \\ \boldsymbol{Y}(z) &= \boldsymbol{C}\boldsymbol{X}(z) + \boldsymbol{D}\boldsymbol{U}(z) \ . \end{aligned} \tag{2.2}$$

Eliminating $\boldsymbol{X}(z)$ in the second equation of (2.2) using the first equation of (2.2), resulting in the following expression:

$$\boldsymbol{Y}(z) = (\boldsymbol{C}(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D})\boldsymbol{U}(z) \ . \tag{2.3}$$

For the p^{th} sensor, the individual observation model is,

$$Y_p(z) = (\boldsymbol{C}_p(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}_p)\boldsymbol{U}(z) \ , \tag{2.4}$$

where $\boldsymbol{C}_p$ and $\boldsymbol{D}_p$ are the p^{th} rows of the matrices $\boldsymbol{C}$ and $\boldsymbol{D}$, respectively. As a result, the transfer function $H_{pq}(z)$ between the outputs of sensor p and q is:

$$H_{pq}(z) = \frac{Y_p(z)}{Y_q(z)} = \frac{(\boldsymbol{C}_p(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}_p)\boldsymbol{U}(z)}{(\boldsymbol{C}_q(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}_q)\boldsymbol{U}(z)} \ . \tag{2.5}$$

The above expression shows that there exists a linear relationship between any pair of sensors. If the expression $(\boldsymbol{C}_p(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}_p) \in \mathbb{R}^{1\times l}$ is expressed as a polynomial function of z, then the i^{th} element of the vector can be expressed as $\sum_{j=0}^{n} a'_{ij}z^j$, where n is the rank of $(z\boldsymbol{I}-\boldsymbol{A})^{-1}$ and the coefficients a'_{ij}, $i = 1, \cdots, l,\ j = 1, \cdots, n$, are determined by the various state space matrices (i.e., $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{D}$). Similarly, the i^{th} element of $(\boldsymbol{C}_q(z\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}_q)$ can be expressed as $\sum_{j=0}^{n} b'_{ij}z^j$. Let $\boldsymbol{U}^T(z) = [U_1, \cdots, U_l]$, and the transfer function (2.5) can be written as:

$$\frac{Y_p(z)}{Y_q(z)} = \frac{\sum_{j=0}^{n}(\sum_{i=1}^{l} U_i a'_{ij})z^j)}{\sum_{j=0}^{n}(\sum_{i=1}^{l} U_i b'_{ij})z^j)} \ . \tag{2.6}$$

Notice that this relationship depends on the system input $\boldsymbol{U}$. The input driving the system dynamics may be known or can be measured by sensors; in other cases, it can be difficult to obtain. However, if the excitation of the system can be aggregated as a single source (i.e., as a scalar time-history function), then $l = 1$ and U in the numerator and denominator cancel each other. Thus the dependence on the system input in (2.5) is eliminated. Hence, a linear relationship uniquely defined by the physics of the system and the sensors measurements can be obtained:

$$\frac{Y_p(z)}{Y_q(z)} = \frac{a'_n z^n + a'_{n-1} z^{n-1} + \cdots + a'_1 z + a'_0}{b'_n z^n + b'_{n-1} z^{n-1} + \cdots + b'_1 z + b'_0} \ . \tag{2.7}$$

Reduction of the system input to a single source is quite common in many engineering systems. For example, mechanical systems excited by ambient, white noise processes and civil engineering structures exposed to base motion (i.e., earthquakes), would all be modeled by a single excitation source. In order to simplify the discussion, the scalar system input case is considered in the remainder of the paper. Note however that the following derivation and discussion remain valid under multi-input systems if they are known (or measurable) and a'_j(and b'_j) is replaced with $\sum_{i=1}^{l} U_i a'_{ij}$ (and $\sum_{i=1}^{l} U_i b'_{ij}$) in (2.7).

Both y_p and y_q correspond to system outputs. However, if y_p is viewed as the output while y_q is viewed as the *input* of another, unspecified system, then (2.7) is essentially the Z-domain representation of an autoregressive with exogenous input (ARX) time-series model (*Lennart*, 1999). ARX(ν_1,ν_2) is a linear time series model with ν_1 output terms (autoregressive terms) and ν_2 (exogenous) input terms. It is widely used to model various types of systems and natural phenomena. The definition of an ARX model is :

$$\sum_{i=0}^{\nu_1} a'_i y_q(k-i) = \sum_{i=0}^{\nu_2} b'_i y_p(k-i) \ . \tag{2.8}$$

The above derivation shows that the relationship between the outputs of two sensors is precisely captured by the ARX model, which is defined by of coefficients a'_i and b'_i. This ARX model (a'_i and b'_i) can be acquired by first storing the output pairs, y_p and y_q over a certain period of time when sensors work under normal conditions and then a'_i and b'_i are calculated from the stored data through standard least square calculations (*Bishop*, 2006; *Lennart*, 1999) or through the iterative Yule-Walker based method (*Monden et al.*, 1982b; *Pan and Levine*, 1990). Even if the historical data are corrupted by (zero mean) Gaussian noise, these training methods are able to extract accurate model coefficients. This is because when the size of the historical data is large enough, least square calculation or Burg's method is able to eliminate the variance of the noise in the data. Therefore, the ARX model training is insensitive to noise existing in the training data. After the training, only the ARX model coefficients (a'_i and b'_i) need to be stored for use in future fault detection. The length of the time history needed is equal to the dimension of the state $\boldsymbol{x}$ of the original state space model in (6.2). Consequently, the number of coefficients n of the ARX model should be equal to or larger than the size of the state dimension.

In terms of computational complexity on wireless sensors, the training of ARX coefficients requires $\mathcal{O}(\nu^2 N)$ operations for the least square regression method (*Trefethen and Bau III*, 1997) or $\mathcal{O}(\nu^2 + \nu N)$ operations for the Yule-Walker equation based algorithms (*Monden et al.*, 1982b; *Pan and Levine*, 1990), where ν is the number of ARX coefficients and N is the number of training data samples. The ARX training with Yule-Walker equation based algorithms has been implemented on real wireless sensors in a field study presented in Chapter V. In the field study, each sensor was equipped with a 8-bit, 16MHz low power controller (Atmel ATmega128). The controller takes 62.44 seconds of execution time for training the ARX coefficients with $\nu = 18$ and $N = 6000$. This is feasible in WSNs as the need for ARX model training is usually infrequent. The detection process, as shown in Chapter III and Chapter

IV, is significantly faster than the training process.

The ARX model representation of the relationship between sensor pairs is extremely valuable and will be exploited fully. While (2.7) provides a closed-form analytical expression for the relationship between sensor pairs, it would require an accurate representation of the system in the form of a state space model (i.e., knowledge of $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, and $\boldsymbol{D}$). In contrast, the equivalent ARX model in (2.8) can be determined after the network has been deployed only using sensor outputs y_p and y_q. After an ARX model has been determined between two sensors, the model is stored locally in each wireless sensor in the form of the model coefficients, a'_i and b'_i.

In the previous section, it was shown that an ARX model can accurately represent the linear relationship between sensor outputs when the number of coefficients is equal to or larger than the dimension of the system state, n. In this section, the actual sensor fault detection methodology based on ARX models is presented. As shown in the previous section, using the ARX model allows the outputs of two sensors, y_1 and y_2 to be related through the use of coefficients a'_i and b'_j:

$$\sum_{i=0}^{\nu_1} a'_i y_1(k-i) = \sum_{j=0}^{\nu_2} b'_j y_2(k-j) \ . \tag{2.9}$$

To simplify the expression, the coefficient values are normalized by letting $a_i = \frac{a'_i}{a'_0}$ and $b_j = \frac{b'_j}{a'_0}$ for $i = 0, \ldots, \nu_1, j = 0, \ldots, \nu_2$. Moreover, as in the rest of the paper, the model is assumed to have the same number of coefficients for y_1 and y_2 (i.e., $\nu_1 = \nu_2 = \nu$). Now, y_1 can be represented as a function of past outputs and the current and past outputs of y_2:

$$y_1(k) = \sum_{i=1}^{\nu} -a_i y_1(k-i) + \sum_{j=0}^{\nu} b_j y_2(k-j) \ . \tag{2.10}$$

The relationship of sensor outputs as provided by (2.10) will serve as the basis for determining if a sensor is faulty. Specifically, two error functions are defined. The first error function denoted by $e_1(k)$, is the difference in sensor 1's actual output $\tilde{y}_1(k)$ and ideal output without the fault $y_1(k)$, i.e., $e_1(k) = \tilde{y}_1(k) - y_1(k)$; the error function $e_2(k)$ is similarly defined. The second error function is the *cross-error* function denoted by $e_{12}(k)$. This is the difference between the observed output from sensor 1, $\tilde{y}_1(k)$, and the *estimated* output of sensor 1 (based on use of the ARX model using time-history data from sensor 1 and 2), $\hat{y}_1(k)$:

$$\hat{y}_1(k) = \sum_{i=1}^{\nu} -a_i \tilde{y}_1(k-i) + \sum_{i=0}^{\nu} b_i \tilde{y}_2(k-i) \ . \tag{2.11}$$

The estimated output $\hat{y}_1(k)$ is stable due to the fact that, unlike a classical observer, there is no feedback of the estimation error. Moreover, $\hat{y}_1(k)$ is a one-step estimator with inputs solely based on the sensor measurements of a stable system.

The cross-error function can then be stated as:

$$e_{12}(k) = \tilde{y}_1(k) - \hat{y}_1(k) \tag{2.12}$$

$$\begin{aligned} e_{12}(k) = {} & (y_1(k) + e_1(k)) \\ & - \left(\sum_{i=1}^{\nu} -a_i (y_1(k-i) + e_1(k-i)) \right. \\ & \left. + \sum_{i=0}^{\nu} b_i (y_2(k-i) + e_2(k-i) \right) \end{aligned} \tag{2.13}$$

$$e_{12}(k) = e_1(k) + \sum_{i=1}^{\nu} a_i e_1(k-i) - \sum_{i=0}^{\nu} b_i e_2(k-i) \tag{2.14}$$

It should be noted that (2.14) can be rewritten in a compact form using vector notation:

$$e_{12}(k) = \boldsymbol{a}^T \boldsymbol{e}_1(k) - \boldsymbol{b}^T \boldsymbol{e}_2(k) \ , \tag{2.15}$$

where $\boldsymbol{a}^T = [1, a_1, a_2, \ldots, a_\nu]$, $\boldsymbol{b}^T = [b_0, b_1, \ldots, b_\nu]$, and $\boldsymbol{e}_1^T(k) = [e_1(k), e_1(k-1), \ldots, e_1(k-\nu)]$, with $\boldsymbol{e}_2^T(k)$ similarly defined.

The cross-error function consists of a weighted summation of errors from a pair of sensors over a period of time. In general, the cross-error function gives zero values when there is no faults within the two sensors in a designated pair (i.e., they agree with each other) and gives non-zero values when any kind of faults ($\boldsymbol{e} \neq \mathbf{0}$) exist. Therefore, faults happened within a pair of sensors can be detected by the cross-error function. However, it is difficult to determine which sensor is faulty and what type of fault is present since the cross-error function superimposes the error vectors together into a scale value. Although the cross-error function may not able to identify the faulty sensor(s) directly, it provides valuable information on the faults characteristics. For instance, if $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ represent the sensor measurement noise which is i.i.d. Gaussian distributed, e_{12} provides information of the noise characteristic of the sensors. Specifically, if the Gaussian distribution (mean, variance) of the measurement noise of sensor 1 and 2 are (μ_1, σ_1^2) and (μ_2, σ_2^2) respectively, the distribution of e_{12} is Gaussian with mean equals to $\sum_{i=0}^{\nu}(a_i\mu_1 - b_i\mu_2)$ and variance equals to $\sum_{i=0}^{\nu}(a_i^2\sigma_1^2 + b_i^2\sigma_2^2)$. Based the known characteristics of the faults and the outcome of the cross-error function, specific sensor faults could be detected and isolated.

In the following two Chapters, more sophisticated algorithms are introduced for detecting and isolating two common sensor faults, Spike faults and non-linearity faults, without reference sensors. The performance of the algorithms is extensively evaluated on the detection rate, false alarm rate, signal recoverability, and separability of different fault types.

CHAPTER III

Detection and Identification of Spike Faults

This chapter presents an algorithm which detects and identifies spike faults based on the distributed model-based sensor fault detection framework introduced in Chapter II. This chapter first introduces the spike faults and its definition used in analysis. Then the detection and identification algorithm is presented. The performance of the algorithm is evaluated by both simulation and real spike corrupted sensor data.

3.1 Spike Faults

A spike fault is a voltage spike (or impulse) superimposed on the sensor measurement. Spikes typically occur randomly in time and can be constant or of varying magnitude. Here, it is assumed that each sensor can potentially suffer from a spike fault and that there are no reference (i.e., known faultless) sensors at the time of execution. The spike error could occur randomly at any time and on any sensor. It is assumed that the duration of a spike error is short and the occurrence of these spike errors is sparse (i.e., the probability that the spikes occur consecutively is low). The precise definition of sparsity to be used herein is given by: the spikes occurrence follows a Poisson process with parameter λ, (or the independent arrival times of spikes faults follows an exponential distribution with average arrival time $1/\lambda$.

3.2 Spike Detection by Matched Filters

As mentioned in Chapter II, the distributed model-based framework models a pair of sensors by a ARX model with coefficients $\boldsymbol{a}^T = [1, a_1, a_2, \ldots, a_\nu]$, $\boldsymbol{b}^T = [b_0, b_1, \ldots, b_\nu]$ learnt from historical data. With $e_i(k)$ defined to be the error happened on sensor i at time k, the cross-error function (which measures the differences between observed sensor measurement and predicted sensor measurement) has the form Eq. (2.15):

$$e_{12}(k) = \boldsymbol{a}^T \boldsymbol{e}_1(k) - \boldsymbol{b}^T \boldsymbol{e}_2(k) \ ,$$

where $\boldsymbol{e}_i^T(k) = [e_i(k), e_i(k-1), \ldots, e_i(k-\nu)], i = 1, 2$. Generally, the cross-error function fails to indicate which sensor(s) is(are) faulty because the fault information from the two sensors are superimposed together. However, if the error is due to a spike (with amplitude d) in one of the sensors, say sensor 1, at time $k-i$, then $\boldsymbol{e}_1$ will be a perfect impulse function with zero entries except for the component at $k-i$ of magnitude d (Fig. 3.1(a)). When sensor 2 is fault-free, e_{12} will be equal to $a_i d$ at component $k-i$ according to (2.14). As a result, a spike in sensor 1's output produces a cross-error function e_{12}, proportional to the ARX coefficient vector, $\boldsymbol{a}$ (Fig. 3.1(b)). Similarly, if a spike occurred in sensor 2 and no spike error occurred in sensor 1, the cross-error e_{12} will be proportional to the ARX coefficients $\boldsymbol{b}$ (Fig. 3.1(c)). When both sensors have spike errors, the cross-error function e_{12} will be equal to the sum of ARX coefficients $\boldsymbol{a}$ and $\boldsymbol{b}$ with appropriate proportionality (Fig. 3.1(d)). This insight provides a method for identifying spikes in the cross-error function and to classify the sensor fault status (i.e., no faults, sensor 1 faulty, sensor 2 faulty and both sensors faulty). In addition, this method allows the fault to be identified in time. The detection performance depends on the baseline ARX coefficient vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ which can be divided into 2 cases.

Case1: $\boldsymbol{a} \neq c\boldsymbol{b}$ for any $c \in \mathbb{R}$

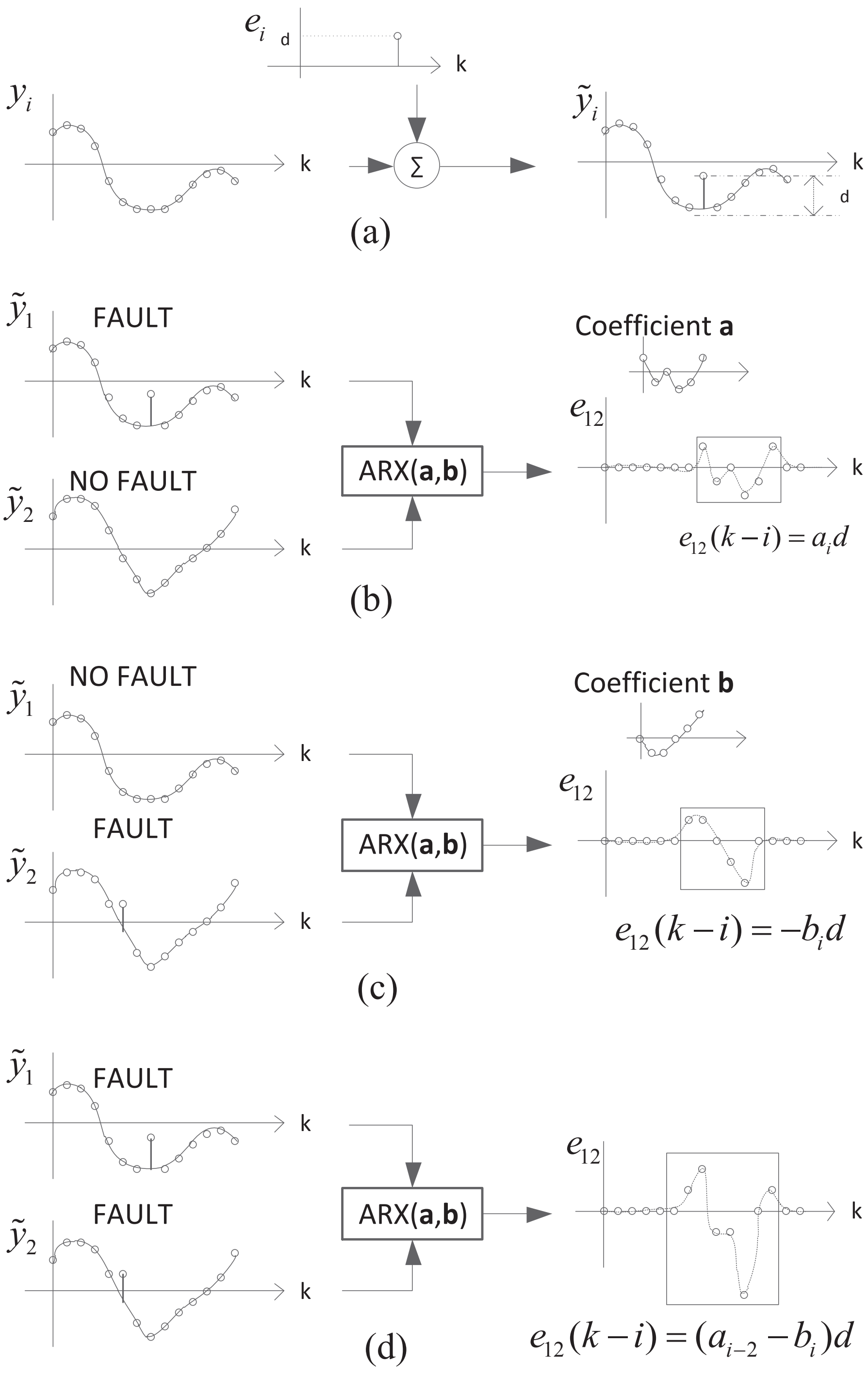


Figure 3.1: Description of cross-error functions due to spike faults: (a) the superposition of a spike fault on a sensor output; (b) sensor 1 exhibits a spike fault; (c) sensor 2 exhibits a spike fault; (d) both sensor 1 and 2 exhibit spike faults.

For case 1, coefficient $\boldsymbol{a}$ is not proportional to $\boldsymbol{b}$. As the cross-error function of a spike signal carries the characteristics of the ARX coefficients, detecting spike errors is similar to detecting a target signal with a known waveform but with unknown amplitude and delay. In the telecommunication field, one popular method to detect signals under a binary hypothesis is to use matched filtering (*Proakis*, 2007). It convolves the received unknown signal with a filter which is the same as the target signal. When the target signal arrives, it will "match" with the filter and yield a high output. To identify spike errors, the cross-error function can be passed through two matched filters which have coefficients equal to $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively. The spikes can have either positive or negative magnitude; this information is unknown to the system. As a result, only the absolute value of a matched filter's output are considered.

Consider the cross-error function $e_{12}(k)$ with measurement noise, ξ, which is additive from both sensors' noise processes. Assume there is a spike error on sensor 1 at time j ($e_1(k) = 0$ for $k \neq j$) and a spike error on sensor 2 at time l ($e_2(k) = 0$ for $k \neq l$), the cross-error function (2.14) becomes:

$$e_{12}(k) = \sum_{i=0}^{\nu} a_i e_1(k-i) - \sum_{i=0}^{\nu} b_i e_2(k-i) + \xi(k) \tag{3.1}$$

$$e_{12}(k) = a_{k-j} e_1(j) - b_{k-l} e_2(l) + \xi(k) \ . \tag{3.2}$$

If we treat the ARX coefficients $\boldsymbol{a}$ and $\boldsymbol{b}$ as the matching filter, then the output of the matched filters, defined as matched error function e_{12}^a and e_{12}^b respectively, at time k are:,

$$e_{12}^a(k) = |\sum_{i=-\infty}^{\infty} a_{i-k} e_{12}(i)| \tag{3.3}$$

$$e_{12}^b(k) = |\sum_{i=-\infty}^{\infty} b_{i-k} e_{12}(i)| \ , \tag{3.4}$$

where $a_i = 0, b_i = 0$ when $i < 0$ or $i > \nu$ (i.e., outside the range of the filter).

Replace $e_{12}(i)$ with (3.2) in (3.3):

$$e^a_{12}(k) = |\sum_{i=-\infty}^{\infty} a_{i-k}(a_{i-j}e_1(j) - b_{i-l}e_2(l) + \xi(i))| \tag{3.5}$$

$$e^a_{12}(k) = |\sum_{i=-\infty}^{\infty} (a_{i-k}a_{i-j}e_1(j) - a_{i-k}b_{i-l}e_2(l) \tag{3.6}$$

$$+ a_{i-k}\xi(i))| \ .$$

Equation (3.5) is dominated by the first two terms if the spike errors (e_1, e_2) have much larger magnitude than the measurement noise ξ. Also, the first term is maximized when $k = j$ and this maximum value is always bigger than the second term if $\|\boldsymbol{a}\|^2 \approx \|\boldsymbol{b}\|^2$ and $e_1(j) \approx e_2(l)$. This means that if a spike error appears in sensor 1, it will be enlarged by matched filter $\boldsymbol{a}$ to result in a large value in e^a_{12} and suppressed by matched filter $\boldsymbol{b}$ to result in a small value in e^b_{12}. The reverse holds true for a spike error in sensor 2. As a result, the fault detection algorithm can detect peaks in the function and discriminate the corresponding spike error in sensor 1 and sensor 2, respectively. Moreover, the matched error function can locate exactly when the spike error occurred.

The following shows the computation complexity of the detection process. Let ν be the number of ARX coefficients and N' be the number of data to be detected. From (2.11) and (2.12), the complexity for calculating the cross-error function for N' data is $\mathcal{O}(\nu N')$. Similarly, the complexity for calculation the matched filter outputs (Equation (3.3) and (3.4)) for N' data is $\mathcal{O}(\nu N')$. Therefore, the overall complexity for the detection process is $\mathcal{O}(\nu N')$.

Case2: $\boldsymbol{a} \approx c\boldsymbol{b}$ where $c \in \mathbb{R}$

When $\boldsymbol{a}$ and $\boldsymbol{b}$ can be related as $\boldsymbol{a} \approx c\boldsymbol{b}$ for some constant $c \in \mathbb{R}$, the transfer function of $\frac{\boldsymbol{Y_1}}{\boldsymbol{Y_2}}$ will be close to c. This means that the outputs of sensor 1 and sensor 2 are highly correlated to each other. For these systems, the proposed fault detection algorithm can detect a spike fault which has occurred in the sensor pair, but is not able to decide which sensor the spike fault belongs to. Intuitively, this is because the characteristic waveforms of the spike error in the cross-error function corresponding to sensor 1 and sensor 2 will have the same shape. Hence, the separating ability of the matched filter is lost. Although the magnitude and sign of the waveforms (which depend on the magnitude, d, and sign of the spike errors) are different and thus have different matched filter outputs, these outputs are not useful in identifying a faulty sensor since the sign and magnitude of the spike error is not known *a priori*. However, the detection algorithm does at least know a spike error has occurred because the cross-error function still carries the characteristics of the ARX coefficients and is thus not equal to zero. In fact, the output of sensors being highly correlated is equivalent to having hardware redundancy. If a pair of sensors gives contradicting outputs, the algorithm is not able to tell which sensor is abnormal (without knowing which sensor is normal).

For the case $\boldsymbol{a} \approx c\boldsymbol{b}$, it is possible that the method fails to tell there is a spike error within the sensor pair when both sensors have spikes at the same time with appropriate magnitude. For example, if the system output at sensor 1 is proportional to the system output at sensor 2, $y_1(k) = cy_2(k)$, and the spike errors are $e_1(k) = m$ and $e_2(k) = cm$, then the cross-error function $e_{12}(k) = \boldsymbol{a}^T\boldsymbol{e}_1 - \boldsymbol{b}^T\boldsymbol{e}_2 = 0$. Therefore, both matched filters will give zero results, which means no error is detected. In other words, if both sensors have faults such that their faulty outputs agree with each other, the method is unable to detect either error solely by evaluating the two sensor outputs.

Although there are limitations to the proposed detection algorithm, it should be mentioned that these limitations are neither common nor important cases. Having highly correlated sensors deployed in the same system is in general not cost effective because the output of one sensor is just a scale of the other. Also, the case where the algorithm is completely ineffective (i.e., when both sensors have spike errors at the same time instance with appropriate magnitude) has a very low probability of occurring because spike errors due to loose electrical contacts or electro-magnetic wave occur randomly.

Once spike errors are detected, the detection algorithm also provides a method to correct these errors. Recall that the coefficient a_0 is normalized to 1. The cross-error function, therefore, is directly representing the magnitude of the spike error if signals used in prediction $\hat{y}(k)$ are not corrupted by other spike errors and the sensor does not have background noise. Moreover, the matched error functions, e_{12}^a and e_{12}^b, reveal the position of the spike errors. As a result, the spike error can be eliminated easily.

3.3 Simulation and Results

This section verifies the performance of the proposed ARX-based sensor fault detection method. These simulations will explore the accuracy of the methodology with respect to: the magnitude of the spike fault, the frequency of spike fault occurrence, and the type of sensor measurement.

3.3.1 Simulation Settings

A generic lumped mass dynamical system will be adopted to simulate a physical system such as a bridge, vehicle, etc. Fig. 4.9 presents a five degree-of-freedom lumped mass dynamical system whose degrees-of-freedom are denoted as $x_1(k)$ through $x_5(k)$. The masses, m_i, are connected via discrete springs and viscous dampers with spring constants, k_i, and damping coefficients, c_i, respectively. An external force, $u_i(k)$, is applied to each mass. The model parameters used in (*Li et al.*, 2007) are adopted in this study. Each mass is set to be $1kg$ with each spring constant set to $10kN/m$. Similarly, each viscous damper has its damping coefficient set to $10.5Nsec/m$. The natural frequencies of the dynamical system are: 4.52, 13.22, 20.84, 26.78, and

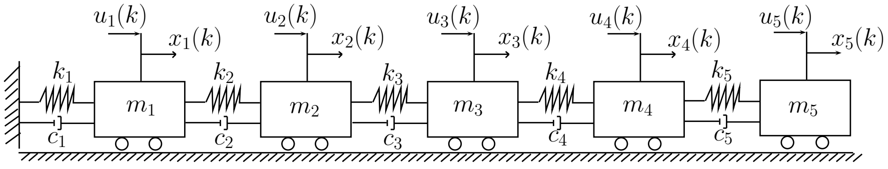


Figure 3.2: Five degree-of-freedom (DOF) spring-mass-damper system for methodology validation

$30.54Hz$. Each mode of vibration response is under-damped with damping ratios of 1.5%, 4.4%, 6.8%, 8.8%, and 10.1% for mode 1 ($4.53Hz$) through 5 ($30.54Hz$), respectively. The system is observed using either displacement sensors (i.e., $y_i = x_i$) or accelerometers (i.e., $y_i = \ddot{x}_i$) at each degree-of-freedom. The system is excited by three sources of excitation. For the first excitation, a harmonic load is applied identically to each degree-of-freedom defined by a single frequency ($\bar{w}$: uniformly distributed between 10 and 40 rad/sec) and with a random amplitude ($|U_{max}|$: uniformly distributed between 10 and $13N$) and offset. For the second excitation, the external load is again harmonic but with two major frequencies ($\bar{w_1}$: uniformly distributed between 10 and 40 rad/sec and $\bar{w_2}$: uniformly distributed between 100 and 150 rad/sec). The amplitude associated with both tones is also random ($|U_{max}|$: uniformly distributed between 10 and $13N$). The third excitation is a white noise source with a variance of 100N identically applied to each degree-of-freedom. Table 3.1 summarizes the excitations used; all three are used for training ARX models and for spike fault detection validation. For each single simulation, the parameters of the excitation are randomly chosen and fixed through out that simulation. It should be noted that the excitation, even of the same type, are different (i.e., generated separately) for training and testing simulations.

For the training of the ARX pair-wise time series models, all three excitation types are utilized with a unique ARX model found for each excitation and measurement type. A validation analysis is done to determine the optimal number of coefficients (*Bishop*, 2006). Here, the model order is set to 25 a_i coefficients and 25 b_i coefficients.

As previously mentioned, the coefficient a_0 is set to 1.

In the simulations, the system is excited by the external excitation and the system response is measured at each degree-of-freedom, y_i. Random Gaussian noise is added to all measured responses to simulate a low level of sensor measurement noise. To emulate sensor spike faults, spikes are introduced to the measurements of degree-of-freedom one and two with random sign and magnitude. For example, consider the fault free sensor response at m_1 and m_2 under the two-tone excitation as shown in Fig. 3.3(a). An example of the two time histories with simulated spike faults with the actual spikes denoted is shown in Fig. 3.3(b). Each time history in Fig. 3.3 has 4% noise (with respect to the signal variance) introduced. To determine the accuracy of the proposed sensor fault detection algorithm, the detection rate is used as a metric. Detection rate is the percentage of spikes to be correctly identified by the algorithm. This is equivalent to the percentage of true-positives; related metric would be the percentage of false-positives.

For illustrative purposes, consider the measured displacement of mass m_1 and mass m_2 denoted in Fig. 3.3(b). Using the ARX pair-wise model between y_1 and y_2, the output at m_1 is predicted by the output at m_2. The difference in the predicted and measured output, e_{12}, is plotted in Fig. 3.4(a). As can be seen, the spike faults in both outputs is creating non-trivial elevations in e_{12} in the vicinity of the actual faults. However, which sensor in the pair is experiencing the faults cannot be determined by the cross-error function alone. Rather, the use of the matched filters is needed to determine which sensor has the fault and where in time the faults are located. Use of (3.3) and (3.4) are used to determine e^a_{12} and e^b_{12} in Fig. 3.4(b) and Fig. 3.4(c), respectively. As can be seen, the convolved error function reveals when the spike faults occur. To identify the spikes, a threshold level is defined. Any disturbance that is larger than the threshold will be declared as a spike error of the corresponding sensor. For example, e^a_{12} exceeding a defined threshold corresponds to

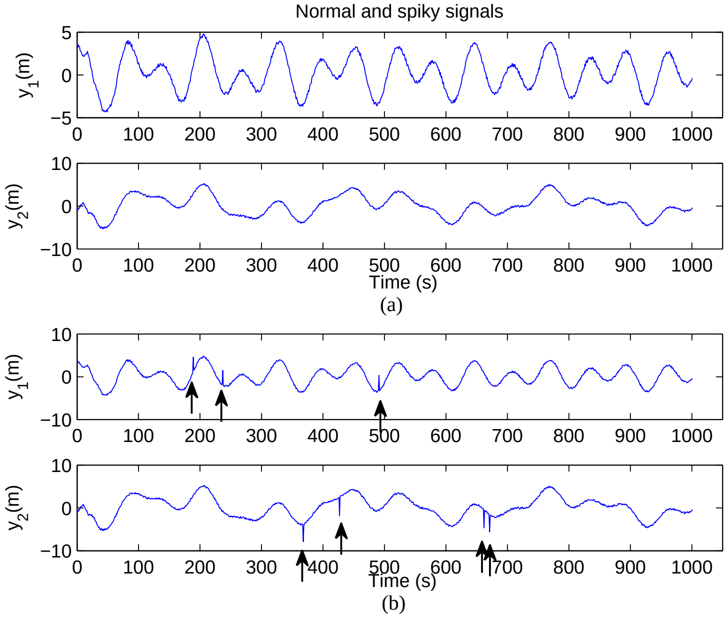


Figure 3.3: Response of m_1 and m_2 under a two tone harmonic excitation: (a) fault free displacement time history; (b) same displacement time histories with spike faults (40% of the maximum response amplitude) randomly introduced.

faults in y_1 while exceedance of a threshold in e_{12}^b corresponds to faults in y_2. For all the simulations presented, the threshold is determined by the following steps. First, the cross-error function, e_{ij}, is calculated by the trained ARX model for a pair of normal (fault-free) sensors. The cross-error function is not expected to be equal to zero because there exists observation noise in the sensors and the ARX model will also have some prediction error. Afterward, the cross-error function is passed through the two matched filter and the variance of the e_{ij}^a and e_{ij}^b calculated. The threshold is set to be 6 times the standard deviation of the convoluted error function of the fault-free sensors such that the false alarm rate caused by the sensor measurement noise is almost zero.

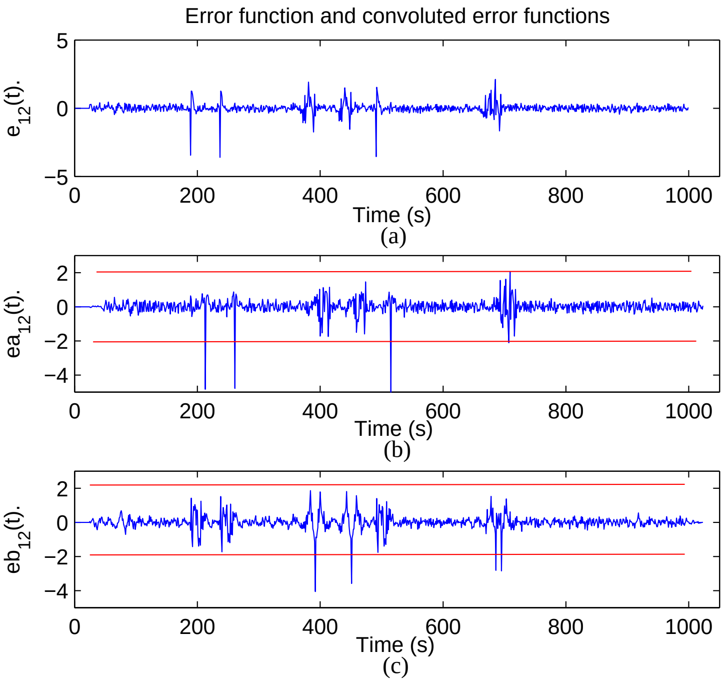


Figure 3.4: (a) Cross-error function between sensor 1 and 2 corresponding to outputs presented in Fig. 3.3(b); (b) error function convoluted with coefficients $\boldsymbol{a}$; (c) error function convoluted with coefficients $\boldsymbol{b}$.

3.3.2 Simulation Results under Various Scenarios

Simulations are carried out to evaluate the detection rate of spike errors versus different spike error amplitudes when measuring the acceleration and displacement response of the spring-mass-damper structure. These simulations considered the three different combinations of system excitations for both training and testing (validation). The legends of the different excitation combinations for Fig. 3.5 to 3.9 are summarized in Table 3.1. For example, if the system response to the single-tone sinusoidal excitation is used during training and white Gaussian noise excitation is used during performance testing, a red solid curve with triangle markers is used for the detection curve. In the simulations, the sensor noise is set to 10% of the sensor output variance. As can be seen for both displacement (Fig. 3.5) and acceleration (Fig. 3.6) outputs, the detection accuracy of the proposed method increases with the amplitude of the

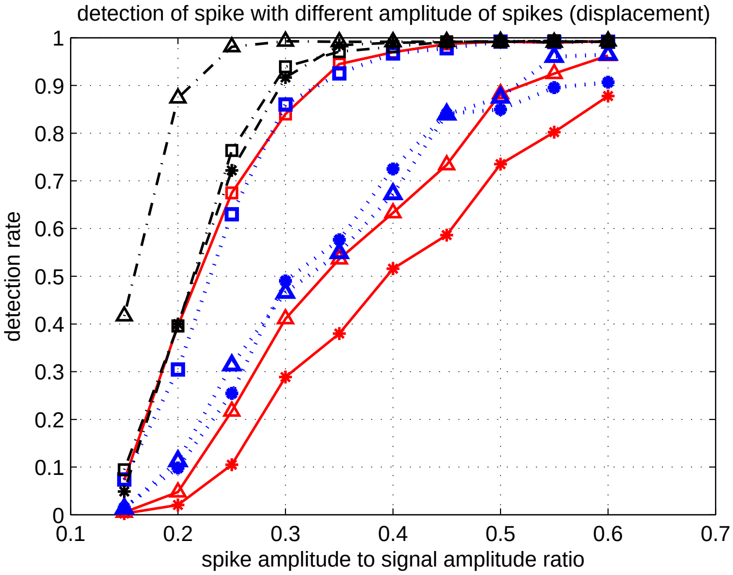


Figure 3.5: Detection rate of spike faults versus different spike amplitudes when measuring displacement.

spikes. This is an obvious finding because smaller amplitude spike faults are more likely to be obscured by the sensor noise and thus more difficult to identify. Fig. 3.5 presents the detection rate when the sensors measure the displacement of the masses in the spring-mass-damper system. The results show that the detection algorithm performed well in all system input combinations. For large spike faults such as those whose amplitudes were 60% or greater than the signal amplitude, the sensor fault detection rate of the algorithm was high ($> 90\%$) regardless of the excitation used to train ARX models or when determining the sensor faults. However, for smaller

Excitation	Training	Testing
Single-tone harmonic $\bar{w} = [10\ 40] rad/sec$ $\lvert U_{max}\rvert = [10\ 13]N$	Solid line (Red)	Square
Double-tone harmonic $\bar{w}_1 = [10\ 40] rad/sec$ $\bar{w}_2 = [100\ 150] rad/sec$ $\lvert U_{max}\rvert = [10\ 13]N$	Dotted line (Blue)	Star
White Gaussian signal $variance(\sigma^2) = 100N$	Dash-dot line (Black)	Triangle

Table 3.1: Simulation excitations

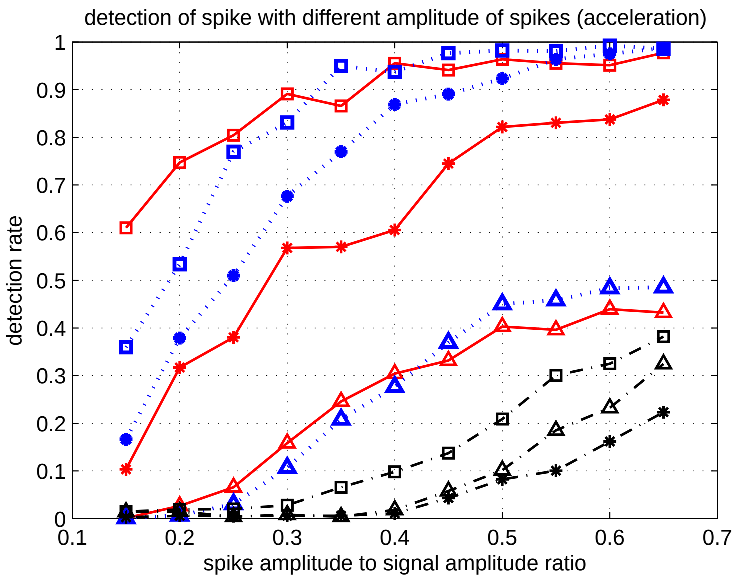


Figure 3.6: Detection rate of spike faults versus different spike amplitudes when measuring acceleration.

spike amplitudes, the method accuracy exhibits some dependency on the nature of the excitation used to train the ARX models. In general, ARX models trained from white noise excitations provided the best baseline models. Especially when used to determine sensor spikes from similarly broadband excited time-history outputs, even small spikes (e.g., spikes white amplitudes only 25% of the signal amplitude) can be detected with detection rates in exceedance of 98%. Even for systems excited by harmonic loads, sensor faults with spike to signal amplitudes of 0.3 have detection rates of 90% or greater. In the absence of white noise excitations, the more broadband an excitation is, the better suited it is for the training of the ARX relationships between sensor pairs. For example, Fig. 3.5 shows that the ARX models created using the response of the system to the double tone harmonic excitation were more effective compared to those created using the single tone harmonic response.

When measuring displacement, the ARX models fall into Case 1 as discussed in Section 3.2 (i.e., $a \neq cb$ for all $c \in \mathbb{R}$). In contrast, when measuring acceleration of the system, the system outputs are more correlated. This fact is confirmed when

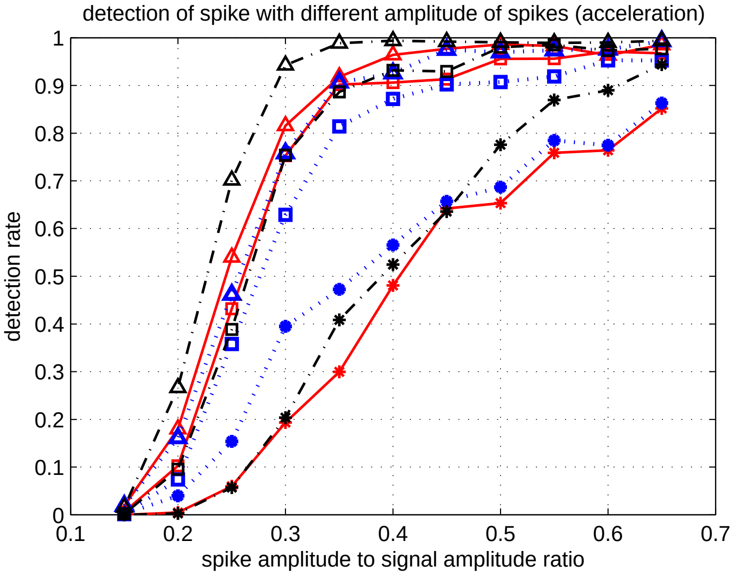


Figure 3.7: Detection rate of spike faults versus different spike amplitudes when measuring acceleration. The viscous damping constant is decreased from 10.5 to $0.6 Nsec/m$.

investigating the ARX model coefficients of relationships trained between various system outputs. With $a \approx cb$, the detection accuracy of the method decreases. For example, Fig. 3.6 presents the detection rate when the sensors measure the acceleration of the system. The performance of the sensor fault detection method in this setting is similar to the one measuring displacement except that the combinations in which Gaussian white noise is used as a training signal appear to perform worse with detection rates significantly lower than the other input combinations. When the system is excited by Gaussian white noise, its acceleration response exhibits the greatest correlation resulting in the lowest detection accuracy. While the detection algorithm is able to detect that faults exist in the sensor pairs, it is not able to classify which sensor the fault belongs to resulting in low detection rates.

To verify the deteriorated performance presented in Fig. 3.6 is due to strong correlation in the measured system response, the viscous damping coefficient of the spring-mass-damper system is reduced from $c_i = 10.5 Nsec/m$ to $c_i = 0.6 Nsec/m$. All the other model parameters remain the same (i.e., $m_i = 1kg$ and $k_i = 10kN/m$).

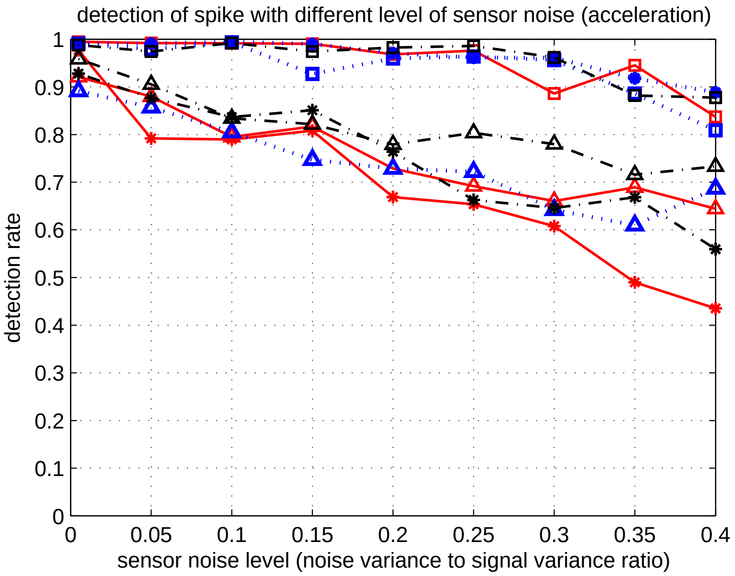


Figure 3.8: Detection rate of spike faults versus different levels of sensor observation noise when measuring displacement. Spike are fixed at 30% of the maximum peak-to-peak amplitude of the sensor measurement.

ARX models fitted to the new acceleration data shows $a \neq cb$ resulting in a Case 1 system. The detection rate results are shown in Fig. 3.7, with the detection rate similar to that when using displacement outputs (Fig. 3.5).

Next, the accuracy of the sensor fault detection algorithm is quantified for noisy sensor measurements. Here, the original system is used (i.e., $c_i = 10.5Nsec/m$). Fig. 3.8 shows the detection rate of spike errors versus different levels of sensor observation noise while measuring the displacements of the system degrees-of-freedom. The magnitude of the spike errors are fixed at 30% of the maximum peak-to-peak amplitude of the sensor outputs. As can be seen, the detection accuracy deteriorates when the sensor noise exceeds 20% in all cases. When noise increases, the response of the spike error in the matched filter output will be increasingly dominated by noise. As a result, it is harder to set a good threshold for detection since the threshold must rise to be above the noise level.

Finally, the performance of the fault detection algorithm is investigated under scenarios of sensor faults occurring at the same time on both sensors in a pair. During

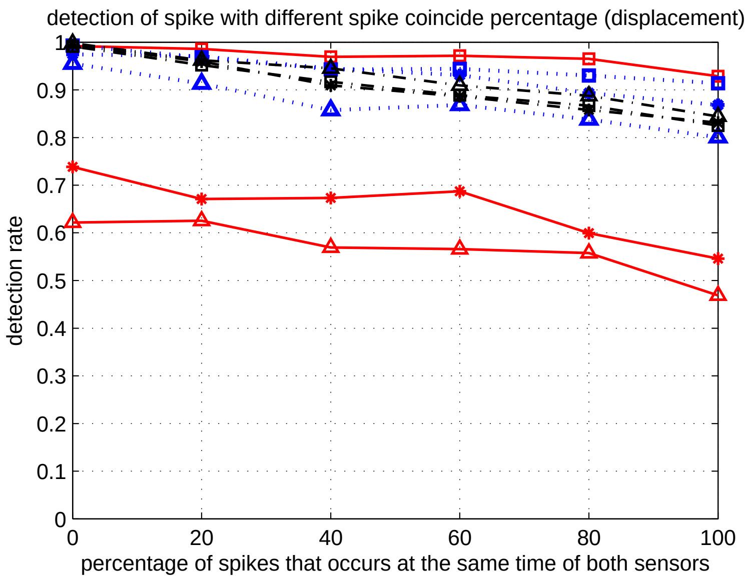


Figure 3.9: Detection rate of spike faults versus different percentage of coincide spike error on both sensors.

this simulation, system displacements are measured with sensor noise fixed at 10% of the sensor output variance. Fig. 3.9 investigates the situation when spike faults happen at the same time on both sensors. The sign and magnitude of spikes are random in this simulation. The plot shows the detection rate as a function of the percentage of spike errors that happen at the same time on both sensors. Although the spike errors of different sensors have different characteristic waveforms and match to different matched filters, the coincident occurrence of spike errors still affects the detection performance. Due to faults occurring at the same time, the characteristic waveform on the cross-error function can be partially canceled out by the other spike with appropriate sign. As a result, the detection accuracy decreases (slowly) with the rate of coincidence. However, it should be noted that the overall performance of the algorithm is still high (> 0.8 detection rate for most cases).

In all of the simulations presented, the threshold level is set to prevent false alarms. Hence, most of the errors in the detection algorithm are misses. In fact, higher detection rates can be achieved if false alarms can be tolerated by changing

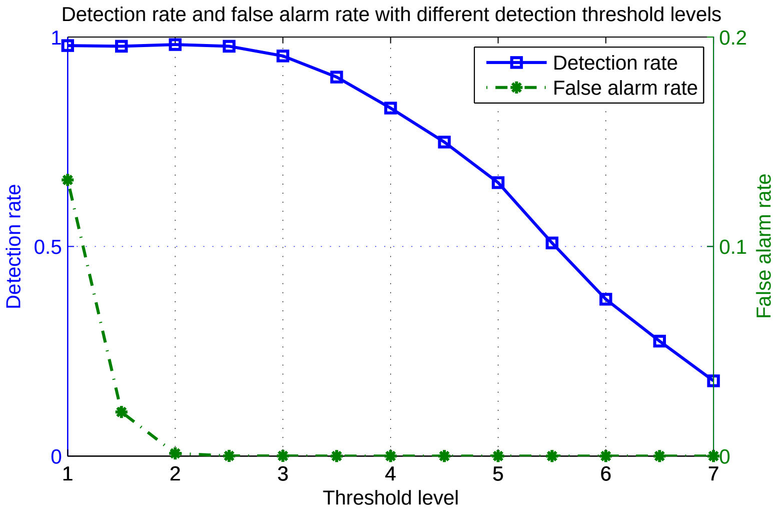


Figure 3.10: The trade off between the sensor fault detection method's detection rate and false alarm rate with different threshold levels.

the threshold level. Fig. 3.10 illustrates the trade off between the detection rate and false alarm rate with different thresholds. The magnitude of the spike errors are fixed at 30% of the maximum peak-to-peak amplitude of the sensor outputs and the sensor noise is set to 10% of the sensor output variance. The rest of the simulation settings are the same as the simulation shown in Fig. 3.5. The threshold level is the multiplier of the variance of the convoluted error function of the fault-free sensors. As can be seen, lowering the threshold level can increase the detection rate while the false alarm rate also increases rapidly.

3.3.3 Performance on Real Spike Corrupted Sensor Data

The rest of this section shows the performance of the proposed method on sensor data from a field-deployed WSN. A previous study was conducted focused on reducing the cost and installation complexity of monitoring systems on ships; wireless sensors were proposed by Swartz, *et al.* (*Swartz et al.*, 2012) as an alternative to traditional wired sensors. A monitoring system consisting of 20 wireless sensors were installed on a U.S. Navy ship in 2008. Moreover, a traditional wired hull monitoring system was installed in the ship alongside the WSN. During sea trails, some of the wireless sensors suffered from spike errors and excessive noise. Consider two wired sensors measuring hull strain shown in Fig. 3.11; these sensors were from the wired monitoring system and are generally fault-free (i.e., no spikes, etc.) Two wireless sensors in the WSN were collocated with these wired sensors and had spike errors and noise. These faulty sensors are used to illustrate the sensor fault detection algorithm in this study. An ARX model of order $\nu = 30$ is trained using 60 seconds (6000 points) of the spike free signals from the two wired sensors. With the trained ARX coefficients, 270 seconds (27000 points) of strain signals from the wireless sensors with spike errors are examined by the proposed method. Part of the signals (3000 points) of the two wireless sensors are shown in Fig.3.12(a) (sensor s1) and (c) (sensor s2). The spikes detected by naked eye are marked by squares in Fig.3.12(a) and (c). The output of the matched filters is shown in Fig.3.12(b) and (d), and the automatically detected spikes are marked in stars. As can be seen, 21 out of 25 (84%) spikes were detected with 2 false alarms. For the examination of 27000 data points, 83.3% of spikes were detected (194 out of 233) and the false alarm rate is 0.04%. This level of accuracy is regarded as impressively high for actual field deployed wireless sensors with moderate amounts of noise.

3.4 Summary

This chapter proposed a Matched filter based algorithm for detecting and identifying spike faults without *a priori* establishment of reference sensors. In the simulations conducted, the detection accuracy exhibited dependency on the magnitude of the spikes, the sensor observation noise, the ARX coefficients and the threshold set for spike detection after the matched filter. The algorithm gives good performance; it only loses part of its effectiveness under situations where a pair of sensors is highly correlated. However, such situations are either unlikely in practice or can be avoided when pairing sensors for execution of the proposed method. Moreover, since the de-

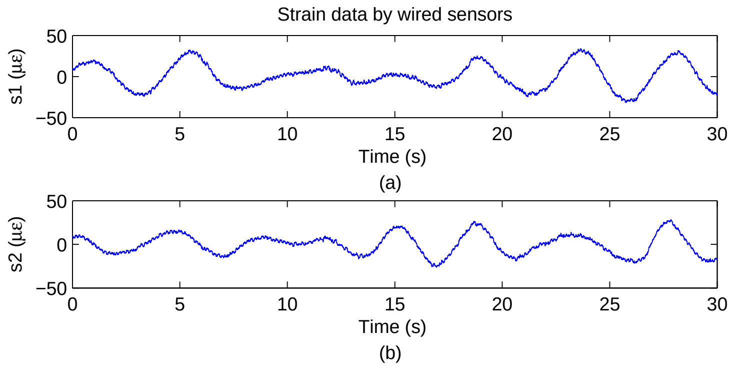


Figure 3.11: Strain signals from wired sensors.

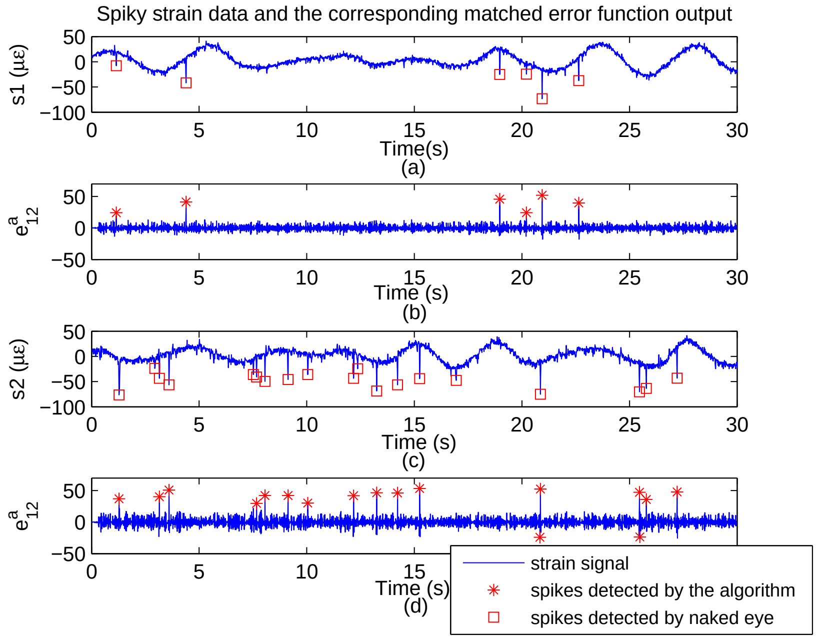


Figure 3.12: Strain signals form wireless sensors with spike errors ((a) and (c)) and the corresponding output of the matched error function ((b) and (d))

tection is done on a pairwise basis, it is well suited for WSNs in which power and communication resources are limited.

CHAPTER IV

Detection and Identification of Non-linearity Faults

This chapter presents an algorithm which detects and identifies non-linearity faults based on the distributed model-based framework introduced in Chapter II. This chapter first introduces the characteristics of non-linearity faults. Two different non-linearity fault models are introduced for algorithm evaluations. The detection of sensor nonlinearities is shown to be equivalent to solving the largest empty rectangle (LER) problem given a set of features extracted from the cross-error function. A low-complexity algorithm that gives an approximate solution to this problem is proposed so that it is feasible, it can be embedded in low-power wireless sensors. By solving the LER problem, sensors corrupted by non-linearity faults can be isolated and identified. Finally, extensive analysis is performed to evaluate the performance of the proposed algorithm through simulations.

4.1 Non-linearity Faults

Non-linearity faults are a multiplicative measurement error which depend on the sensor output. A non-linearity fault can be represented by a nonlinear transfer function between the true signal and the output of the sensor. The function (Fig. 4.1) has two regions: normal and abnormal. When the actual signal is within the normal region, the measurement correctly reflects the actual signal. When the sensor signal falls into the abnormal region, the sensor transfer function is altered leading to errors in the measurement. In this study, two simplified non-linearity models are used. The first one is the bilinear model of Fig. 4.1. Within the normal region, the measurement is exactly equal to the true signal. Within the abnormal region(s), which is the compliment of the normal region, the function between the measurement and true signal follows another linear function with a different slope. The mathematical expression of the bilinear model is:

$$y = \begin{cases} x & \text{if } r_2 < x < r_1, \\ \tan(\theta_1)(x - r_1) + r_1 & \text{if } x \geq r_1, \\ \tan(\theta_2)(x - r_2) + r_2 & \text{if } x \leq r_2. \end{cases} \tag{4.1}$$

where x is the actual signal, y is the sensor measurement and $r_1 > 0,\ r_2 < 0$ define the boundaries of the normal region. Moreover, the slopes θ_1 and θ_2 have the range $[0, \pi/2)$. The parameters of the bilinear model are the normal region boundaries, r_1, r_2 and the slope of the linear profile in the abnormal region. A non-linear model can have either one-sided or two-sided abnormal region(s). For example, if $r_1 < \infty$ and $r_2 = -\infty$, we have a one-sided abnormal region. The measurement-true signal transfer function of (4.1) as illustrated in Fig. 4.1 is used to model the measurement of a sinusoidal signal with unit amplitude. The bilinear fault is assumed to be one-sided with $r_1 = 0.6$ and $r_2 = -\infty$; also, $\theta_1 = 30°$. As shown in Fig. 4.2, the faulty signal is evident in the positive amplitude with the amplitudes lower than the true signal amplitude.

The second non-linear model uses exponential functions to model the sensor's abnormal region(s). Within the normal region, the measurement is equal to the actual signal. Within the abnormal region, the function between the measurement

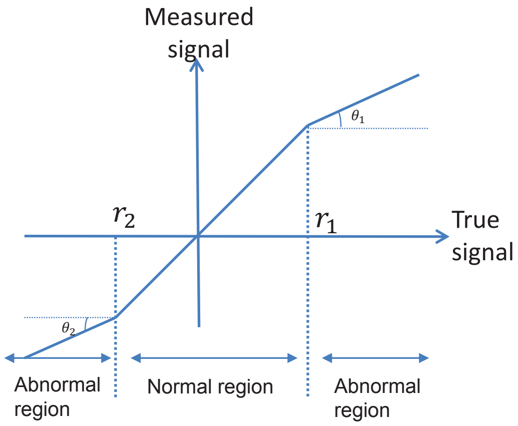


Figure 4.1: Bilinear non-linear sensor fault model

and true signal follows an exponential profile as defined by:

$$y = \begin{cases} x & \text{if } x < r_1, \\ \max\{x - (e^{(\psi|x-r_1|)} - 1), \frac{1}{\psi}\ln\frac{1}{\psi} - \frac{1}{\psi} + r_1 + 1\} & \text{if } x \geq r_1, \texttt{Type I} \\ \max\{x + (e^{(\psi|x-r_2|)} - 1), -\frac{1}{\psi}\ln\frac{1}{\psi} + \frac{1}{\psi} + r_2 - 1\} & \text{if } x \leq r_2, \texttt{Type I} \\ x + (e^{(\psi|x-r_1|)} - 1) & \text{if } x \geq r_1, \texttt{Type II} \\ x - (e^{(\psi|x-r_1|)} - 1) & \text{if } x \leq r_2, \texttt{Type II} \end{cases} \tag{4.2}$$

where α, β, r_1 and r_2 (with $r_1 > 0$ and $r_2 < 0$) are the parameters of the model. Compared to the bilinear model, the function of the exponential model has a gradual change from the normal region to the abnormal region. Fig. 4.3 shows the functions of a one-sided exponential with varying model parameter α and with normal region boundary set at $r_1 = 30$. The output saturation problems of amplifier-based sensors, where measurements are smaller than the actual signal, are best modeled by the exponential nonlinear fault model (Type I). On the other hand, excessive gain problems (such as sensor resonance), where measurements are larger than the actual signal are best model by Type II exponential functions.

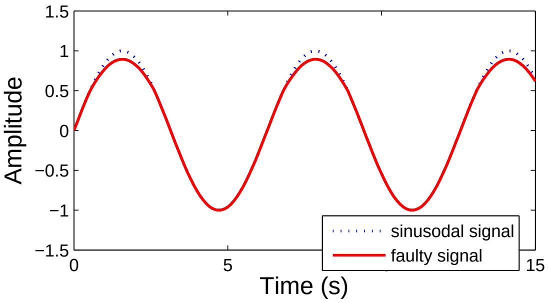


Figure 4.2: A unit amplitude sinusoidal signal and the corresponding non-linearity faulty measured signal with $r_1 = 0.6$, $r_2 = -\infty$ and $\theta_1 = 30°$

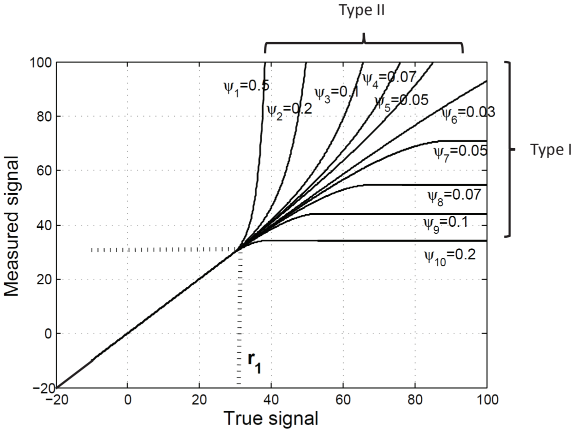


Figure 4.3: Exponential non-linearity sensor fault model (one-sided version shown).

4.2 Non-linearity Fault Detection and Identification Methodology

Consider a N-sensor WSN where each sensor has the potential to suffer from non-linearity faults but the network has no knowledge of which sensors are normal. The WSN is partitioned into peer-to-peer sensor pairs as shown in Fig. 4.4 (*e.g.*, $y_2 - y_4$, $y_1 - y_3$, $y_5 - y_6$ and $y_6 - y_7$). As mentioned in Chapter II, the relationship between any pair of sensors can be captured by an ARX model trained using sensor outputs. The model parameters can be learned from the historical data by a fast iterative algorithm (*Monden et al.*, 1982a) when the sensors are working normally. Fig. 4.5 provides an overview of the detection method proposed to detect and identify non-linearity faults

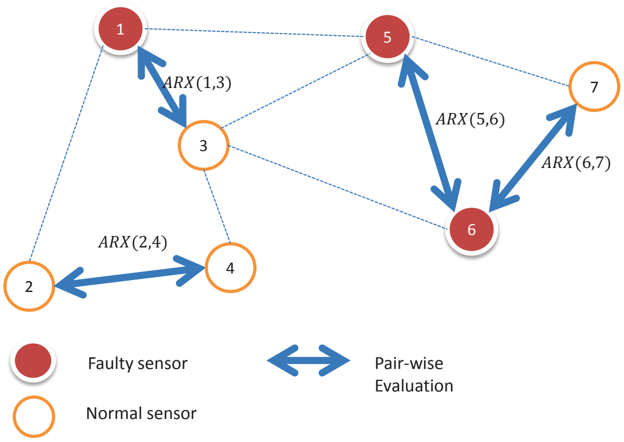


Figure 4.4: Pair-wise fault detection in a sensor network

within a pair of sensors. There is no a priori knowledge of the health of sensor 1 or 2 in this approach (*i.e.*, both could be faulty). First, sensor 1, S_1, transmits it output, $\tilde{y}_1$ to sensor 2, S_2. S_2, then uses its measured output, $\tilde{y}_2$, to predict S_1's output, $\hat{y}_1$, using a previously trained ARX model between the $S_1 - S_2$ pair. The difference between the estimated signal, $\hat{y}_1$, and measured signal, $\tilde{y}_1$, constitutes a cross-error function, which represents the signal measurement accuracy of the sensor pair. As will be described, this error function will be analyzed to extract feature points (say P_{S_1} and P_{S_2}) which are useful for fault detection and identification. Under the largest empty rectangle (LER) identification process, the faulty sensor(s) within the sensor pair can be isolated and the non-linear fault model parameters can be identified. This fault diagnosis process can be carried out in parallel between each sensor pair and thus the proposed fault diagnosis method is scalable as the size of a WSN grows. In the rest of this section, each component of the fault diagnosis methodology will be discussed in detail.

4.2.1 Feature Point Calculation

Before introducing the non-linearity identification method, the sensor output is assumed to transgress into the abnormal region of the non-linearity fault model occa-

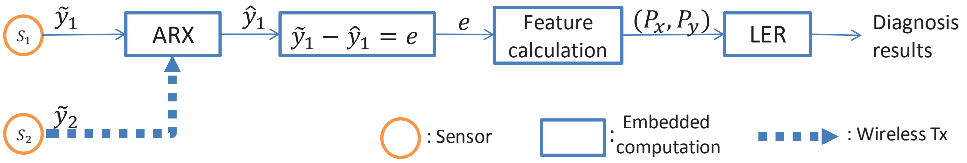


Figure 4.5: Overview of detecting and identifying non-linearity faults within the $S_1 - S_2$ pair.

sionally. In other words, the normal region of the sensor covers a significant portion of the signal range. This assumption is valid for most systems if appropriate sensors (with large enough sensing dynamic range) are chosen for the monitoring task. Recall the cross-error function, $e_{12}(k)$, is defined to be the difference between the observed output and estimated output:

$$\begin{aligned} e_{12}(k) =& \tilde{y}_1(k) - \hat{y}_1(k) \\ =& \sum_{i=0}^{\nu} a_i e_1(k-i) - \sum_{i=0}^{\nu} b_i e_2(k-i) \end{aligned}$$

The cross-error function utilizes past outputs of S_1 and S_2 over a time window from time 0 to time $-\nu$ (relative to the time index, k, of the cross-error function in (2.14)). The cross-error function is non-zero whenever a measurement error (*i.e.*, where the abnormal range is entered) occurs within the ARX lag window on any sensor. It should be noted that it is possible that errors between two sensors can cancel each other out, but this is an extremely rare event. Therefore, when the cross-error function is zero, it is almost certain that there is no fault within the time window. However, when the cross-error function is non-zero, it is uncertain when and which sensor the error occurred in unless it is the first time the cross-error function output deviates from zero (or near zero). This is because the cross-error function output can only be non-zero due to an error initiating at time k in one or both of the sensors. The measurement value of the faulty sensor at the time the cross-error function experiences its first non-zero value should fall in the abnormal region and thus have magnitude bigger than r_1 or r_2 if the sensor is corrupted by a non-linearity fault. However, it is not clear which sensor(s) is (are) faulty as the normal region boundaries ($r_1,\ r_2$) of the non-linearity model remain unknown. As a such, a feature point, $\mathbf{P}$, is defined as $\mathbf{P} = (P_{S_1}, P_{S_2})$. P_{S_1} and P_{S_2} are the measurement values of both sensors ($\tilde{y}_1$ and $\tilde{y}_2$) at the time the cross-error function first deviates from zero and exceeds a pre-defined threshold. Therefore, the total number of feature points available is equal to the number of times the cross-error function exceeds the threshold. Fig. 4.6 shows an example of the way the feature points are extracted. In this example, $S1$ is faulty with a non-linearity fault and $S2$ is normal. Fig. 4.6(a) and(b) are the true signals of sensor 1 and 2 (denoted as S_1 and S_2), respectively. Fig. 4.6(c) is the (corrupted) signal measured by faulty sensor 1, $\tilde{y}_1$, and Fig. 4.6(d) is the calculated cross-error function. As can be seen, four feature points are available in this example.

After a set of feature points, $\wp$, are generated over a time history, they can be plotted on a 2-dimension plane with the x-axis corresponding to $S1$ (P_{S_1}) and the y-axis corresponding to $S2$ (P_{S_2}). The plot will have the patterns shown in Fig. 4.7 when the errors are caused by a non-linearity fault. In Fig. 4.7, the dotted lines represent the boundaries of the normal region of a non-linearity model (if the sensor is faulty). When $S1$ is corrupted by a non-linearity fault and $S2$ is normal, the x-coordinates of the set of collected feature points fall in the abnormal region while the y-coordinates of the feature points can have any value. Therefore, the feature points only fall into the region highlighted in Fig. 4.7(a). Similarly, Fig. 4.7(b) shows the regions the feature points should fall in when $S1$ is normal and $S2$ is corrupted by a non-linearity fault. When both sensors are corrupted by a non-linearity fault, the collected data should fall in the highlighted regions in Fig. 4.7(c).

Consequently, if the sensor pair under investigation can be classified into one of the 3 different patterns in Fig.4.7, the faulty sensor(s) can be isolated even if there

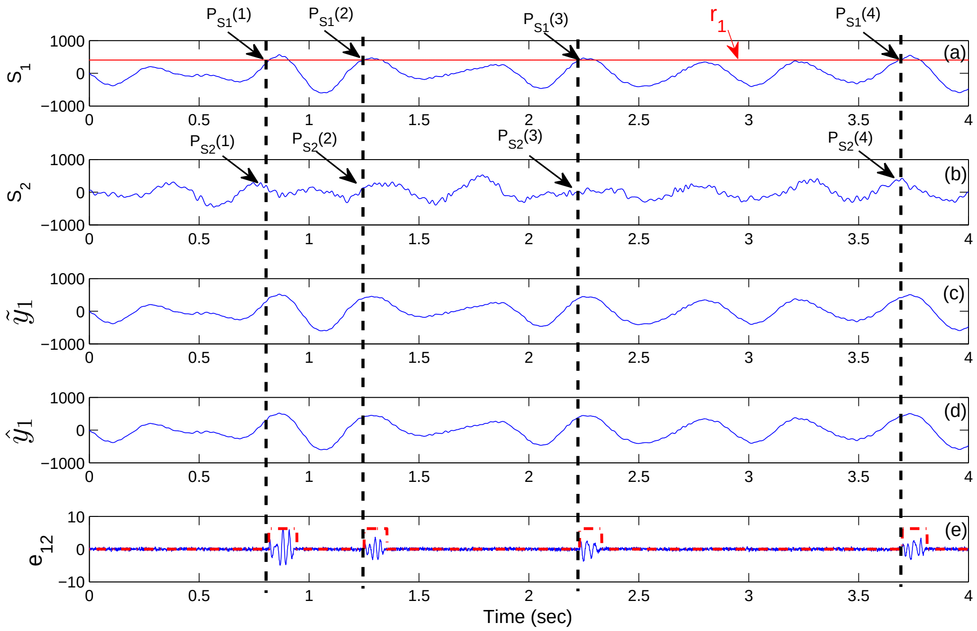


Figure 4.6: Abnormal signal detection and feature data point extraction: (a) true signal of S_1, y_1, (b) true signal of S_2, y_2, (c) measured signal from sensor 1, $\tilde{y}_1$, (d) the predicted sensor 1 output, $\hat{y}_1$, and the (e) cross-error function, e_{12} of the sensor pair.

are no reference sensors. Furthermore, as corrupted measurements should have values outside the normal region, the normal region boundaries can also be detected from the collected data. In fact, this classification problem can be modeled as a largest empty rectangle problem (LER) with a query point (*Kaplan and Sharir*, 2011; *Gutiérrez and Paramá*, 2012). Given a set of points and the boundaries in a 2-D space, the largest empty rectangle problem is to find the largest rectangle (with sides parallel to the axes) that does not contain any of the given points but contains the query point. Moreover, this rectangle should locate within the boundaries which can be set to be slightly larger than the maximum amplitudes of the given points. The non-linearity fault isolation and identification problem is equivalent to identifying the largest empty rectangle that contains the origin. The sides of the largest empty rectangle which intercept with the x-axis (y-axis) illustrate the parameters (*i.e.*, r_1

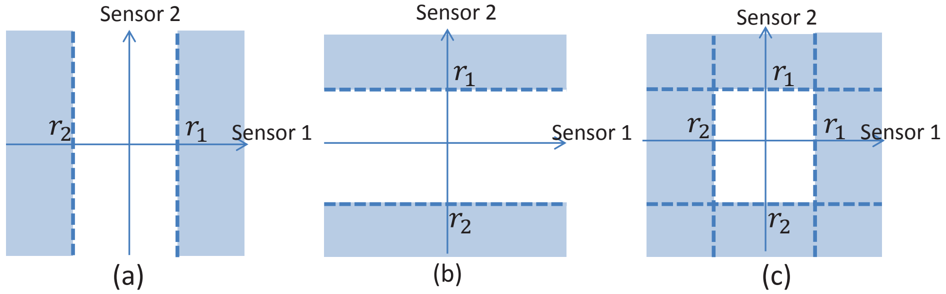


Figure 4.7: Location of extracted feature points when: a) $S1$ is faulty and $S2$ is normal, b) $S1$ is normal and $S2$ is faulty and c) both $S1$ and $S2$ are faulty. The dotted lines correspond to the boundaries between the normal and abnormal regions.

and r_2) of the non-linearity fault in sensor 1 (sensor 2). When a side is collocated with the boundary, it means no fault is detected on that region given the collected data; otherwise the coordinate of the side represents the range of the normal region of the non-linearity model.

4.2.2 Largest Empty Rectangle (LER) Problem

Finding the largest empty rectangle is an important problem in many applications including VLSI layout optimization (*Nandy et al.*, 1990) and database management (*Gutiérrez and Paramá*, 2012). Given its ubiquitous nature, many algorithms have been developed to solve this problem. However, even one of the faster algorithm (*Kaplan and Sharir*, 2011) requires $\mathcal{O}(N\phi(N)\log^4(N))$ operations, where N is the number of given points and $\phi(N)$ is the slowly increasing inverse Ackermann function (*Tarjan*, 1975). This requirement is quite demanding for low power wireless sensors when the size of the collected data is large. Therefore, a more efficient algorithm that findd an approximate largest empty rectangle containing the origin is proposed. The main concept of the algorithm is first to find a small rectangle that does not contain any given points and then to enlarge the rectangle by expanding the sides separately. The proposed method, which requires $\mathcal{O}(N)$ operations, is shown in Algorithm 1. In the algorithm, $\wp$ is the set of collected feature points, x_{max}, x_{min}, y_{max} and y_{min} define the boundaries and v_x (v_y) represents the x-coordinate (y-coordinate) of a point v.)

Algorithm 1 Approximate largest empty rectangle algorithm

```
Require: $\wp,\ x_{max},\ x_{min},\ y_{max},\ y_{min}$
  Calculate the shortest distance $d$ of the points in the set $\wp$ from the origin: $d = \min\{\sqrt{P_{S_1}^2 + P_{S_2}^2}|\mathbf{P} \in \wp\}$
  Construct an empty rectangle with sides: $x_{pos} = d,\ x_{neg} = -d,\ y_{pos} = d,\ y_{neg} = -d$
  Subroutine 1:
    $Q = \{v \in \wp | y_{pos} \geq v_y \geq y_{neg}\}$
      $x_{pos}^1 = min\{x_{max},\ \{v_x \in Q | v_x \geq 0\}\}$
      $x_{neg}^1 = max\{x_{min},\ \{v_x \in Q | v_x < 0\}\}$
    $Q = \{v \in \wp | x_{pos}^1 \geq v_x \geq x_{neg}^1\}$
      $y_{pos}^1 = min\{y_{max},\ \{v_y \in Q | v_y \geq 0\}\}$
      $y_{neg}^1 = max\{y_{min},\ \{v_y \in Q | v_y < 0\}\}$
    $A_1 = (x_{pos}^1 - x_{neg}^1) \times (y_{pos}^1 - y_{neg}^1)$
  Subroutine 2:
    $Q = \{v \in \wp | x_{pos} \geq v_x \geq x_{neg}\}$
      $y_{pos}^2 = min\{y_{max},\ \{v_y \in Q | v_y \geq 0\}\}$
      $y_{neg}^2 = max\{y_{min},\ \{v_y \in Q | v_y < 0\}\}$
    $Q = \{v \in \wp | y_{pos}^2 \geq v_y \geq y_{neg}^2\}$
      $x_{pos}^2 = min\{x_{max},\ \{v_x \in Q | v_x \geq 0\}\}$
      $x_{neg}^2 = max\{x_{min},\ \{v_x \in Q | v_x < 0\}\}$
    $A_2 = (x_{pos}^2 - x_{neg}^2) \times (y_{pos}^2 - y_{neg}^2)$
  if $A_1 \geq A_2$ then
     Output the largest empty rectangle with sides:
     $\{x_{pos}^1,\ x_{neg}^1,\ y_{pos}^1,\ y_{neg}^1\}$
  else
     Output the largest empty rectangle with sides:
     $\{x_{pos}^2,\ x_{neg}^2,\ y_{pos}^2,\ y_{neg}^2\}$
  end if
```

As can be easily verified, the rectangles calculated from subroutines 1 and 2 do not contain any feature points but do contain the origin. Each side of the rectangle either touches at least one data point or is collocated with one of the boundaries. The difference between subroutines 1 and 2 is to which direction (x- or y-directions) the rectangle expands first. The bigger rectangle is used for fault isolation and identification.

The assumption declared in the outset assumes that the normal region of the

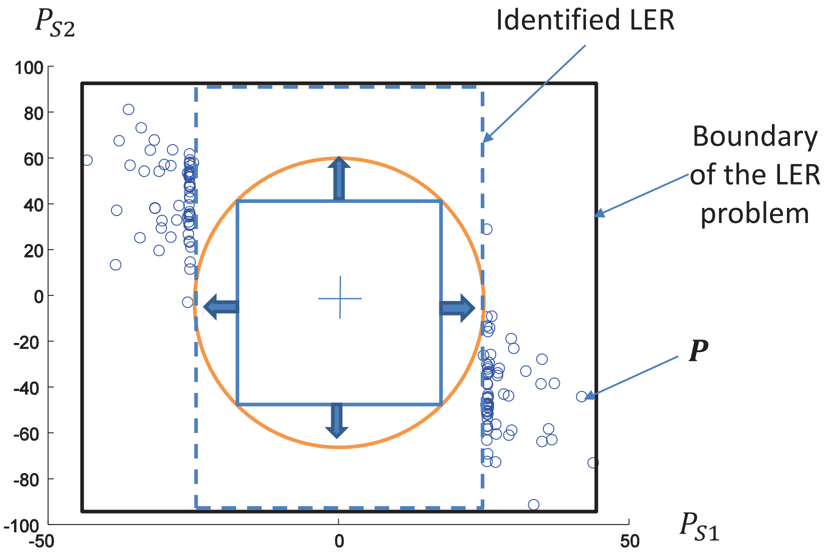


Figure 4.8: Illustration of the sub-optimal LER detection algorithm proposed to identify non-linear fault types in the two sensors.

faulty sensors covers a significant dynamic range of the signal. As a result, the area of the discovered largest empty rectangle should be comparable to the product of the maximum signal amplitudes of the two sensors (due to large r_1 and r_2). If the corruption of measurement is not due to a non-linearity fault, errors can occur at any signal amplitude and the largest empty rectangle found by the proposed algorithm would have a small area if the number of collected data points is large enough. An illustration of this property can be found in Section 4.3.3.

Fig. 4.8 shows an example of how the proposed largest empty rectangle algorithm identifies non-linearity faults. The data shown in the figure is extracted from a simulation experiment; the details of the experiment can be found in Section 4.3. In Fig. 4.8, the circular markers represent the set of feature points. The solid lines of the outer rectangle represent the boundaries of the largest empty rectangle problem. The dotted rectangle represents the final identified LER by Algorithm 1. As the top and bottom sides of the rectangle are collocated with the outer rectangle boundaries, there is no fault on $S1$. However, $S2$ is corrupted by non-linearity faults and the

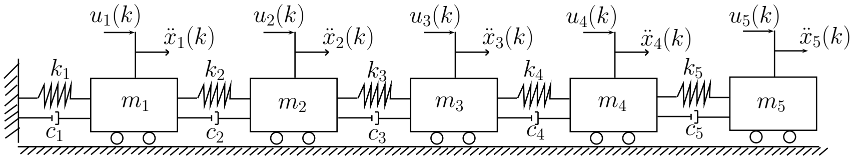


Figure 4.9: Five degree-of-freedom (DOF) spring-mass-damper system for validation of the non-linearity fault diagnosis method.

normal region of the fault is defined by the y-coordinates of the left and right sides of the rectangle.

4.3 Simulation and Results

In this section, the performance of the proposed algorithm is investigated including validation of its ability to detect and isolate sensors with non-linearity faults. The detection performance will be evaluated over different system excitations and under different non-linearity models. The diagnosis algorithm performance using the optimal LER and the proposed sub-optimal LER algorithm will be investigated. Also, the relationship between the detection accuracy and the number of collected feature points will be explored.

4.3.1 System for Validation

A 5-degree-of-freedom lumped mass dynamical system (Fig. 4.9) is adopted for simulation experiments. A lumped mass dynamical system is able to model different type of physical systems, such as bridges, vehicles, machines, among others (*Li et al.*, 2007). The masses, m_i, are connected via discrete springs and viscous dampers with spring constants, k_i, and damping coefficients, c_i, respectively. An external force, $u_i(k)$, is applied to each mass. Each mass also has a translational degree-of-freedom, x_i. In this study, similar model parameters used in (*Li et al.*, 2007) are adopted: each mass is set to $1kg$, and each spring constant and each damping coefficient are set to $10kN/m$ and $10.5Nsec/m$ respectively. Under these model parameters, the system has natural frequencies at: 4.52, 13.22, 20.84, 26.78, and 30.54 Hz. Each mode of vibration is under-damped with damping ratios of 1.5%, 4.4%, 6.8%, 8.8%, and 10.1% for mode 1 (4.53Hz) through 5 (30.54Hz), respectively. The vibration of the system is recorded as the acceleration of each degree-of-freedom ($y_i = \ddot{x}_i$) with a sampling rate of $200Hz$. The excitation inputs on each degree-of-freedom are from a single source $u(k)$ but with different levels of magnification: $u_1 = 5u,\ u_2 = 1u,\ u_3 = 0.2u,\ u_4 = -1.5u,\ u_5 = 12.2u,$.

4.3.2 Simulation Methodology

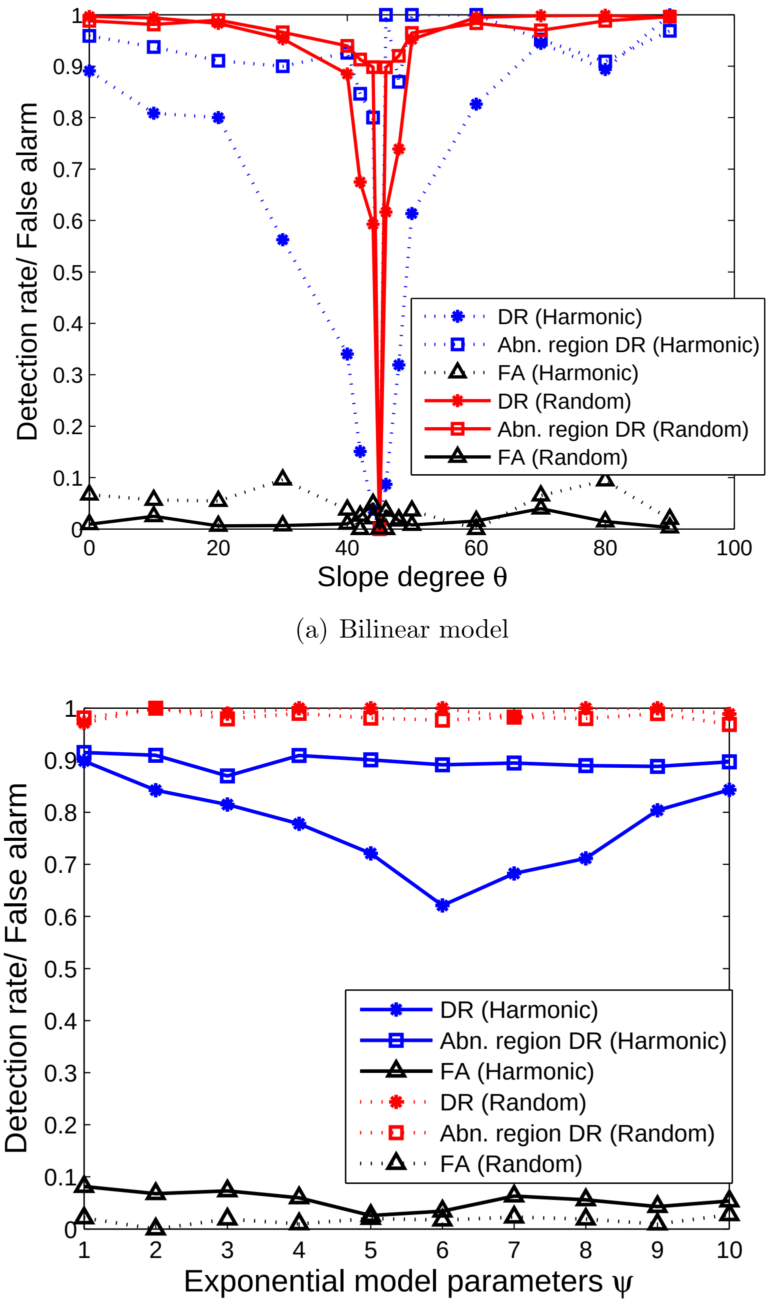
In this experiment, two types of excitation are used. The first excitation is a double-tone harmonic signal. The two frequencies of the signal are randomly chosen between 2 and 8 Hz using a uniform distribution. In a similar fashion, the magnitude of each tone is also randomly and uniformly chosen between 10 and $13N$. The second excitation is a white noise signal. When fitting ARX models between sensor pairs, the order of the ARX model is set to $\nu 20$.

For each simulation, whether a sensor is faulty or not is decided with equal probability ($p = 0.5$). For each simulation, it is decided at the outset what non-linearity fault model will be used (bilinear versus exponential). However, whether the non-linearity is one-sided on the positive side, one-sided on the negative side, or both-sided is determined randomly with equal probability assigned to each case. The ARX coefficients are trained by sensor outputs when independent white noise excitations are applied. During performance evaluation, a pair of sensors are randomly chosen and then the proposed algorithm for fault diagnosis is applied. A threshold on the cross-error function is used to determine whether there are discrepancies between a pair of sensors. In the following experiments, the threshold is set to the maximum amplitude of the cross-error function output when both sensors are functioning normally (or during the training of the ARX model). The proposed sub-optimal LER algorithm is used for all the experiments except when evaluating the differences in performance between the optimal and sub-optimal LER algorithms. For all the experiments, unless another noise value is explicitly stated, a zero-mean Gaussian white noise with variance of 0.5% of the maximum amplitude of the sensor measurements is superimposed to simulate measurement noise.

4.3.3 Simulation Results

In the first experiment, the detection performance of the proposed algorithm is evaluated over different parameters of the bilinear and exponential non-linearity models, respectively. Three performance criteria are evaluated. The first criterion is the percentage of faulty sensors that are correctly detected to be faulty (*i.e.*, detection rate and denoted as "DR" in the legends of the figures). The second criterion is the percentage of normal sensors falsely detected as faulty (*i.e.*, false alarm and denoted as "FA" in the legends of the figures). The third criterion is, among the correctly detected faulty sensors, the accuracy of detecting the abnormal region(s) of the non-linearity model. As the abnormal region can be one-sided or two-sided, we evaluate each side separately. For example, if an abnormal region appear on the positive side and the algorithm detects it correctly, a correct detection is recorded. This accuracy measure is denoted as abnormal region detection rate (*i.e.*, "Abn. region DR" in the legend of the result figures).

Fig. 4.10(a) shows the performance when a bilinear non-linearity model is used. The x-axis of the plot represents the degree θ of the slope in the abnormal region (note that a normal slope is 45°). As can be seen, the proposed algorithm achieved over a 90% detection rate and less than a 4% false alarm rate for most of the slope degrees when a white-noise excitation signal is used to excite the system. When a double-tone harmonic excitation is used, the detection rate is about 75% and the false alarm rate is about 7%. The difference in performance is because the sensor measurements

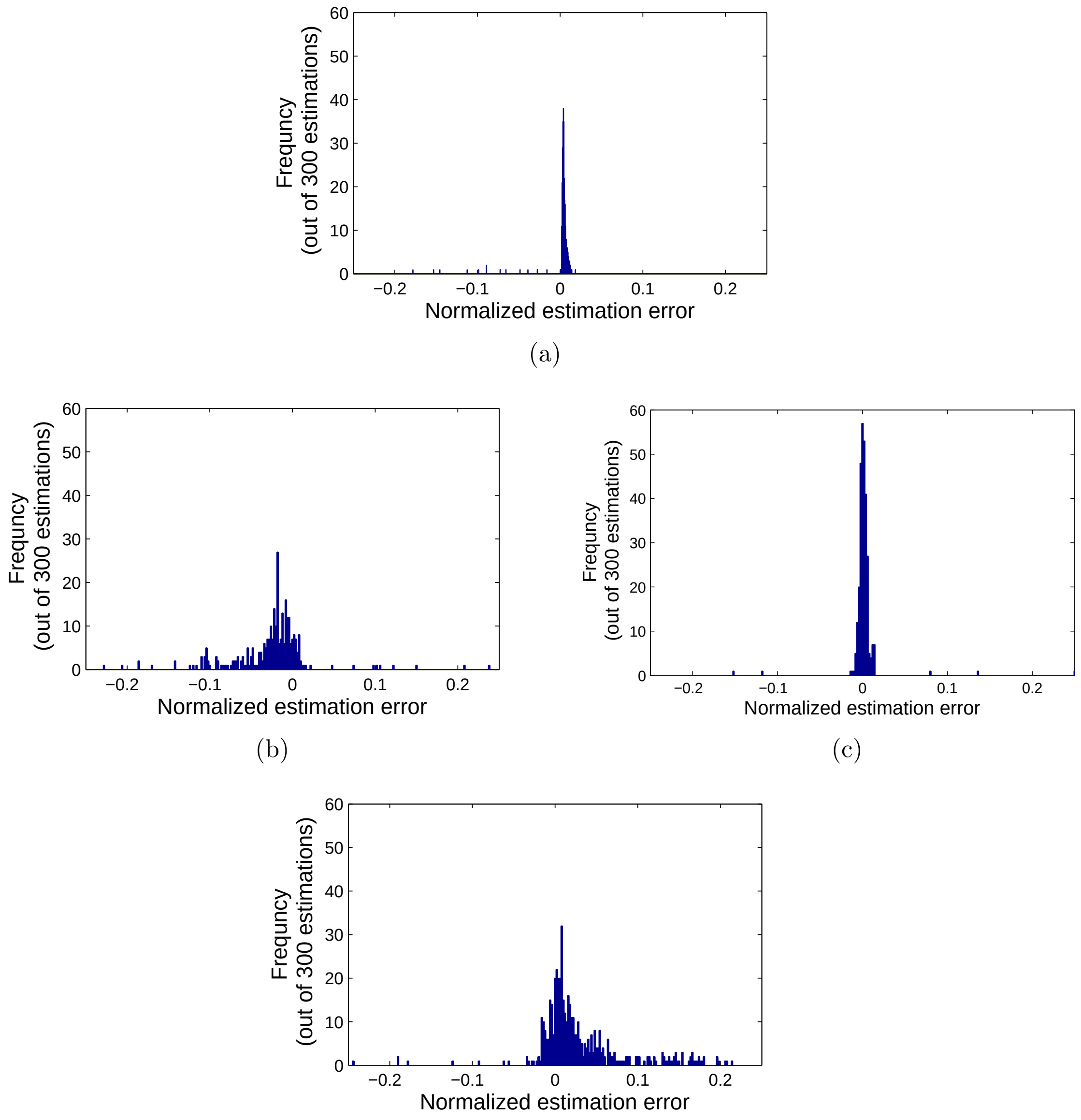


(a) Bilinear model

(b) Exponential model

Figure 4.10: Faulty sensor detection accuracy versus a) different slope degree θ (bilinear model) and b) variations in the exponential model parameter ψ.

of white-noise excited systems fall into the abnormal region rapidly. Consequently, the algorithm yields a more sensitive cross-error function and has better diagnosis accuracy. The detection rate decreases as the deviation is very close to the normal



(a)

(b)

(c)

(d)

Figure 4.11: Normalized normal region boundary detection accuracy: a) White noise excitation under bilinear model, b) harmonic excitation under bilinear model, c) White noise excitation under exponential model and d) harmonic excitation under exponential model

condition, (*i.e.*, 45°). This result is expected as the non-linearity error is also getting smaller and thus more difficult to detect from the cross-error function. The detection rate remains very high at greater than 85% when the slope deviates more than 10° from normal for white-noise excitations; this detection rate is greater than 65% for double-tone harmonic excitations. For detecting the abnormal type (*i.e.*, one-sided on the positive side, one-sided on the negative side or two-sided) of the non-linearity model, the algorithm is able to achieve an accuracy of 90% for every deviation degree and for both system excitations. (Notice for zero faulty sensor detection rate, the abnormal type detection rate is simply set to 0).

The same experiment is repeated using the exponential non-linearity model with the results shown in Fig. 4.10(b). Similar to the bilinear model, white-noise excitations achieve better results than double tone harmonic excitations. When exciting the system harmonically, detection rates of 95% or greater are found. However, the double tone harmonic has a detection rate between 60 and 90% depending on ψ. When white noise excited, the detection rate does not show a significant drop in performance when the non-linearity model function is close to the normal function (ψ_6). However, when excited by the harmonic signal, the detection rate increases as the exponential nonlinearity is more dominant. Note that the x-axis of Fig. 4.10(b) represents various exponential model parameters ψ but the distance between them is not related to the error magnitude.

Apart from detecting and isolating the faulty sensors, another important capability of the proposed method is its ability to detect the boundary values of the normal region. Fig. 4.11 shows the histogram of boundary estimation accuracy under the bilinear non-linearity model with parameter $\theta = 30°$ and white-noise excitation. (Fig. 4.11(a) represents the white-noise excitation under the bilinear model, Fig. 4.11(b) represents harmonic excitation under the bilinear model, Fig. 4.11(c) represents the white-noise excitation under the exponential model and Fig. 4.11(d) represents the harmonic excitation under the exponential model.) The estimation error is normalized (*i.e.*, the estimation error is divided by the actual boundary value). As can be seen, 95% of the estimations are within 2% of the actual boundary value for white-noise excitations; and 80% of the estimations are within 3% of the actual boundary value

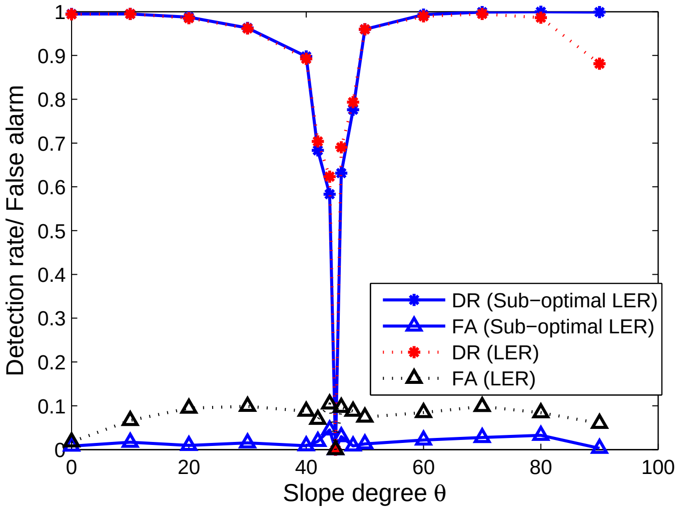


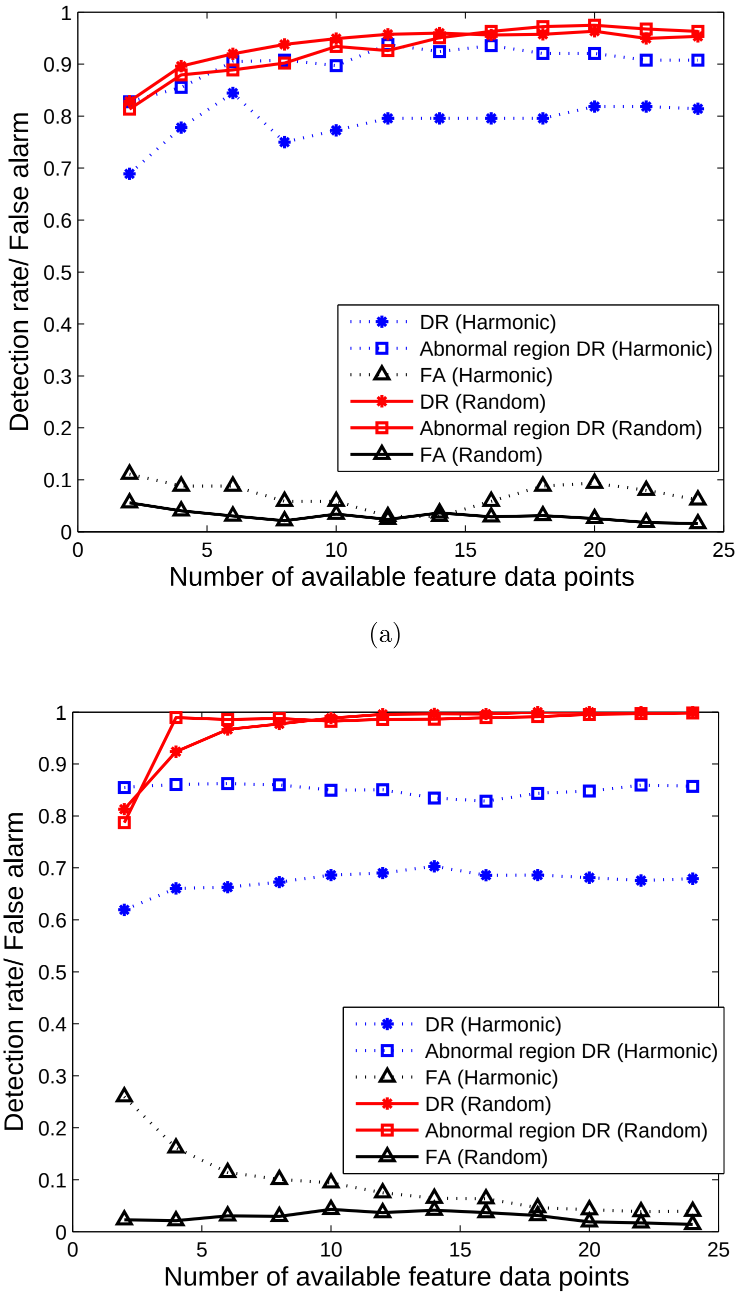
Figure 4.12: Comparison of detection performance between using the optimal LER method and the proposed low complexity LER method.

for double-tone excitations. Most of the boundary estimations (except the harmonic excitation under bilinear model case) are higher than the actual boundary value because the magnitude of the faulty measurements (thus the corresponding value of the selected feature points) are always larger than the true normal region boundary value. Therefore, the boundary values detected by the LER method are usually larger than the actual boundary values. This error can be reduced by using more feature points and avoiding using unnecessarily large thresholds on the cross-error function. For the harmonic excitation under the bilinear model (Fig. 4.11(c)), more than half of the estimations are smaller than the actual boundary values. This is mainly because the feature points collected from the less sensitive cross-error function have lower accuracy. As can be seen from the results, the proposed algorithm is able to achieve high accuracy on the boundary value estimation, which is important in signal recovery or for preventing the sensor from being used in the abnormal region in the future.

The proposed sub-optimal LER method is used in all of the previous experiments. This method has much lower complexity than the optimal LER method but it does not guarantee the largest empty rectangle. Therefore, a comparison is performed to reveal differences in the accuracy of these two methods. The experiment is performed under the bilinear non-linearity model with white-noise excitations. The results (Fig. 4.12) show that the proposed low complexity LER method is able achieve similar detection rates as the optimal LER method. Although the optimal method achieves a higher level of accuracy when the non-linearity error is small (*i.e.*, when θ is close to 45°), the optimal method generally has higher false alarm. Similar results are also observed for the exponential non-linearity model. Therefore, for detecting non-linearity faults, the proposed low complexity LER method is deemed sufficient.

The proposed algorithm diagnoses faulty sensors using the feature points extracted from the sensor outputs based on the cross-error function crossing a defined threshold. Only one such data point is collected every time the sensor output falls into an abnormal region. For systems whose signal changes slowly, it may take a long time to collect a large number of required data points. As a result, it is necessary to know the relationship between the number of feature points available and the corresponding detection accuracy, especially in identifying the fault pattern in Fig. 4.7 and estimating the boundaries of the normal regime of a faulty sensor. Two simulation are conducted using the bilinear fault model with θ set to 20° and the exponential model with ψ set to 0.1. Both double-harmonic and white noise excitations are evaluated. As shown in Fig. 4.13 and as expected, the detection rate and fault region detection rate increase when the number of available feature points increases. The detection rates plateaus when the available feature points reach 10 for the bilinear model and 6 for the exponential model. The result implies the proposed algorithm is able to detect sensors with non-linear faults using only a small number of feature points (*e.g.*, 10 or more).

Next, we evaluate the algorithm performance when measurement noise exists in the sensors. The simulation is conducted using the same bilinear fault model with θ set to 30° and the same exponential model with $\psi = 0.1$. Random white Gaussian

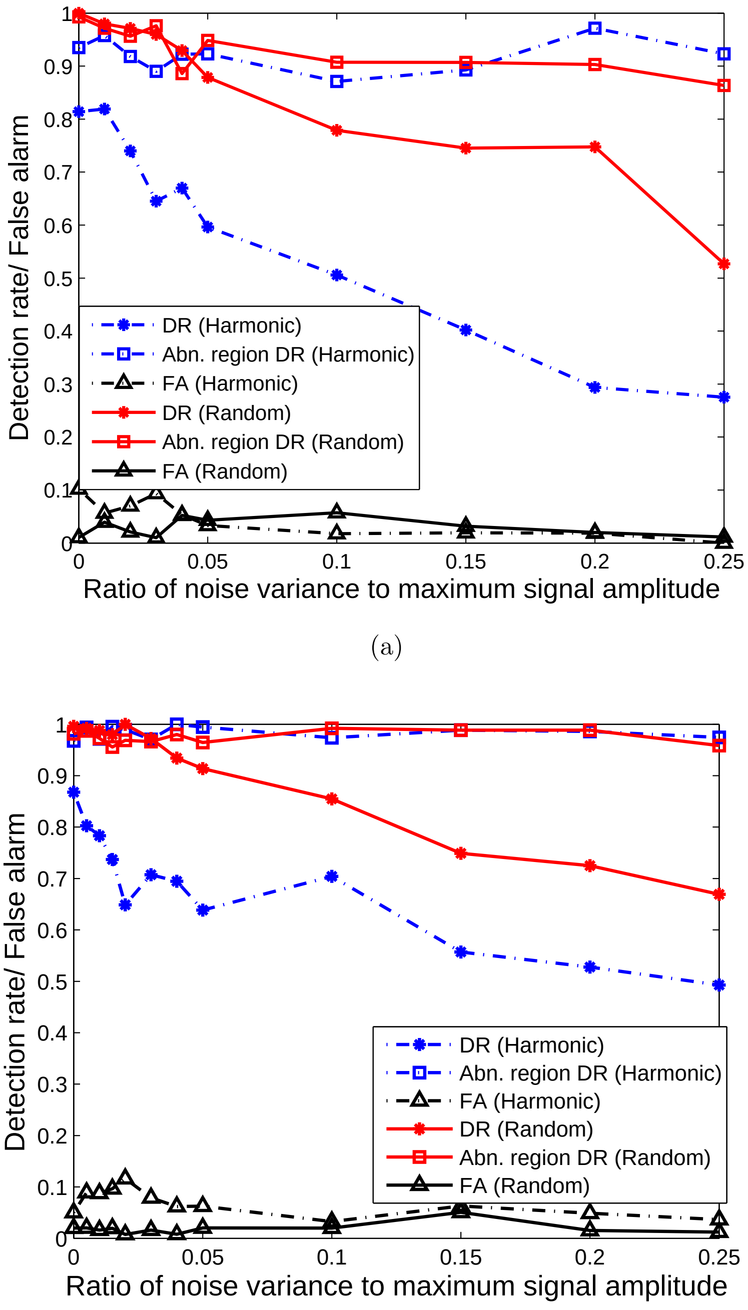


(a)

(b)

Figure 4.13: Detection accuracy versus number of available feature points when using a) bilinear model and b) exponential model.

noise with different noise variance is superimposed to the sensor measurements. The noise variance ratio in Fig. 4.14 is defined as the ratio of the noise variance to the maximum magnitude of the sensor measurements. As can be seen in Fig. 4.14,



(a)

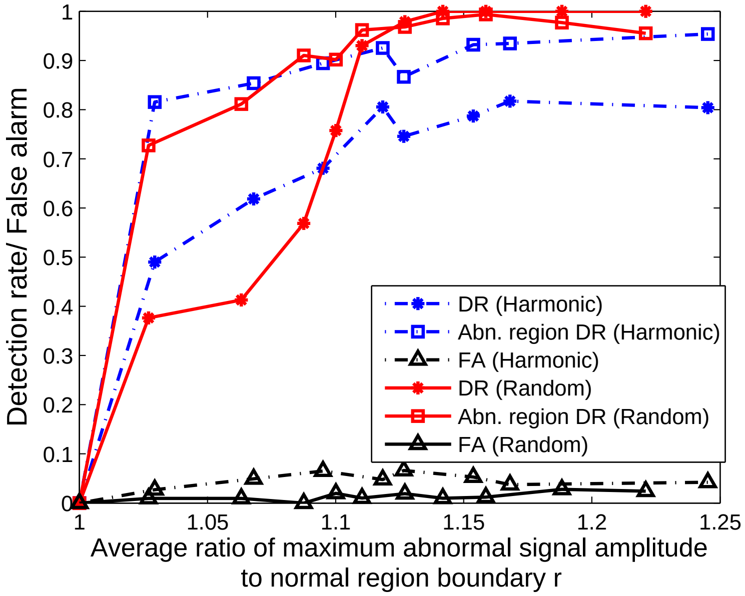
(b)

Figure 4.14: Detection accuracy versus sensor measurement noise when using a) bilinear model and b) exponential model.

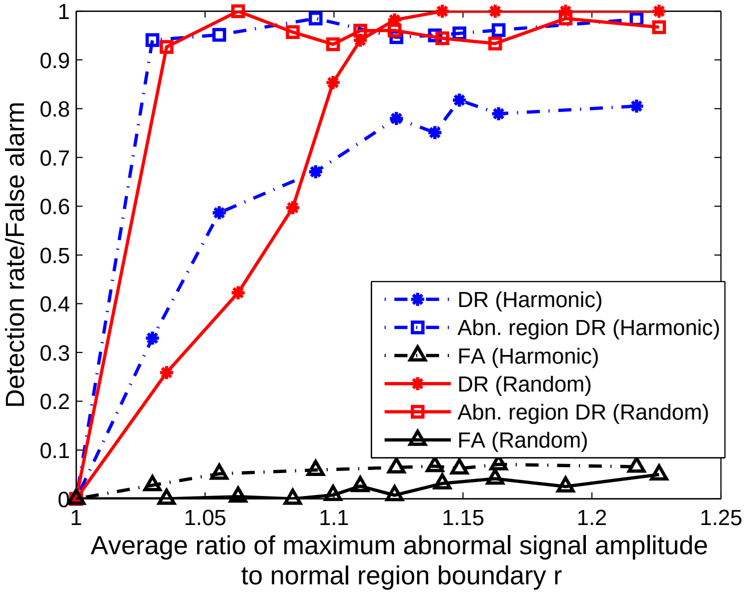
the detection rate decreases as the measurement noise variance increases in both non-linear models. For white-noise excited systems using the bilinear non-linearity model, the fault detection rate is 90% or greater when the noise variance is within 5% of the sensor measurement magnitude. Typical sensor noise is significantly less than 5% for common sensors. Similarly, white-noise excited systems using the exponential model achieved detection rates of 90% or greater under the same noise variance level. As was observed in the previous experiments, the accuracy of the algorithm on double-tone harmonic systems are slightly lower than that of white-noise excited systems. The detection rate on double-tone harmonic excited systems drops to 60% when the noise variance reaches 5% of the sensor measurement magnitude for bilinear models. For exponential non-linear models, the same detection rate drops to under 60% when the noise variance is larger than 15% of the sensor measurement magnitude. For the abnormal region detection of the correctly detected faulty sensors, the detection rate remains at about 90% of accuracy on different noise variances (for both excitation types and for both non-linear models). This implies the abnormal region detection is less affected by the measurement noise once the faulty sensor is correctly detected.

The influence of the maximum amplitude of the faulty signal relative to boundary, r, is evaluated. Fig. 4.15 shows the detection rate and false alarm rate as a function of the signal amplitude-normal region boundary (r) ratio for the bilinear and exponential models. The x-axis value is defined as the mean of the maximum sensor output each time the actual signal exceeds the normal region boundary divided by the normal region boundary. As shown in the figure, the fault detection accuracy increases in tandem with the average ratio reaches its maximum when the average ratio is 10% or higher.

As mentioned in Section 4.2.2, the proposed fault detection algorithm is able to distinguish non-linearity faults from other types of faults by examining the area of the detected largest empty rectangle. This property is illustrated by applying the LER algorithm to sensor data in which $S1$ is corrupted by spike faults, mean drift faults and excessive noise faults, respectively. These faults are common in sensors. Spike faults are sparse impulses superimposed on normal sensor measurements. Mean drift faults



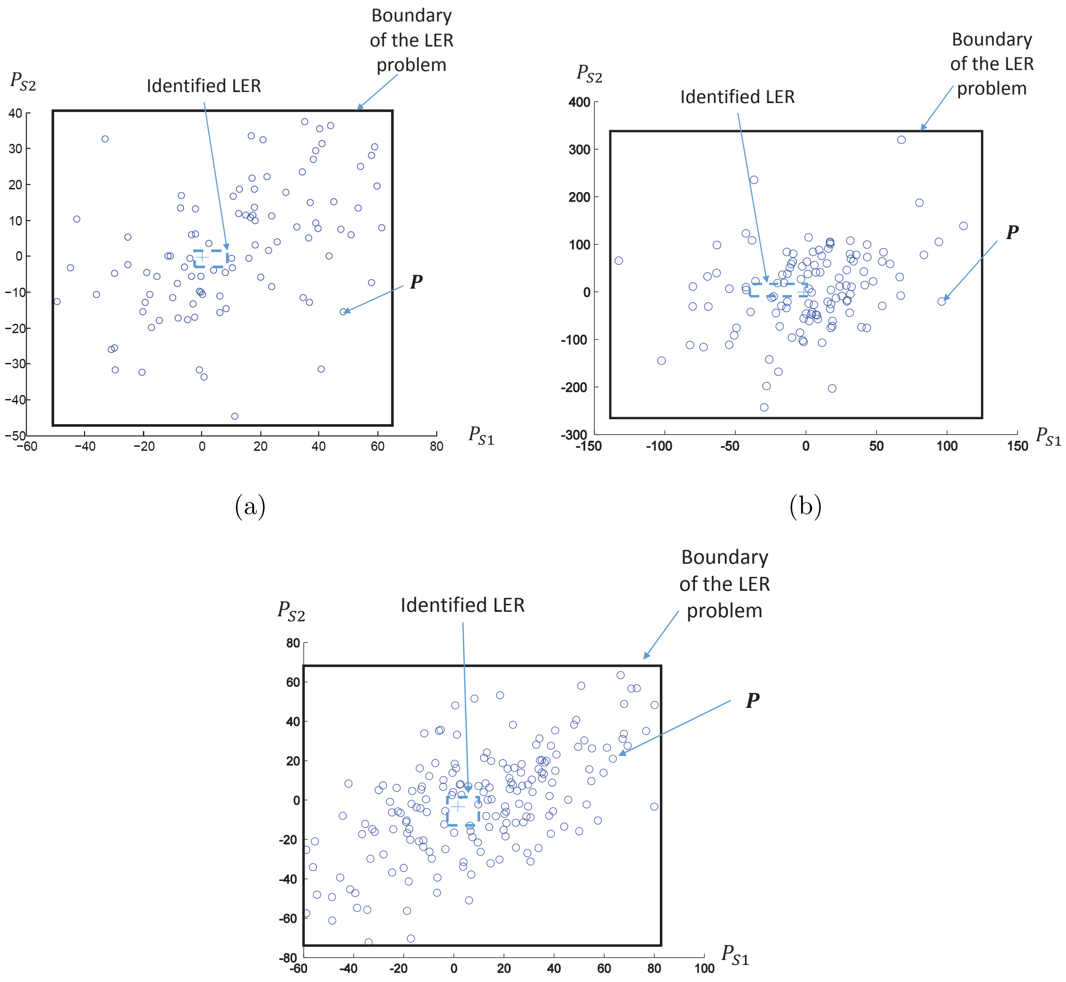
(a)



(b)

Figure 4.15: Detection accuracy versus abnormal signal amplitude when using: a) bilinear model and b) exponential model.

preserve the output dynamics but not its mean value. This type of faults generates outputs whose mean drifts away from the true mean of the signal slowly compared to the output dynamics. Excessive noise refers to a large amount of Gaussian noise



(a)

(b)

(c)

Figure 4.16: Illustration of the detected LER when one of the sensors is corrupted by: a) spike faults, b) mean-drift faults, and c) excessive noise faults.

in the output of a sensor. The results of the LER algorithm on the sensor data corrupted by these faults are shown in Fig. 4.16. All the detected LERs have very small areas (compared to the maximum amplitude of the sensor data). This is because the feature points caused by these faults do not always bigger than a fixed boundary. These feature points can locate in anywhere in the 2-D plane and thus do not follow any of the patterns in Fig. 4.7. As a result, non-linearity faults can be distinguished from other types of faults by examining the area of the largest empty rectangle.

4.4 Summary

In this Chapter, a model-based decentralized non-linearity fault detection and identification algorithm is proposed. The algorithm is carried out locally within a pair of sensors. Simulations show that the proposed method is able to identify non-linearity accurately. The algorithm generally achieves 90% or greater of detection rate on detection faulty sensors and obtains accurate values on detecting the boundaries of the normal region of the non-linearity models. A low-complexity sub-optimal LER algorithm is suggested and it has similar performance as the optimal LER algorithm in non-linearity faults detection. The proposed algorithm is able to distinguish the non-linearity faults from the other types of faults. Having the ability to identifying fault types is important as it provides information to the system to find out the fundamental cause of the sensor fault. Also, knowing the fault type helps to recover the corrupted data and thus reduces the sensor replacement cost.

CHAPTER V

Field Experiment on Grove Street Bridge

5.1 Introduction

In the previous chapters, distributed model-based fault detection and identification algorithms are proposed. These proposed algorithms are verified and evaluated through simulated and real data with faults. In this chapter, the performance of the distributed model-based fault detection and identification method is further investigated by a field study on the Grove Street Bridge located in Ypsilanti, Michigan. The spike detection and identification algorithm is implemented on 16 vibration sensing wireless sensors which are deployed on the bridge. Spike faults are generated on site and superimposed onto the true vibration signals before being sensed. Compared to the real spike corrupted data, this field experiment has true information on the spike time stamps and magnitudes. In addition to accuracy evaluation, this study also focuses on the relationship between the detection accuracy and the network partition methods. Based on this relationship, communication energy saving partition methods are presented. The main objectives of this study include: 1) evaluating the performance of the algorithm under real world environment, 2) testing the computational requirements of the suggested method on simple wireless sensors, 3) obtaining the relationship between the detection accuracy and the network partition methods and 4) comparing the energy consumption between the proposed distributed method and

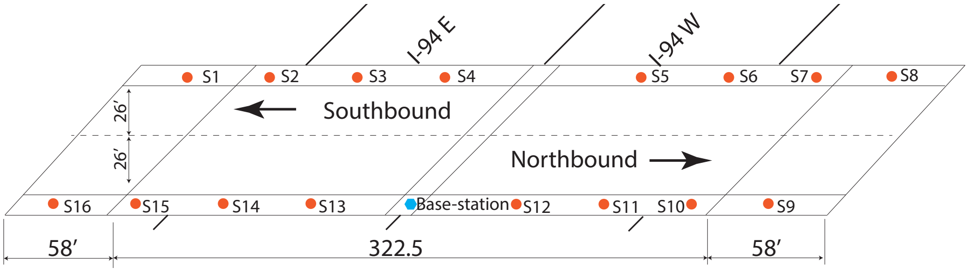


Figure 5.1: The Grove Street Bridge

generic centralized methods.

5.2 Performance Evaluation on Grove Street Bridge

5.2.1 Deployment Details

The proposed spike fault detection method is evaluated by implementing the detection algorithm on real vibration sensors and deploying them on the Grove Street Bridge located in Ypsilanti, Michigan. Spike faults are generated on site and superimposed on to the true signal before being sensed by the sensors. The Grove Street Bridge, as shown in Fig. 5.1, is a two-lane highway bridge over the interstate 94. This steel girder-concrete deck-composite-structured bridge is 455-feet long and 52-feet wide. Sixteen wireless sensors are deployed along the pedestrian walkways for vibration signal measurement as shown in Fig. 5.1.

The wireless sensors used in this experiment were developed by the University of Michigan. Each wireless sensor consists of a 8-bit low power controller (Atmel ATmega128), a ZigBee wireless communication unit, 4 channel inputs and 2 channel outputs. Each sensor is powered by 6 NiMH batteries. The vibration of the bridge is sensed by an accelerometer (Silicon Design model 2220) and then amplified by 10 times using a signal conditioning circuit. Finally, the amplified signal is sampled by a wireless sensor at 200Hz. For spike corrupted sensors, spikes are generated (by an extra wireless sensor unit) and superimposed to the amplified vibration signal before

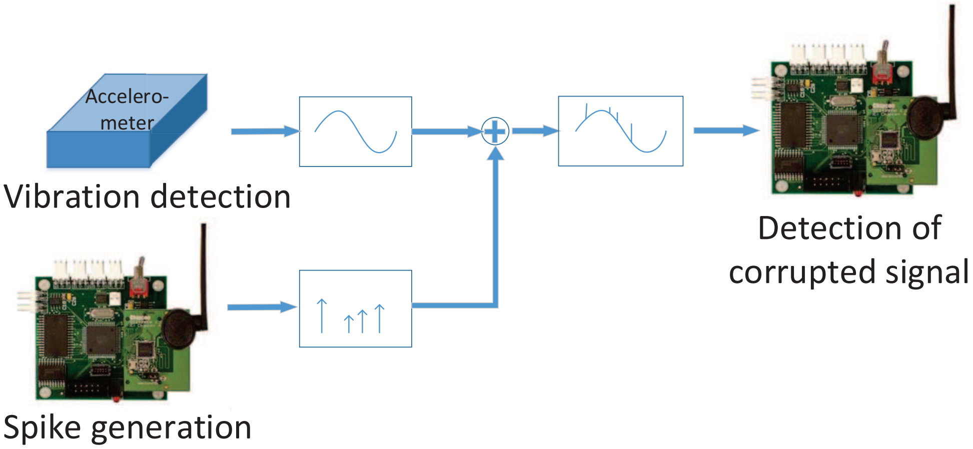


Figure 5.2: The flow diagram of spike generation

being sensed by a wireless sensor (Fig. 5.2). The occurrence of the spikes follows the Poisson Process. The time between two consecutive spikes (i.e., the inter-arrival time) follows an exponential distribution and the inter-arrival times are independent with each other. In this experiment, sensors $S2$, $S4$, $S11$ and $S13$ are designated to be the spike corrupted sensors. Various inter-arrival times and spikes amplitudes are used in the experiment.

In addition to the wireless sensors, a laptop computer is located in the middle of a pedestrian walkway and act as a base-station. This base-station is responsible for: 1) partitioning the network into sensor pairs, 2) initiating the ARX model identification task, 3) initiating the fault detection task and 4) collecting data and fault detection results from the wireless sensors.

5.2.2 ARX Model Training Method

The coefficients of the ARX model between sensor p and q can be identified by least square methods. Least square method requires the calculation of matrix multiplication and matrix inversion, the required computation complexity is $O(\nu^3 + \nu^2 M)$ (where ν is the order of the ARX model and M is the length of the data used for training) when Gauss-Jordan elimination method is used. This complexity is high for embedded deployment in wireless sensors.

The coefficients of the ARX model can be found by solving the following Yule-Walker equation (*Percival*, 1993):

$$[a_0, \ldots, a_{\nu_1}, b_0, \ldots, b_{\nu_2}] \begin{bmatrix} R_{y_p,y_p} & R_{y_p,y_q} \\ R_{y_q,y_p} & R_{y_q,y_q} \end{bmatrix} = [\delta, 0, \ldots, 0] \tag{5.1}$$

where the 2 × 2 matrix is a bi-Toeplitz matrix, and the four blocks of Toeplitz sub-matrices are the autocorrelation matrices or cross-correlation matrices of the outputs of sensors p and q.

By exploiting the structure of the bi-Toeplitz matrix in Eq. (5.1), *Monden et al.* (1982b); *Pan and Levine* (1990) propose fast coefficient identification methods for ARX coefficients training. In this experiment, Monden's method is adopted and implemented in the wireless sensors for ARX model training. This method iteratively calculates the ARX coefficients from order (1,0) to the desired order (ν, ν). The computational complexity is $O(\nu^2)$ and storage requirement is $O(\nu)$.

5.2.3 The Flow of the Experiments

The field experiments consist of two main processes: the model training process and the spike detection process. The model training process involve three steps: first, the base-station partitions the network into sensor pairs and requests the corresponding sensor pairs to identify their ARX coefficients; second, a master sensor of each sensor pair requests observation data from the slave sensor and starts the coefficient identification process immediately after the data is received; and third, the master sensor informs the completion of the identification process and send the ARX coefficients back to the base-station. The spike detection process also consists of three steps: first, the base-station requests a sensor pair to preform spike detection; second, the master sensor requests observed data from the slave sensor and performs

Table 5.1: The time requirements of different processes in fault detection on processing 12000 data points.

ARX order (2n)	Coefficient training	Cross-error function	Matched filter function
12	45.32s	8.90s	15.42s
18	62.44s	11.23s	21.04s
24	81.21s	14.57s	31.78s
30	96.11s	16.44s	39.60s

spike identification using the two matched filter functions (3.3 and 3.4); and third, the identification results are reported to the base-station. The second steps of both processes dominate the time consumption. The time required, under different model orders, for the coefficient training, cross-error function computation, and matched-filter function computation are shown in Table 5.1. The time requirement for data transmission is negligible when compared to the coefficient training process and fault detection process. For example, the transmission of 6000 data point from the slave sensor to the master sensor usually takes about 3 seconds. The transmission of ARX coefficients and spike detection results consumes even shorter of time.

5.3 Results

This section presents the performance of the proposed fault detection algorithm in the field experiment. The 16 sensors are deployed on the Grove Street Bridge for a few days. After the sensors being confirmed to be working properly, the ARX model of each sensor pairs are trained using 12000 data points (equivalent to 60s at 200Hz) and 18 coefficients (both coefficients **a** and **b** have size 9). After the ARX model training process is done, the spike fault detection performance is evaluated under different spike faults (different spike amplitudes and inter-arrival times) and different sensor pair combinations. The rest of this section first shows the method of detecting spikes from the matched filter result. Then, we presents the relationship between the

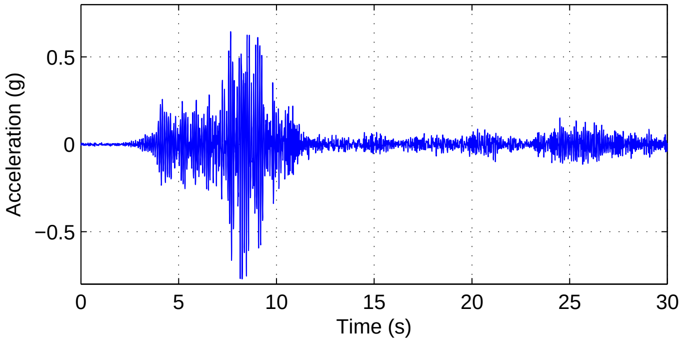


Figure 5.3: The observation of a normal sensor

performance of the algorithm and the similarity between coefficients **a** and **b**, and its influence on network partition. Finally, a three network partition methods are proposed, and the corresponding energy consumption is discussed.

5.3.1 Spike Detection using Matched Filters

The following shows the method used for detecting spikes from the sensor data. Fig. 5.3 shows typical vibration data being observed by a normal sensor. Typical spike corrupted data is shown in Fig. 5.4. If the coefficients of a ARX model have similar energy, i.e., $|\mathbf{a}|^2 \cong |\mathbf{b}|^2$, or the coefficients are highly uncorrelated, the spikes can be easily identified by first setting a threshold and then declared any data that the corresponding matched filter function output is larger than the pre-defined threshold as a spike (as shown in Chapter III). This is because a spike at sensor p will cause a high value in output of the matched filter **a** but a small value in output of the matched filter **b**. When the difference of the two coefficients' power is big, the higher power filter (say Matched filter **a**) could give high output even it does not match with the waveform (**b**) cause by a spike happened in the partner sensor (sensor q), and consequently might lead to wrong results. Most of the identified ARX models in this bridge experiment have imbalanced coefficient powers. As a result, the coefficients are normalized to have unit power before calculating the cross-error function and matched filter output.

The new cross-error function, $\dot{e}_{pq}$ is calculated by using the normalized coefficients $\dot{\mathbf{a}} = \frac{\mathbf{a}}{|\mathbf{a}|}$ and $\dot{\mathbf{b}} = \frac{\mathbf{b}}{|\mathbf{b}|}$:

$$\dot{e}_{pq}(k) = e_p(k) + \sum_{i=1}^{n} \dot{\mathbf{a}}_i e_p(k-i) - \sum_{i=0}^{n} \dot{\mathbf{b}}_i e_q(k-i) \tag{5.2}$$

and the corresponding matched filters are:

$$\begin{aligned} \dot{e}^{\mathbf{a}}_{pq}(k) &= |\sum_{i=-\infty}^{\infty} \dot{\mathbf{a}}_{i-k} \dot{e}_{pq}(i)| \qquad (5.3) \\ \dot{e}^{\mathbf{b}}_{pq}(k) &= |\sum_{i=-\infty}^{\infty} \dot{\mathbf{b}}_{i-k} \dot{e}_{pq}(i)| \qquad (5.4) \end{aligned}$$

The spikes are identified by the following criteria: when the value of a matched filter at time k, say $\dot{e}^{\mathbf{a}}_{pq}(k)$, exceeds a pre-defined threshold, a window (with length equals to the length of $\mathbf{a}$) of values that are next to time k is extracted from both matched filters, i.e., $\{\dot{e}^{\mathbf{a}}_{pq}(k-4), \ldots, \dot{e}^{\mathbf{a}}_{pq}(k+4)\}$ and $\{\dot{e}^{\mathbf{b}}_{pq}(k-4), \ldots, \dot{e}^{\mathbf{b}}_{pq}(k+4)\}$. The maximum magnitudes of both extracted vectors are compared, if $max\{\dot{e}^{\mathbf{a}}_{pq}(k-4), \ldots, \dot{e}^{\mathbf{a}}_{pq}(k+4)\} > max\{\dot{e}^{\mathbf{b}}_{pq}(k-4), \ldots, \dot{e}^{\mathbf{a}}_{pq}(k+4)\}$, the spike is regarded to be occurred in sensor p, otherwise the spike is regarded to be occurred in sensor q. The time k, where the highest value locates, indicates the time that the spike appeared in the sensor.

Note that the new cross-error function (5.2) is no longer a good indicator of whether a fault occurred within the sensor pairs. It is only for calculating the new matched filter functions ((5.3) and (5.3)). In the following of this section, the new cross-error function and matched filters are used.

Fig. 5.4 shows the data observed by two sensors ($S4$ and $S13$) within a sensor pair. Both sensors are spike corrupted and the corresponding cross-error function and matched filter functions are shown in Fig. 5.5. The spikes from both sensors appear in the cross-error function and become more apparent in the output of matched filters. The spikes occurred in $S4$ have higher amplitude in the matched filter $\mathbf{a}$ than that in the matched filter $\mathbf{b}$. Therefore, spikes occurred in sensor 4 can be distinguished from the spikes occurred in sensor 13. The detected sensor 4 spikes and sensor 13 spikes are indicated by square and star markers respectively in Fig. 5.4 and Fig. 5.5.

5.3.2 The Relationship between the Detection Accuracy and the Similarity between the ARX Coefficients

Without model errors and measurement errors, the highest output magnitude in the matched filter function, $\dot{e}^{\mathbf{a}}_{pq}$, which is caused by a spike with amplitude d in the sensor p, is $|\mathbf{a}|^2|d| = |d|$ (according to Eq. (5.2) and Eq. (5.3)). On the other hand, the highest output magnitude in the matched filter function, $\dot{e}^{\mathbf{b}}_{pq}$, which is caused by the same spike occurred in the sensor q, is $|\mathbf{a}||\mathbf{b}|C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})|d| = C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})|d|$, where $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$ is the highest cross-correlation measure between $\dot{\mathbf{a}}$ and $\dot{\mathbf{b}}$, (*i.e.*, the height output when a waveform $\mathbf{a}$ passes through the matched filter $\mathbf{b}$):

$$C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}}) = \max_{-n \le k \le n} \left| \sum_{m=0}^{n-1} \dot{\mathbf{a}}(m)\dot{\mathbf{b}}(m-k) \right| \tag{5.5}$$

Similarly, a spike with amplitude d occurred in sensor q has the highest output magnitude, $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})|d|$, in the output of the matched filter function, $\dot{e}^{\mathbf{a}}_{pq}$, and has the highest output magnitude $|d|$ in the output of the matched filter function, $\dot{e}^{\mathbf{b}}_{pq}$. Thereby, as long as $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}}) < 1$, the proposed spike detection criteria is able to identify which sensor contains the spike correctly. The measure $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$ equals to 1 only when $\dot{\mathbf{a}} = \dot{\mathbf{b}}$ (*i.e.*, $\mathbf{a}$ is proportional to $\mathbf{b}$).

However, when the ARX model is not ideal and/or measurement noise exists, $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$ needs to be smaller than 1 in order to achieve high detection accuracy because the output of the matched filters are now corrupted. In order to explore the relationship between the detection accuracy and the measure $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$, fault detection

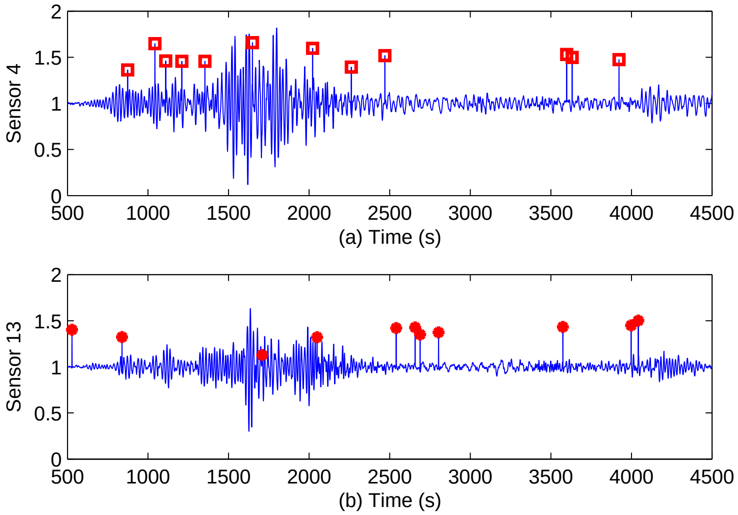


Figure 5.4: An example of spike corrupted data observed by sensor 4 and sensor 13 on the bridge. The detected sensor 4 spikes are indicated by rectangle markers and The detected sensor 12 spikes are indicated by star markers.

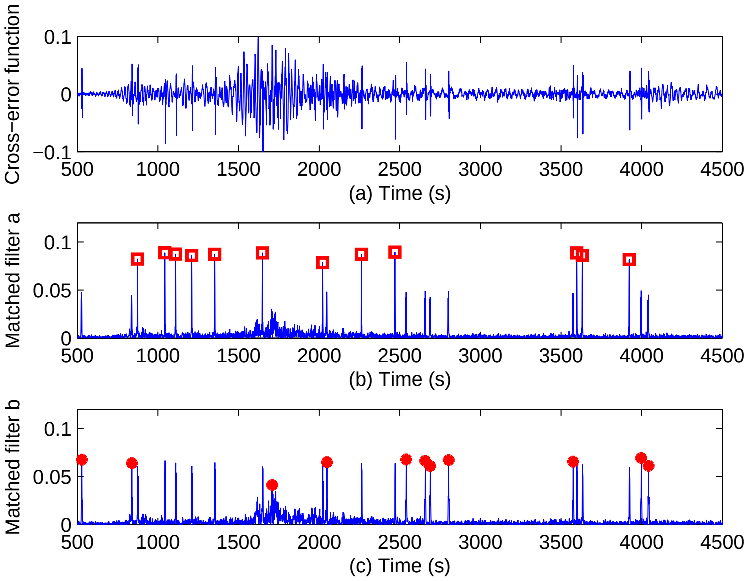


Figure 5.5: The output of cross-error function and matched filters. The detected sensor 4 spikes are indicated by rectangle markers and The detected sensor 12 spikes are indicated by star markers.

is performed on all sensor pair combinations that at least one of the sensors in the pair is faulty (*i.e.*, $S2, S4, S11$ and $S13$). The detection rate (DR) and false alarm

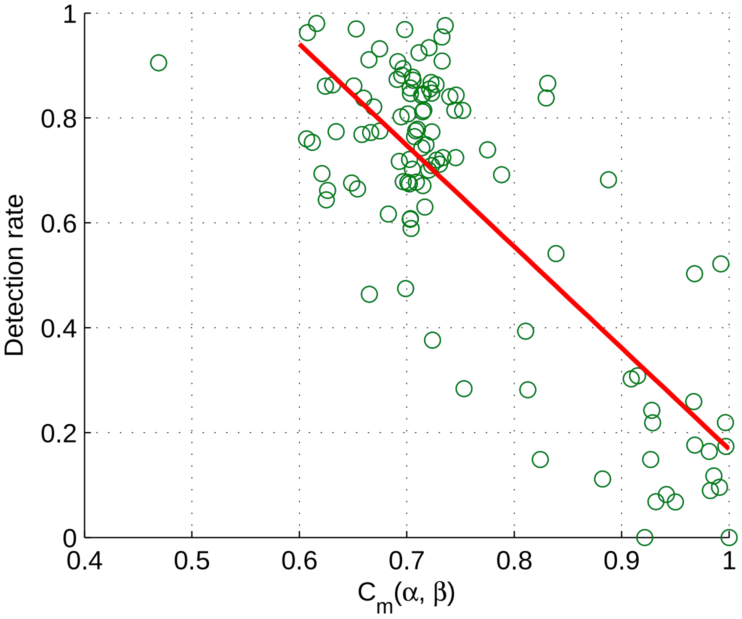


Figure 5.6: The relationship between detection rate and $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$.

rate (FA) of the spike detection within each pair of sensors and the corresponding similarity measure, $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$, are plotted on Fig. 5.6. A spike is correctly detected if the algorithm correctly determine the time and which sensor the spike was occurred. The detection rate is the number of correctly detected spike errors divided by the number of total spike errors. The false alarm rate is the number incorrect detection of spike errors divided by the number of normal data points. Figure 5.6 shows that the detection rate increases when the measure $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$ decreases. When $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}}) < 0.8$, most of the sensor pairs is able to achieve more than 70% of detection rate. By fitting the data with a order 1 polynomial, a linear relationship between the detection rate and the similarity measure is obtained and shown in Fig. 5.6.

The similarity measure $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$ gives the highest cross-correlation between $\dot{\mathbf{a}}$ and $\dot{\mathbf{b}}$. As $\dot{\mathbf{a}}$ and $\dot{\mathbf{b}}$ are the coefficients of the transfer function between the pair of sensors, $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$ is a measure of the similarity of the sensor observations. It is expected that sensors in close proximity have similar observations and a higher similarity measure, $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$, thus less desirable to be paired together. Figure 5.7 shows the similarity

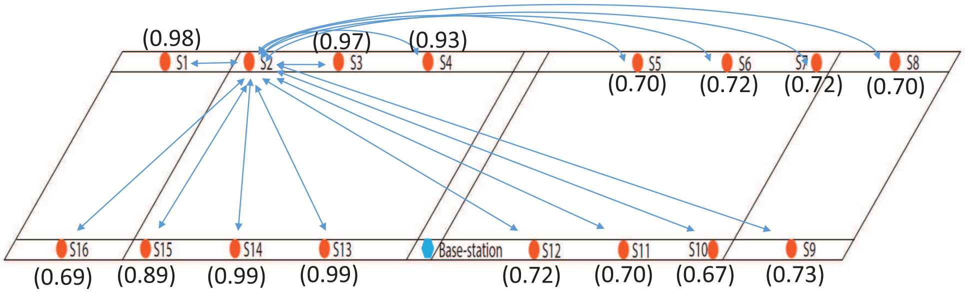


Figure 5.7: The measured $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$ between sensor 2 and the other sensors.

measures between sensor 2 and the other sensors. As can be seen, sensors that are located close to sensor 2 or located directly opposite to sensor 2 have a higher similarity measure. The similarity measure also depends on the structure of the bridge. Sensor 16 is close to sensor 2 but their similarity measure is not as high as other neighbor sensor pairs. This is because the surfaces that these two sensors are located are not belonging to a single structure. There is a gap/buffer between the two structures which allow expansion when the environmental temperature is high. Although vibration can transfer from one structure to the other, the buffer reduces the similarity measure, $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$.

Based on the relationship between the similarity measures and the corresponding spike detection rates, three network partition methods are proposed. The first method maximizes the detection rate without considering the communication energy. The second method reduces the communication energy while maintaining a high detection rate. This method first removes the communication links with high similarity measures and then constructs a network partition that minimizes the communication energy. The third method further reduces the communication cost of the second method by utilizing the broadcast of sensor data.

The first method partitions the network by choose the sensor pairs that have lowest $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$. This method maximizes the detection rate if the relationship between the similarity measures and the corresponding detection rates is assumed to be linear.

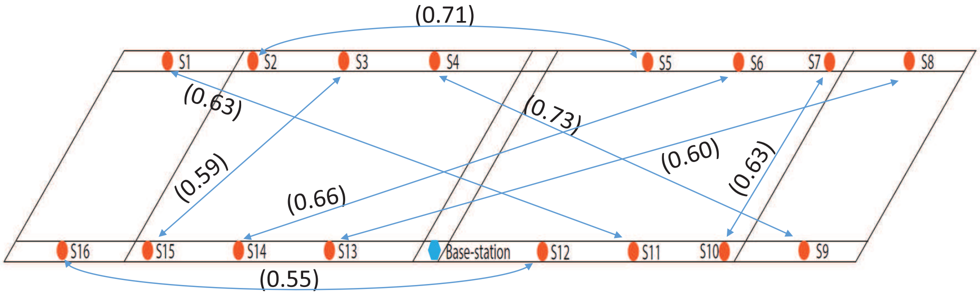


Figure 5.8: A minimal edge cover based on the $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$ measure.

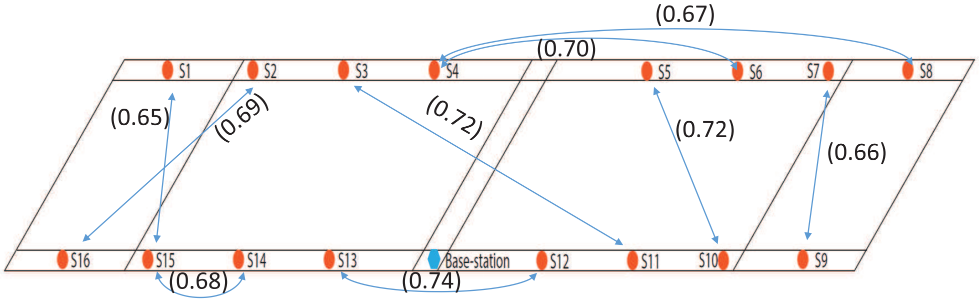


Figure 5.9: A minimal weight edge cover based on the distance squares.

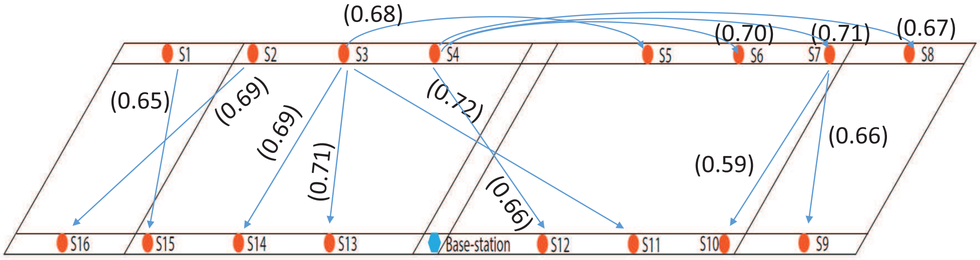


Figure 5.10: A minimal weight set cover based on the distance squares.

This partition can obtained by calculating the minimal weight edge cover (*Murty and Perin*, 1982) of the sensor network. An edge cover of a network is a set of edges that every sensor is connected to at least one other sensor. A minimal weight edge cover is an edge cover that the total weight of the edges is the smallest among all possible edge covers. Consider a network that for any two sensors in the sensor network, a link exist between them if they can communicate with each other directly. Also, the weight of the link is equal to the similarity measure, $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$, of the two sensors. Consequently, a minimal weight edge cover gives a good pair-wise partition of the network for the proposed sensor fault detection algorithm. A minimal weight edge cover based on this method is shown in Fig. 5.8.

The advantage of the first partition method is the high detection accuracy. However, sensors tend to pair up with far away sensors and thus resulted in high communication cost. To overcome this issue, the second partition method is introduced to minimize the communication energy consumption while maintains the high detection accuracy. First, the links that have high similarity measure are removed. Second, change the weight of the link to the square distance between the corresponding two sensors. Finally, calculate a minimal weight edge cover based on these new weights. The distances between the sensor pairs $\{S9, S10\}$ and $\{S15, S16\}$ are 33 inches, the distances between the sensor pairs $\{S10, S11\}$, $\{S11, S12\}$, $\{S13, S14\}$ and $\{S14, S15\}$ are 54 inches and the the distance between the sensor pairs $\{S12, S13\}$ is 108 inches. Sensor 1 to sensor 8 has the same distance profile as Sensor 16 to sensor 9, as shown in Fig. 5.1. The vertical separation between the two lines of sensors is 52 inches and inclination of the bridge surface is 60°. For example, if sensor 16 is defined to be the origin (0,0), then sensor 1 is located at coordinate (30, 52). The weight of the link is defined as the square distance between the two sensors because the communication energy is inversely proportional to the distance square. For example, when links with $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}}) < 0.75$ are removed and the minimal weight edge cover is calculated based on the square distance, a different edge cover is obtained (Fig. 5.9).

Compared to the first partition (Fig. 5.8), the sensor pairs in Fig. 5.9 have shorter separation in distance. To simplify the discussion, we assume communication energy is solely depending on the communication distance. Assume 1mJ of energy is required to transmit data for one second over a distance of 100 inches, then 4mJ of energy is required to transmit data for one second over a distance of 200 inches, according to the inverse square law of the propagation of EM waves. Based on these assumption, The power consumption of the first partition method (Fig. 5.8) is 39.4mW while the

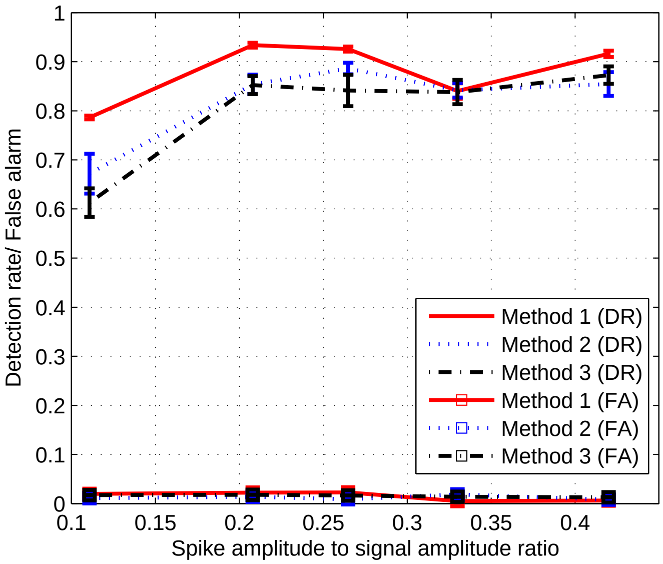


Figure 5.11: The detection rate (DR) and false alarm (FA) versus the spike amplitude of the three partition methods.

power consumption of the first partition method (Fig. 5.9) is 15.7mW, which is about 60% of saving.

Communication cost is further reduced if data is broadcasted from one sensor to multiple sensors. The third partition method constructs a network partition which minimizes the communication energy consumption when broadcasting is allowed. Similar to the second method, communication links with high similarity measures are first removed. Note that the broadcast range, or equivalently the number of sensors covered under broadcasting, increases with the transmission energy. With all possible broadcast ranges and the corresponding communication energy, the optimal network partition, which minimizes the total communication energy, is the minimum weight set cover (Fig. 5.10) of the network. As can be seen, only sensor 1, 2, 3, 4, and 7 transmit their data to other sensors. The total communication power consumption is 11.2mW, which is about 71% of the second method. The third method requires more sensors to perform the fault detection calculation. As a result, this method is preferred when the computational cost is lower than the communication cost.

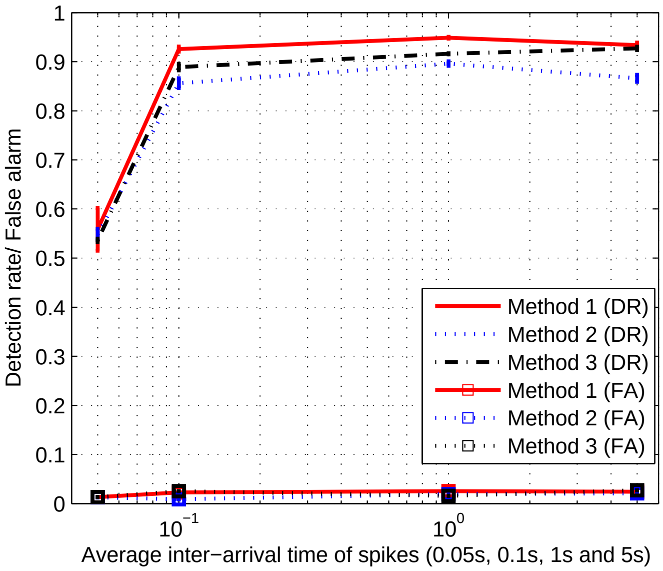


Figure 5.12: The detection rate (DR) and false alarm (FA) versus the spike inter-arrival time of the three partition methods.

The performance of the proposed method under the three network partition methods are evaluated with various spike amplitudes and average inter-arrival times. The results, which are presented in Fig. 5.11 and Fig. 5.12, are obtained from 2-3 sets of vibration data. Each set of data consists of 8-9 sensor pairs (depends on the partition method) and each sensor pair inspects 12000 data points for spike detection. Fig. 5.11 shows the result of the detection rate and false alarm versus the average spike amplitudes. The x-axis of the figure is the ratio of the spike amplitude to the maximum peak-to-peak amplitude of the observation signal. The average inter-arrival time of the spikes is set to 0.5 second. When the spike amplitude ratio is higher than 0.2, the first partition method achieves around 90% of detection rate, the second and the third partition methods achieves similar accuracy at around 85%. When the spike amplitude ratio smaller than 0.2, the detection rate starts to be affected by the observation noise and the error caused by the inaccurate ARX model. When the ratio is 0.1, the detection rate of the three partition methods drop $10\% - 15\%$ in detection rate. The false alarm rate is maintained at about 1% for the three partition methods under different spike amplitudes.

Fig. 5.12 shows the detection rate and false alarm rate versus different average inter-arrival times of the spike faults. The average amplitude ratio of the spike faults is set to 0.2. When the average inter-arrival time is larger than or equal to 0.05 second, which is equivalent to 10 data points, the detection rates of the three partition methods seem not being affected by changes of the inter-arrival times. When the average inter-arrival time is smaller than 0.05 second, the identification of a spike fault could be affected by the another nearby spike fault. This is because the proposed spike detection method (mentioned in Chapter 5.3.1) considers a window of outputs from both matched filter functions to determine which sensor contains the spike. As the low quality sensor pairs (high $C_m(\dot{\mathbf{a}}, \dot{\mathbf{b}})$) are eliminated, the performances of the proposed fault detection method using the second and third partition methods are only slightly lower than that using the first partition method (as shown in Fig. 5.11 and Fig. 5.12).

One may be interested to know how many energy is required if centralized fault detection method is used, i.e., all the sensor observations are sent back to the base-station and then perform fault detection by the base-station. For example, in out experiment, we put the base-station in the center of the bridge, which is beneficial to communication energy saving, has a power consumption of 34.5mW. In this example, the energy consumption is similar to the first partition method but 120% more than the second partition method, and about 200% more than the third partition method. The second and third partition methods are recommended because energy is usually scarce in wireless sensor network and their detection rates are only slightly lower than the cross-correlation based partition method.

To summarize, the accuracy of the proposed method depends on the similarity measures of the ARX coefficients. Sensors with low similarity should be paired together. Three network partition methods are introduced and evaluated. Results show that the communication energy can be reduced without sacrificing too much of the detection accuracy.

5.4 Summary

A field experiment is conducted on the Grove Street Bridge located in Michigan. This field experiment, together with the simulations presented in Chapter III, provided a throughout evaluation of the performance of the spike detection and identification algorithm. With the Yule-Walker equation based fast iterative ARX coefficient training method, the ARX training process can be done in a reasonable time by a 8-bit micro-controller (Atmel ATmega128) running at 8MHz. The accuracy of the algorithm is greatly depending on the similarity measure of the ARX model coefficients, especially for simple and regular structures such as bridges. When the similarity measure is low, the accuracy of the field experiment is similar (5% lower) to the simulation results presented in Chapter III. Network partition methods are suggested to reduce communication energy for fault detection and identification. This study also investigated the recovery of the true signal from the detected spike information. The recovery based on the ARX coefficients trained by the wireless sensors does not always give correct results. This is because the sensors only support single precision calculation and the trained ARX coefficients do not have enough high accuracy as required by the recovery process. The recovery is accurate if the ARX coefficients are trained under double precision calculation. This limitation, however, will soon be eliminated by the continuous advancement of micro-controllers.

CHAPTER VI

Efficient Sensor Fault Detection Using Combinatorial Group Testing

6.1 Introduction

The previous chapters, efficient distributed sensor fault detection and identification methods are proposed. These algorithms reduces communication energy by carrying out the algorithm locally. In this chapter, efficient sensor fault detection algorithms that use fewer number of tests are introduced.

Most of the existing sensor fault detection methods, including the distributed model-based sensor fault detection algorithm proposed in Chapter II, require the number of tests at least on the order of the size of the network, i.e., $\mathcal{O}(N)$ tests are required, where N is the number of sensors in the network. Some methods even need $\mathcal{O}(mN)$ (where m is the number of neighbors of a sensor) or $\mathcal{O}(N^2)$ tests. A summary of the detection complexity is given in Table 6.1. For applications using an extremely large number of sensors (*Cho and Chandrakasan*, 2001), running a fault detection algorithm can involve a large amount of resources and cause significant delay.

We observe that while certain regional effects or catastrophic failure may result in a large number of faulty sensors at the same time, in the absence of such systemic problems and during normal operation faults occur randomly and sporadically. In

Method type	Complexity	Condition needed
Model-based:		
Kobayashi and Simon (2003)	$\mathcal{O}(N)$	At most one faulty sensor
Da and Lin (1995)	$\mathcal{O}(N)$	At most one faulty sensor
Li et al. (2007)	$\mathcal{O}(N)$	Reference sensor
Ricquebourg et al. (1991)	$\mathcal{O}(N)$	
Xu et al. (1999)	$\mathcal{O}(N)$	
Model-less:		
Ding et al. (2005)	$\mathcal{O}(mN)$	$m = \#$ of neighbors
Chen et al. (2006)	$\mathcal{O}(mN)$	$m = \#$ of neighbors
Koushanfar et al. (2003)	$\mathcal{O}(N^2)$	
Blough et al. (1989)	$\mathcal{O}(N \log N)$	

Table 6.1: Summary of existing methods

this case, the effort used for performing the traditional lengthy sensor fault detection seems not proportional to the number of faulty sensors to be found. This motivates us to seek lower complexity fault detection methods when faults may be rare and sparse.

Toward this end, we introduce a novel use of group testing techniques combined with Kalman filtering in detecting faulty sensors in a network. Assuming that the underlying system being monitored may be represented in a linear dynamical system framework and that sensor faults are relatively rare, our goal is to reduce the number of required tests given requirements on detection and false positive probabilities. There have been a few studies on using group testing to detect malfunctioning sensors; they generally differ in the testing/detection methods. For instance, *Goodrich and Hirschberg* (2006) propose a group testing based algorithm for detecting failure (dead) sensors. This algorithm evaluates a group of sensors by counting the number of responses from the group to a broadcast query (thus only applicable to sensor failure detection rather than fault detection). *Tosic and Frossard* (2012) propose a distributive sensor fault detection algorithm that measures a smooth phenomena (which implies neighboring sensors have similar measurements), where a group test is preformed by an unspecified dissimilarity comparison of neighboring sensors' measurements. Our work differs from the former in that we focus on detecting faulty sensors which are still responsive to queries, and differs from the latter in that we do not assume that sensors are highly correlated or that neighboring sensors have similar measurements.

Our approach consists of the following two components: the selection of a test group (also referred to as a test pool), and a Kalman filtering based testing/detection procedure over this group of sensors, which determines whether there exists at least one faulty sensor in this group. These two steps are repeated till desired performance criteria have been achieved. There are in general two ways of selecting the test groups. The first is open-loop, whereby the entire set of test groups are selected prior to performing any tests (this is done randomly in our study); this will be referred to as the combinatorial group testing (CGT) method. The second is closed-loop, whereby each test group is selected adaptively based on outcomes of previous tests (this adaptive section is done using standard criteria like uncertainty reduction maximization in our study); this will be referred to as the Bayesian group testing (BGT) method. The CGT method is presented in this chapter and the BGT method is presented in the next chapter. The next chapter also further consider the detection performance of Kalman filtering, and use such understanding in determining the selection of test groups under the Bayesian group testing method; this will be referred to as the Kalman filtering-enhanced Bayesian group testing method (KF-BGT). It should be emphasized that under all these methods the group tests (the second component) themselves are performed via Kalman filtering; they simply differ in how the test groups are selected (the first component).

The remainder of this chapter is organized as follows: first, the main concepts used in the group testing-based fault detection algorithm is reviewed. Second, the detailed methodology of the detection algorithm based on CGT is explained. Third, the experimental set up and the nature of a set of bridge vibration data we use for numerical evaluation are described. Finally, The performance of the CGT methods on the bridge vibration data is presented.

6.2 Preliminaries

In this section we review two main concepts used in our fault detection algorithm. The first is group testing, the goal of which is to identify sparse faulty items with a number of tests less than the total number of items. The second concept is Kalman filtering, which is able to produce optimal state estimation for a linear dynamical system.

6.2.1 Group Testing

Consider a large number of items of which a few are defective, and we wish to identify them. If each item is tested individually, the cost can be high (linear in the total number of items). However, if it is possible to determine the existence of any defective item in a group of items via a single *group test*, then performing a sequence of group tests over different subsets of these items can potentially lead to much fewer number of tests and thus much lower cost. This is the main idea of group testing; it was first proposed by Dorfman (*Dorfman*, 1943) during World War II for detecting syphilis amongst soldiers.

Consider a length N signal S which is d sparse: this means S has at most d non-zero entries that correspond to the defective items and $d \ll N$. As the "true" signal dimension (i.e., d) is smaller than N, it is conceivable that signal S can be acquired with $M < N$ measurements. In group testing paradigm, signal S is measured M times in the form of $W = \Phi S$, where Φ is the measurement matrix of size $M \times N$. The arithmetic is boolean, meaning that the multiplication is logical AND and addition logical OR. If these operations are noisy, then the group test results are given by Z rather than W, with $P(Z_i = 1|W_i = 0) = \alpha$ and $P(Z_i = 0|W_i = 1) = \beta$, $\forall i$, denoting the two types of errors. The goal of group testing is to design Φ such that S can be reconstructed from Z (i.e., we can find the d defective items) with sufficiently low error probabilities.

We now describe this in the context of a network of N sensors, of which at most d are faulty. Let vector S represent the true *fault state* of the sensors in the network, where $S_i = 0$ if sensor i is normal and $S_i = 1$ if sensor i is faulty. Each row of the matrix Φ, which has $\{0, 1\}$ entries, represents the set of sensors involved in a test. A row of Φ is called a test group/pool and the number of rows equals the number of tests. Finally, the vector Z represents the result of the group tests. Below is a toy example of $\Phi S = Z$:

Example VI.1.

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

In this example, there are 6 sensors; sensor 2 is faulty. A total of 3 group tests are performed: sensors {2, 5, 6} are included in the first test (first row of Φ), and so on. The test result shows correctly that the first group contains at least one faulty sensor and the second group has none, but declares incorrectly that the third group contains a faulty sensor. In a fault detection setting, S is unknown while Φ is known by design and Z is known by observing the test results. S and Z are then used to reconstruct S. As mentioned in the introduction, group-testing a set of sensors in our context is far more complicated than a simple boolean operator, noise-free or noisy. To use this group testing framework in practice, we must specify what a “group test” entails, and how to actually obtain values in the Z vector. This is addressed by a novel use of Kalman filtering detailed next.

6.2.2 Kalman Filter Based Group Test

The Kalman filter (*Maybeck*, 1979) is an algorithm which takes a series of noisy inputs and iteratively calculates a statistically optimal estimate of the state of an underlying linear dynamical system. More specifically, consider a linear dynamical system given by the following state-space model (*Maybeck*, 1979):

$$\mathbf{X}(k+1) = \mathbf{A}\mathbf{X}(k) + \mathbf{B}\mathbf{U}(k) + \mathbf{G}(k) \tag{6.1}$$

$$\mathbf{Y}(k) = \mathbf{C}\mathbf{X}(k) + \mathbf{V}(k) \ . \tag{6.2}$$

where the first equation represents the dynamics of the system while the second represents the (sensor) observation model. Here $\mathbf{X}(k) \in \mathbb{R}^n$ is the state vector of the system, $\mathbf{U}(k) \in \mathbb{R}^l$ the input (or control) vector, and $\mathbf{Y}(k) \in \mathbb{R}^N$ the output vector of sensors. Matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are determined by the physics of the system as well as the sensors. $\mathbf{G}$ and $\mathbf{V}$ are Gaussian white noise with zero mean and covariance matrices $\mathbf{R_G}$ and $\mathbf{R_V}$, respectively. $\mathbf{X}(0)$, $\mathbf{G}(k)$ and $\mathbf{V}(k)$ are assumed to be independent. Assume the noises, $\mathbf{G}$ and $\mathbf{V}$, are small, the next system state, $\mathbf{X}(k+1)$, is mainly depends on the current system state, $\mathbf{X}(k)$, and the current input $\mathbf{U}(k)$. Also, the current output of the sensors, $\mathbf{Y}(k)$, is mainly depends on the current system state $\mathbf{X}(k)$.

The Kalman filter state estimation can be separated into two steps, a prediction step and an update step. In the prediction step, the predicted state (of time k based on the value at time $k-1$), $\hat{\mathbf{X}}(k|k-1)$ and the corresponding uncertainty measure of the prediction, $\mathbf{P}(k|k-1)$ are calculated:

$$\hat{\mathbf{X}}(k|k-1) = \mathbf{A}\hat{\mathbf{X}}(k-1|k-1) + \mathbf{B}\mathbf{U}(k) \tag{6.3}$$

$$\mathbf{P}(k|k-1) = \mathbf{A}\mathbf{P}(k-1|k-1)\mathbf{A}^{\mathrm{T}} + \mathbf{R_W} \ , \tag{6.4}$$

Upon observing a measurement $\mathbf{Y}(k)$, the estimated state and uncertainty measure are updated as follows:

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{C}^{\mathrm{T}}(\mathbf{C}\mathbf{P}(k|k-1)\mathbf{C}^{\mathrm{T}} + \mathbf{R})^{-1} \tag{6.5}$$

$$\hat{\mathbf{X}}(k|k) = \hat{\mathbf{X}}(k|k-1) + \mathbf{K}(k)(\mathbf{Y}(k) - \mathbf{C}\hat{\mathbf{X}}(k|k-1)) \tag{6.6}$$

$$\mathbf{P}(k|k) = (\mathbf{I} - \mathbf{K}(k)\mathbf{C})\mathbf{P}(k|k-1) \ . \tag{6.7}$$

where the updated state, $\hat{\mathbf{X}}(k|k)$, is a weighted sum of the estimated state and the innovation $(\mathbf{Y}(k) - \mathbf{C}\hat{\mathbf{X}}(k|k-1))$. The weight depends on the uncertainty measure $\mathbf{P}(k|k-1)$: the more uncertain the estimated state is, the more weight is placed on the new observation.

The group testing method requires the fault detection method to identify whether an arbitrary group of sensors contains any faulty member. The idea of using Kalman filtering for group testing lies in its ability to estimate the state of the underlying system from the observations of almost arbitrary group of sensors. For example, if one wants to estimate the system state by using the outputs from sensors 1,3 and 4, the observation model (Eq. (6.2)) can be changed to $\mathbf{Y}'(k) = \mathbf{C}'\mathbf{X}(k) + \mathbf{V}'(k)$, where $\mathbf{Y}'(k)$ contains only the 1^{st} and 3^{rd} components of $\mathbf{Y}(k)$, $\mathbf{C}'$ contains the 1^{st} and 3^{rd} rows of $\mathbf{C}$ and $\mathbf{V}'(k)$ contains the 1^{st} and 3^{rd} components of $\mathbf{V}(k)$. The dynamic equation of the system (Eq. (6.1)) remains the same. Specifically, after selecting an arbitrary group of sensors ϕ, we will further split this set into two subgroups A and B, and use the observations from each subset to estimate the state of the underlying

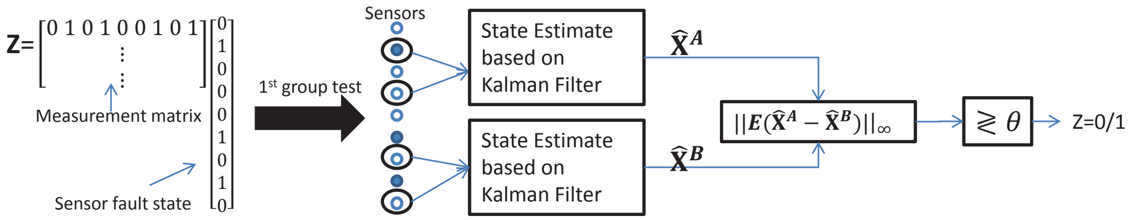


Figure 6.1: State diagram of the proposed sensor fault detection method.

system (thus it is required that a group contain at least two sensors). If the estimated states do not agree with each other, the group is regarded as containing at least one faulty sensor and the corresponding entry of Z is set to 1.

Denote the estimated states of the system, computed from observations of the subgroups A and B, as $\hat{\mathbf{X}}^A(k|k-1)$ and $\hat{\mathbf{X}}^B(k|k-1)$, respectively. The difference between the two estimated states is given by:

$$\mathbf{e}(k) = \hat{\mathbf{X}}^A(k|k-1) - \hat{\mathbf{X}}^B(k|k-1) \ . \tag{6.8}$$

As all states estimated from the Kalman filter are unbiased (i.e., $E[\hat{\mathbf{X}}(k|k-1)] = \mathbf{X}(k)$) (*Maybeck*, 1979), the expected difference $E[\mathbf{e}(k)] = E[\hat{\mathbf{X}}^A(k|k-1)] - E[\hat{\mathbf{X}}^B(k|k-1)] = 0$ if neither A nor B contains any faulty sensor (i.e., the corresponding components in $\mathbf{E}(k)$ are zero). Otherwise this expectation is non-zero. Therefore, a threshold can be used to decide whether a group of sensors, ϕ, contain any faulty sensors. If $\|E[\mathbf{e}(k)]\|$ is larger than this threshold, the group, ϕ, will be regarded as having at least one faulty sensor and the corresponding entry of Z will be set to 1. Otherwise, the corresponding entry of Z will be set to 0. Fig. 6.1 gives an overview of this approach. In all the algorithm performs M group tests.

After the group test results Z are calculated, the sensor fault state is recovered by a straightforward maximum likelihood (ML) decoding. The recovery algorithm evaluates all $\binom{N}{d}$ possible fault states and chooses the one such that the group testing result Z is most likely, i.e., choose $\nu^\star$ if

$$P(Z|L_\nu^\star) > P(Z|L_\nu) \quad \forall \nu \neq \nu^\star \tag{6.9}$$

where L_ν denotes any possible fault state and $\nu \in \{1, 2, \ldots, \sum_0^d \binom{N}{d}\}$. In some cases, and in particular in the experiments shown in the next section, the probability measure in Eq. (6.9) is difficult to obtain and depends on the threshold used in group testing. This study simply assumes each group test has the same false positive and false negative probability and use minimum distance decoding. For each possible fault state L_ν, the recovery algorithm calculates the Hamming distance, defined as the number of distinct entries, between the predicted output ΦL_ν and the detection outcome Z. Fault states with smaller Hamming distance is preferred. Among fault states having the same Hamming distance from Z, states with a smaller support are preferred as the probability of a sensor being faulty is $< 1/2$. If this still results in a tie, then the recovery algorithm will choose randomly.

6.3 A Combinatorial Group Testing Based Fault Detection Method

In this section, a Combinatorial Group testing (CGT) based fault detection method is presented. This section focuses on the design of the measurement matrix, ϕ. The group test is preformed by the Kalman filter based method which is mentioned in Section 6.2.2. Consider a network of N sensors monitoring an underlying physical system that can be modeled as a linear dynamical system. Assume any sensor in the network can be faulty and that at most d of them are faulty at any given time. The dynamic evolution of the underlying system as well as observations by the sensors can be expressed similarly as in (6.2):

$$\mathbf{Y}(k) = \mathbf{C}\mathbf{X}(k) + \mathbf{V}(k) + \mathbf{E}(k) \ , \tag{6.10}$$

where the additional vector $\mathbf{E}(k)$ is an unknown error vector induced by sensor faults: its i^{th} component is zero if sensor i is not faulty.

6.3.1 Group Selection and Number of Group Tests

Recall the fault detection problem represented as $Z = \Phi S$, where S represents the fault state of sensors ("1" means faulty). As the detection performance largely depends on Φ, our primary task is in determining the entries of Φ, i.e., which sensors include in each test. In this sub-section we focus on the non-adaptive CGT method, whereby Φ is designed prior to the tests.

A common way of selecting test groups, which we adopt in this study, is to design a disjunct measurement matrix. A d-disjunct matrix has the property that for any $d+1$ columns, there is always a row with entry 1 in a column and zeros in all the other d columns. For instance, the measurement matrix in Example VI.1 is 1-disjunct (since any two columns differ in at least one row) but is not 2-disjunct. The reason a d-disjunct matrix is desirable, especially in the case when group tests are error-free, is because its output vector Z is distinct for different d-sparse vectors S (a vector is d-sparse if it has at most d non-zero entries), which means that the exact recovery of a d-sparse fault state vector S is guaranteed with a d-disjunct Φ. One simple method to generate a d-disjunct measurement matrix Φ with high probability is to generate each entry randomly such that $\Phi(i, j) = 1$ has probability $1/2$.

The quality of a measurement matrix is reflected in the number of tests needed (the number of rows in Φ) to gain enough information in order to correctly recover the fault state S. If the group tests are error free and the faulty sensors are distributed uniformly at random, then the necessary and sufficient number of rows in Φ are $\mathcal{O}(d\log(N/d))$ and $\Omega(d\log(N))$, respectively (*Gilbert et al.*, 2012). Under the worse-case distribution of faults (i.e., adversarial fault model), the necessary and sufficient number of rows in Φ are $\mathcal{O}(\frac{d^2\log(N)}{\log(d)})$ and $\Omega(d^2\log(N))$, respectively (*Gilbert et al.*, 2012).

The group tests in our problem is not error-free since detection using Kalman filtering is inherently noisy. Noisy group testing problems are relatively less studied than their noise-free counterpart. A recent study (*Atia and Saligrama*, 2012) has been conducted to evaluate the number of tests required for two noisy group testing scenarios: 1) Additive model: the group result, 0, may change to 1 with probability α; and 2) Dilution model: a faulty sensor may act like a normal sensor (diluted) with probability β in a group test. The sufficient number of tests for the additive model and dilution model, under worst case distribution of faults, are $\mathcal{O}(\frac{d^2\log(N)}{1-\alpha})$ and $\mathcal{O}(\frac{d^2\log(N)}{(1-\beta)^2})$, respectively. However, for group tests that can have both false alarm and miss detection, as in our algorithm, the requirement on the number of tests is still an open question.

6.3.2 Practical Implementation

The method outlined above can be implemented in two ways. The first is as a post processing of data already collected at a cluster head or central location, to which parallel computing techniques can be applied. The second is a form of real-time sequential detection process, where a control center solicits input from a single group of sensors at a time. A single group test is then performed over this group of input. This is followed by soliciting input from the next group, and so on. Note that as long as the fault state of the underlying system remains unchanged, the fault state estimate can be done over different segments of observations over time. In other words, the data provided by each group need not be synchronized and can be generated on demand.

6.4 Experimental Setup

The proposed BGT and CGT fault detection algorithms are evaluated using a set of real bridge vibration sensing data collected from the New Carquinez Bridge in California. In this section, we first present the nature of our sensing data. We then introduce a list common sensor fault types that used in controlled experiments and a set of corrupted data collected on the New Carquinez Bridge that used in fault detection analysis.

6.4.1 Sensor Fault Types

We consider four different fault types: spike, non-linear transduction, mean drift and excessive noise in the controlled experiments. These are illustrated in Fig. 6.2 on a sinusoidal signal. More specifically, a spike fault is an impulse superimposed on normal sensor measurements. They are assumed to occur randomly in time with constant or varying magnitudes (consistent with a random signal model). Moreover, the occurrence of these spikes is assumed sparse. A non-linearity fault represents an abnormal discrepancy between the sensor input and output. This fault usually happens when the measurement falls outside a certain dynamic range. In this study, a simple non-linear fault model is used as shown in Fig. 6.2(e): when the measurement is within the normal region, the sensor output reflects the measurement; otherwise the output follows the slope S_f. A mean drift fault preserves the output dynamics but not its mean value. This type of fault generates outputs whose mean drifts away from the true mean of the signal slowly compared to the output dynamics. Finally, excessive noise refers to a large amount of Gaussian noise in the output of a sensor. Compare to regular measurement noise, this fault has much higher amplitude such that the output signal is highly corrupted. Note that only the non-linearity fault is a

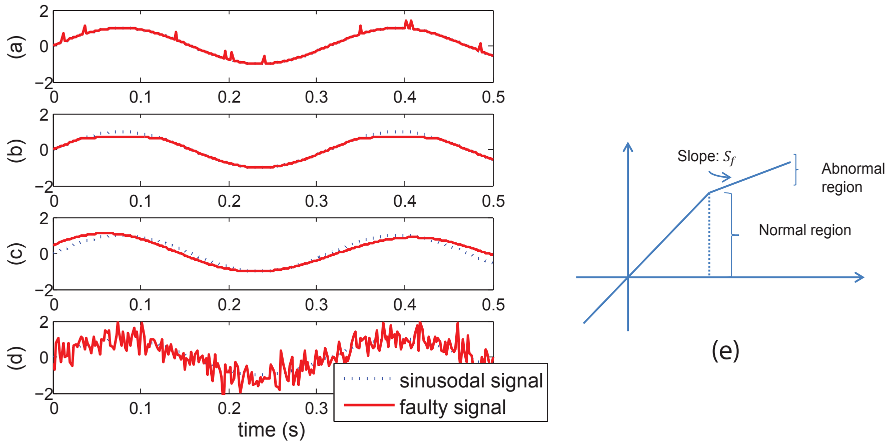


Figure 6.2: Illustration of different faults on a sinusoidal signal: (a) Spike, (b) Non-linearity, (c) mean-drift, (d) Excessive noise and (e) non-linear fault model

function of the measured signal while the other fault types are not.

6.4.2 Bridge Vibration Data and State Estimation

We evaluate our detection method using bridge vibration data collected by a network of 18 vibration sensors deployed on the New Carquinez Bridge in California. This is a 1056-meter long suspension bridge which connects Crockett and Vallejo. The locations of these 18 sensors are shown in Fig. 6.3. They monitor the bridge vibration in the direction perpendicular to the bridge surface. Fig. 6.4 shows an example of the output of a sensor when vehicles pass through. We took 18 data traces at the beginning of the deployment and performed manual inspection. Each data trace consists of 50 seconds of data sampled at 200Hz. All tests, including spectrum analysis and mode-shape calculation on the data suggest that the data traces are correct.

Our first task is to use the collected data to train the linear dynamical model needed in the group testing algorithms. For this we adopt a commonly used approach, the subspace method (*Katayama*, 2005) which utilizes measured output (and input, if available) to calculate model parameters such as matrices **A**, (**B** if input data is

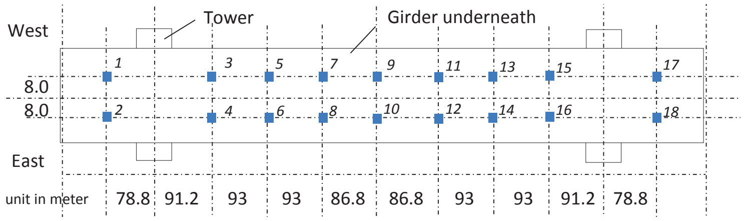


Figure 6.3: Plan map of the deployed sensors.

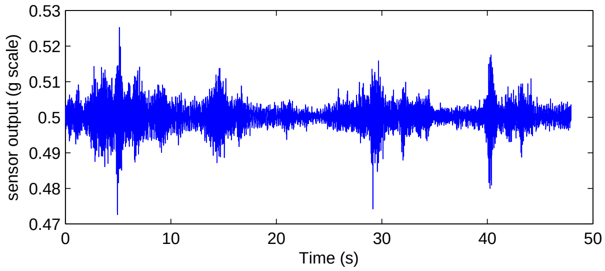


Figure 6.4: Vibration measurement of a sensor

available) and **C** in the state-space model (6.10). Notice that the excitation/input to the bridge is in general unavailable. While input is not necessary for learning the system model by the subspace method, prior study suggests the input can be assumed to be Gaussian for large structures with complex excitations, and that this leads to a better learned system model in terms of output prediction (*Tong and Perreau*, 1998). For our study, we use half of of the vibration data from each of the 18 traces for training of the bridge dynamical model, and the other half for evaluating the group testing method. The order of the dynamical model is set to 162 (An earlier study of the bridge, (*Kurata et al.*, 2012), indicates that a 162-order state space model is sufficient to capture the bridge dynamics), i.e., the length of the state vector is 162. The excitation inputs are assumed to be 18 degree-of-freedom Gaussian signals and each degree-of-freedom input has zero mean and variance equal to the variance of the output of the sensors.

Two experiments are then conducted to evaluate the performance of the proposed algorithms. The first is a control experiment, whereby different fault types are artificially created and superimposed over a random subset of the data traces. The resulting data are then used for evaluation purposes. Specifically, we add different types of faults to the bridge data by randomly selecting up to two sensors (a number ς is first chosen uniformly from $\{0, 1, 2\}$, and then ς number of intended faulty sensors are chosen uniformly among the 18 sensors). We set the maximum number of faulty sensor to be 2 ($d = 2$) so as to keep the percentage of faulty sensors around 10%. A total of 100 random runs are conducted (over the choice of the number and identity of the faulty sensors, as well as over the random injection of faults and the generation of the Φ matrix) for experiments.

In addition to the control experiment, we also evaluated the CGT algorithm performance on real sensor faults. Several weeks after deployment, sensor 11 appears to start having errors (this is done by manual and visual inspection of its data). As shown in Fig. 6.9, the output of sensor 11 has obvious spikes beyond normal fluctuation, and possibly has a shift on the mean amplitude and a small mean-drift error as well. It should be noted that this observation is not the absolute "ground truth" but is the closest we can possible get under the circumstances (the alternative is to take the sensor off the bridge and calibrate it in a lab; even if we could do so the result is only valid if the same type of faults persists in the lab setting).

6.5 Performance of the Combinatorial Group Testing (CGT) Method

In this section we evaluate the performance of the CGT algorithm. The performance in detecting different fault types is evaluated by control experiments. The algorithm is then evaluated on detecting the real faulty sensor shown in Fig. 6.9. Finally, we compare the CGT algorithm to non-group testing methods, in terms of accuracy and efficiency.

We first examine the performance of the CGT algorithm as a function of the

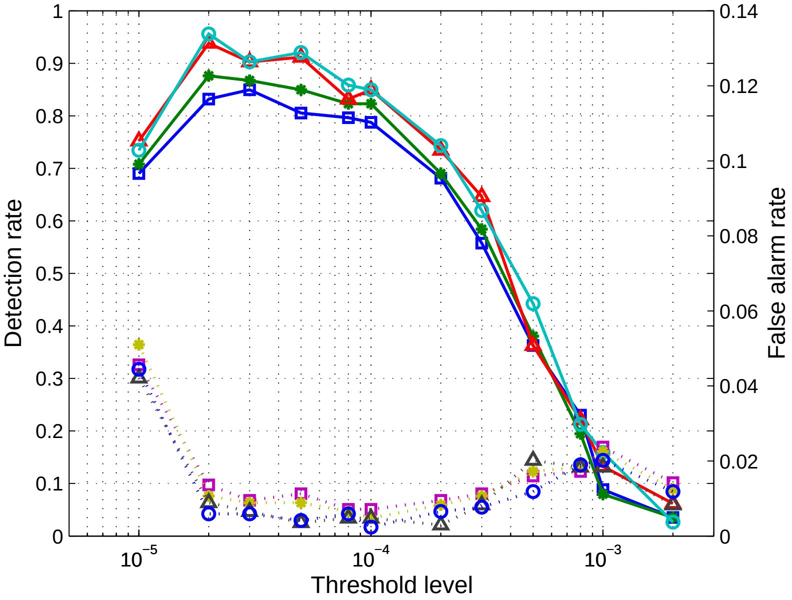


Figure 6.5: Detection and false alarm rate on detecting spike fault.

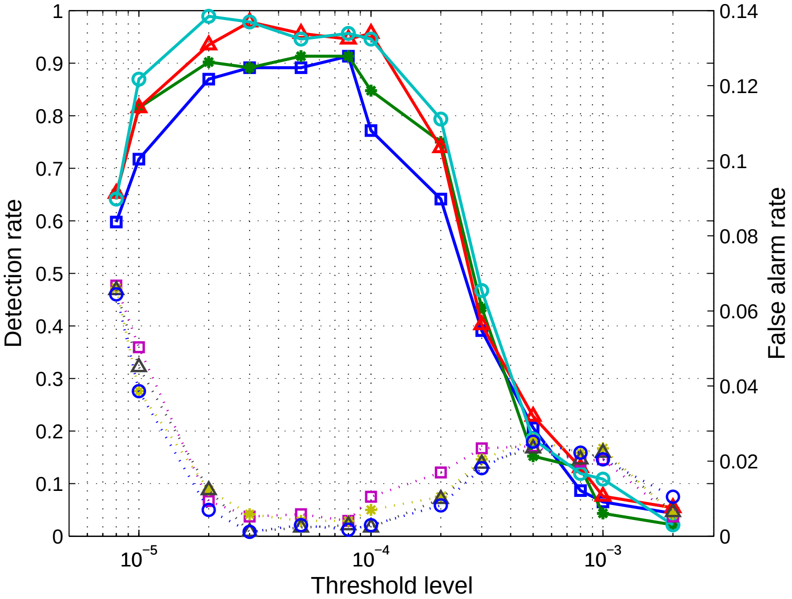


Figure 6.6: Detection and false alarm rate on detecting Non-linearity fault.

detection threshold used in each group test and the number of tests performed. Fig. 6.5 shows the detection rate (the number of detected faulty sensors over the total number of faulty sensors) and false alarm in detecting spike fault under different number of tests and threshold levels. The spike fault was set to appear at 5% of the samples and have mean amplitude equal to the variance of the sensor output, which is common among spike fault in sensors. As can be seen, when the number of tests increases, the detection rate increases while false alarm decreases. When 14 tests are used, the detection rate is above 85% and false alarm is below 1%, with a threshold of 2×10^{-5}. Similarly, when 16 tests are used, the accuracy is over 93% and remains above 80% with a threshold less than 2×10^{-4}.

In all cases we see a fairly wide region of threshold values within which the method enjoys high detection rate ($> 80\%$) and low false alarm ($< 2\%$). This is clearly a desired operating regime for the detection method. In addition, the detection rate first increases with the threshold and then drops slowly with further increase in the threshold. When the threshold increases beyond a certain value (e.g., 3×10^{-4}), the detection rate quickly drops and eventually reaches zero. The false alarm moves in the opposite direction though to a lesser degree. To explain this phenomenon we note there are two sources of error at play, one due to Kalman filtering and the other due to the recovery algorithm. When the threshold is very low, measurement noise or inaccuracy in the model could easily result in false positive in the the group test. These incorrect group testing results cause the recovery algorithm to err, thus lead to both high false alarm and low detection rate. As the threshold increases the error from recovery decreases, which more than compensates for the decreased sensitivity in the group testing, achieving an overall better tradeoff. When the threshold increases beyond a certain level, the group test becomes insensitive to faults and eventually declares all groups normal, resulting in reducing detection rate and false alarm.

The same evaluation is done for the other fault types; these are shown in Figs. 6.6 and 6.7. In Fig. 6.6, results of detecting non-linearity fault are shown. The normal dynamic range is set to 80% of the output maximum, with a slope in the abnormal region of 0.3. The result for mean-drift error is presented in Fig. 6.7. The mean-drift has a maximum frequency of $5Hz$ and a magnitude of 50% of the sensor output variance. All these results show similar behavior to those observed in the spike fault case. Within the preferred threshold range, the detection rate generally exceeds 80% in accuracy while false alarm remains low. Furthermore, the preferred threshold range is smaller when the fault is less pronounced. Finally, the detection performance of the proposed method is tested when the sensor is corrupted by excessive Gaussian noise with zero mean and variance equal to 50% of the variance of sensor output. The result presented in Fig. 6.8 shows that the proposed method is not recommended for detecting this type of fault. The poor detection performance in this case is due to the fact that Kalman filtering, in computing statistically optimal estimates of the system state, tends to eliminate noise variance existing in the sensor measurement. Consequently, zero-mean excessive noise is sufficiently suppressed in the estimate and does not get reflected in the residual of a group test.

For detecting the faulty sensor 11 shown in Fig. 6.9, we used our algorithm on the 18 sensors with 6 and 8 tests respectively. Under the same preferred threshold range (between 3×10^{-3} and 1×10^{-4}) shown in the control experiment, our algorithm was able to identify the faults in sensor 11, with a detection rate $> 78\%$ ($> 92\%$) and false alarm $< 1.8\%$ ($< 0.7\%$) when using 6 (resp. 8) tests.

Next, the proposed combinatorial group-testing based detection method is compared to two existing Kalman-filter based methods, which are Kobayashi *et al.* (*Kobayashi and Simon*, 2003) and Da *et al.* (*Da and Lin*, 1995)., both in terms of their complexity and accuracy. Both Kobayashi and Da are based on a bank of Kalman filters. Specifically, with N sensors in the network, N fault detection tests (N Kalman filters) are required to evaluate all sensors in the network. In each test, all sensors but one are involved, i.e., test i uses $N-1$ sensors and exclude sensor i. A key assumption in this method is that there is *only one* faulty sensor in the network, thus the test which does not contain the faulty sensor will have different characteristics than the

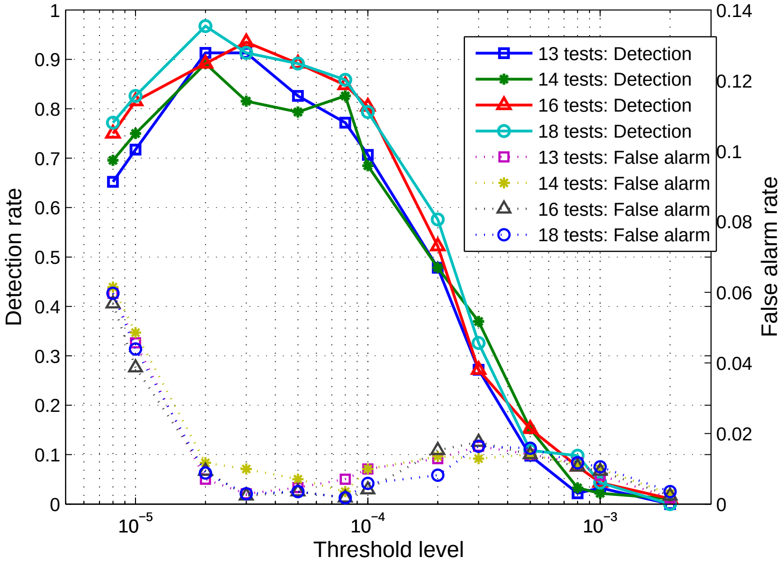


Figure 6.7: Detection and false alarm rate on detecting mean-drift fault.

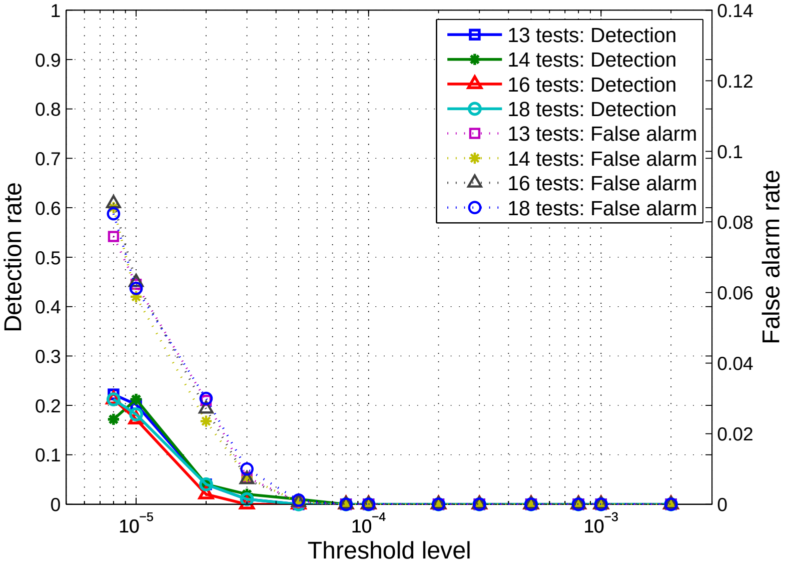


Figure 6.8: Detection and false alarm rate on detecting excessive-noise fault.

other $N-1$ tests, and thus the single faulty sensor can be identified.

The difference between these two methods lies in how to compare the test outcomes to determine the different characteristics with and without the faulty sensor. Under the method by Kobayashi, the estimated sensor output from the Kalman filter is

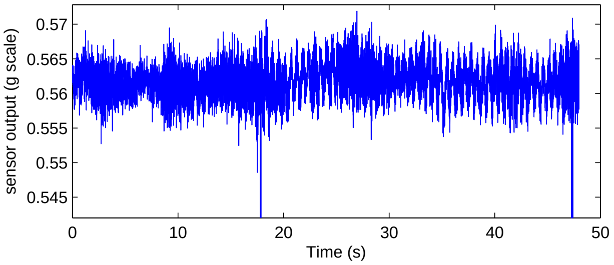


Figure 6.9: Abnormal vibration measurement of sensor 11.

compared to the corresponding observed sensor output. The test which does not contain the faulty sensor will have higher consistency result than the other tests. Under the method by Da, a reference system state estimate is generated by using all N sensor inputs, to which each test compares the estimated system state (from $N-1$ sensors). The test that does not contain the faulty sensor is supposed to have lower consistency result because the reference system contains faulty sensor but the test does not.

Fig. 6.10 shows the detection rate of the three methods under different types of faults, different measurement noises, and with a single faulty sensor, using the same set of bridge data as in the previous section. As we can see, Kobayashi and Da's method achieve similar performance as our proposed method when 8 to 10 tests are used. This result is to be expected when the assumption of no more than one faulty sensor holds, since all methods are based on Kalman filter. As shown in Section 6.2, the the complexity of Kalman filtering largely depends on the size of the system state $\mathbf{s}$, rather than the number of sensors used in state estimation. *One* detection test of Da's and Kobayashi's algorithms has similar complexity as *one* group detection test of the proposed group-testing based detection method if the sensor network size remains the same. Therefore, our proposed method is able to achieve similar, and sometimes better, accuracy when around 8 to 10 tests are used, which is about half of the complexity of Kobayashi's and Da's method (18 tests). The results in Fig. 6.10 also suggested that Kalman filter based fault detection systems are insensitive

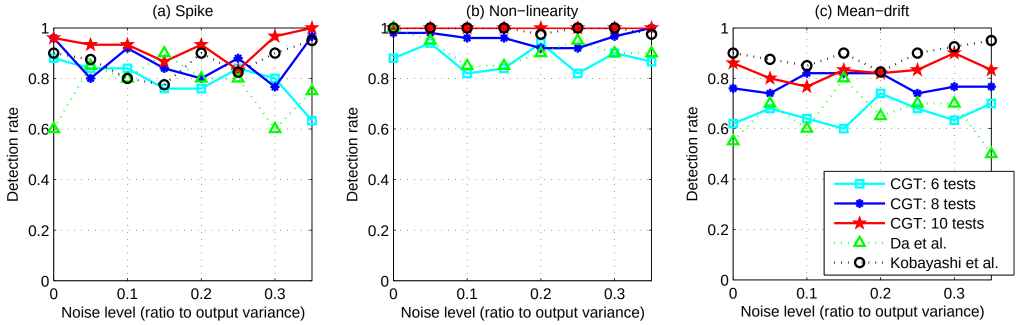


Figure 6.10: Detection rate under different measurement noises and fault types with non-adaptive threshold.

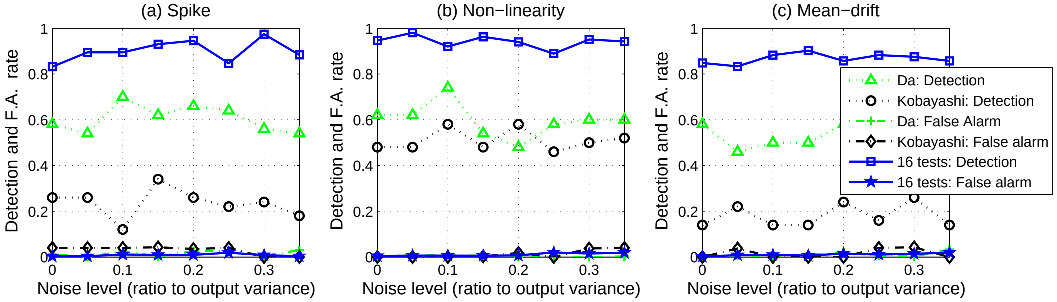


Figure 6.11: Detection rate under different measurement noises and fault types with two faulty sensors.

to Gaussian measurement noise, which is consistent with result in detecting excessive Gaussian noise faults. No significant degradation in the detection rate and false alarm is observed when the variance of the measurement noise increases from 0% of output variance to 30% of output variance.

When the system has two faulty sensors, the performance of Kobayashi and Da's method deteriorates sharply as all the reference system are contaminated by faulty sensor observations. If the false alarm rate is restricted to a reasonable level (5%), the accuracy of Da's method dropped to about 55% and Kobayashi's method dropped to about 50% for non-linearity fault and to about 20% for spike and mean drift fault (Fig. 6.11). At the same time, the proposed algorithm maintains over 85% of accuracy for all fault types. Therefore, compared to other model-based methods, the proposed

CGT method has fewer assumptions on the underlying system and the nature of the faults. It achieves high accuracy with much lower complexity than existing methods, which is particularly relevant for very large sensor networks. Furthermore, the above comparison shows that the proposed method is insensitive to measurement noise. When the system has three faulty sensors, the CGT method is able to achieve high detection rate (90% or higher) by increasing the number of tests (25 tests for detecting spike, 24 tests for detecting non-linearity and 27 tests to detect mean-drift). While these exceed the size of the network (18 sensors), this method does not require the existence of a reference system/sensor.

6.6 Summary

This chapter presents a non-adaptive Group Testing based sensor fault detection method. Combining the Kalman filter based group test method and Combinatorial Group Testing technique, experiment results show that the proposed algorithm is able to reduce the number of tests significantly while achieving similar accuracy. The maximum likelihood decoder allows the Kalman filter based group test method to have small error rate. However, it has high complexity when the size of the network is large. The next chapter presents a Bayesian based adaptive Group Testing method which can further reduces the number of required tests and support noisy group testing systems natively.

CHAPTER VII

Efficient Sensor Fault Detection Using Bayesian Group Testing

7.1 Introduction

This chapter presents an adaptive group testing sensor fault detection method which is based on Bayesian inference. This Bayesian Group Testing (BGT) method adopted the same Kalman filter based method as used in CGT method to evaluation the binary state of a group of sensors. However, different from the non-adaptive CGT method presented in Chapter VI, this adaptive method generate test pools iteratively after the observation of each test results. This is because the test results contain useful information for better test pool selection.

Existing adaptive group testing methods generally assumes error-free detection, thus an entire group of sensors is removed from further consideration when the test result is negative. Examples include Hwang's generalized binary splitting algorithm (*Hwang*, 1972), Allemann's split-and-overlap algorithm (*Allemann*, 2003) and Du *et al.*'s competitive GT algorithm (*Du and Hwang*, 1993). Test errors have been considered in the literature of compressive sensing, (e.g., see (*Malloy and Nowak*, 2012; *Ji et al.*, 2008)), which is closely related to group testing. However, these adaptive methods are not directly applicable to group testing as the latter is given by a Boolean operation whereas compressive sensing based test results are given by a linear operation. Our study further differs from both because our test results are given by a Kalman filtering based detection procedure (neither Boolean nor a linear operation), which is noisy and depends on the design of the test and the detector. This raises significant challenge that we will address in this paper.

This study further consider the detection performance of Kalman filtering, and use such understanding in determining the selection of test groups under the Bayesian group testing method; this will be referred to as the Kalman filtering-enhanced Bayesian group testing method (KF-BGT). It should be emphasized that under all these methods the group tests (the second component) themselves are performed via Kalman filtering; they simply differ in how the test groups are selected (the first component).

7.2 Bayesian Group Testing

In this section, we present a novel adaptive group testing method which is based on the Bayesian inference model. The combinatorial group testing method presented in the previous section designs the entire set of test pools (i.e., the entire Φ) before carrying out any group test. The result of each group test, however, may provide valuable information on the sensor state. For instance, for simple error-free group test systems, a negative group test result implies all the items in the test pool are normal and no further test is required for these items. If the design of a test pool takes the previous test results in account (i.e., adapt to the group test results), the sensor state can be identified with fewer number of tests compared to the combinatorial group testing method. This idea was adopted in several adaptive group testing methods (*Hwang*, 1972; *Allemann*, 2003; *Du and Hwang*, 1993). Although these methods are effective in reducing the number of required tests, they are vulnerable to errors in group test results.

Our method maintains a probability measure on the sensor fault state vector, which is updated following each group test using Bayesian inference. The updated state estimate is then used to determine the next test pool. This process is repeated until the change in the state estimates is sufficiently small. As we shall see, compared to existing adaptive group testing methods, our algorithm is designed specifically for noisy group tests so that errors do not propagate.

In the following presentation, subscript k is used to denote the k^{th} component (row) of a vector (matrix) and superscript k to denote the collection of a variable from time 1 to k. Specifically, denote by $\Phi^k = \{\Phi_1, \Phi_2, \ldots, \Phi_k\}$ the set of tests used up to time k, where Φ_k is the k^{th} row vector of Φ, and $Z^k = \{Z_1, Z_2, \ldots, Z_k\}$ the set of test results up to time k. Let $\mathcal{S}$ be the collection of all possible sensor fault states $\{S = (S_1, S_2, \ldots, S_N) : S_i \in \{0, 1\}\})$. We define two probability measures. The first is $P_{S,k} = P(S|\Phi^k, Z^k)$, the probability of the sensor state being $S \in \mathcal{S}$ after the k^{th} test; the second is $P_{i,k}$, the probability of sensor i being normal after the k^{th} group test. By definition, we have $P_{i,k} = \sum_{S\in\mathcal{S}:S_i=0} P_{S,k}$.

For the $(k+1)^{th}$ test Φ_{k+1}, it is desirable to select sensors such that the test result Z_{k+1} provides the most information for the estimation of the true sensor state. Basic information theory result (*Cover and Thomas*, 2012) tells us that maximizing the information content is equivalent to maximizing the variance of Z_{k+1}. This criterion can be expressed as follows:

$$\Phi^*_{k+1} = \arg\max_{\Phi_{k+1}} VAR[Z_{k+1}|\Phi_{k+1}, \{P_{S,k}\}_{S\in\mathcal{S}}] \; . \tag{7.1}$$

Z_{k+1} conditioned on $\Phi_{k+1}, \{P_{S,k}\}_{S\in\mathcal{S}}$ has a Bernoulli distribution. If we denote by Ω_k the probability that all sensors in test pool Φ_{k+1} are normal given the estimate after the k^{th} observation, then the above variance is given as follows, noting that $Z_{k+1} = 0$ either when all sensors in Φ_{k+1} are normal and the group test is correct or when at least one sensor in Φ_{k+1} is abnormal and the group test is incorrect, i.e., $Z_{k+1} = 0$ with probability $((1-\alpha)\Omega_k + \beta(1-\Omega_k))$, and similarly $Z_{k+1} = 1$ with probability $(\alpha\Omega_k + (1-\beta)(1-\Omega_k))$.

$$\begin{aligned} & VAR[Z_{k+1}|\Phi_{k+1}, \{P_{S,k}\}_{S\in\mathcal{S}}] \\ & = ((1-\alpha)\Omega_k + \beta(1-\Omega_k))(\alpha\Omega_k + (1-\beta)(1-\Omega_k)) \\ & = \beta - \beta^2 + (1-2\beta)(1-\alpha-\beta)\Omega_k - (1-\alpha-\beta)^2\Omega_k^2 \ . \end{aligned} \tag{7.2}$$

The above computation, however, is generally intractable due to the large state space $\mathcal{S}$ when the number of sensors is large. We thus adopt the following approximation by assuming conditional independence between different sensors' fault states, i.e.,

$$P(S_1, S_2, \ldots, S_N|\{P_{S,k}\}_{S\in\mathcal{S}}) = \prod_{i\in N} P(S_i|\{P_{S,k}\}_{S\in\mathcal{S}}) \ , \ \forall k \ . \tag{7.3}$$

With this assumption we have $\Omega_k = \prod_{i\in\Phi_{k+1}} P_{i,k}$, where we have used $i \in \Phi_{k+1}$ to mean that the i^{th} component of Φ_{k+1} is 1.

While this assumption allows us to compute (7.2), finding the optimal solution to (7.1) remains hard when the number of sensors is large. Toward this end we propose a greedy algorithm for choosing a good Φ_{k+1} efficiently, by observing from (7.2) that its maximum is achieved when $\Omega_k^* = (1-2\beta)/(2(1-\alpha-\beta))$. The greedy algorithm starts with a random sensor and calculates Ω_k; in each successive step it selects a sensor such that the resulting new value of Ω_k is as close to $(1-2\beta)/(2(1-\alpha-\beta))$ as possible. This is repeated until no additional sensor can bring Ω_k closer to $(1-2\beta)/(2(1-\alpha-\beta))$. As Ω_k is monotonically decreasing in the inclusion of new sensors, the algorithm is guaranteed to terminate with a new test pool.

Having designed Φ_{k+1} and observed Z_{k+1}, the probability $P_{S,k+1}$ can be updated from $P_{S,k}$ for all $S \in \mathcal{S}$:

$$\begin{aligned} P_{S,k+1} &= P(S|\Phi^{k+1}, Z^{k+1}) = \frac{P(S, \Phi^{k+1}, Z^{k+1})}{P(\Phi^{k+1}, Z^{k+1})} \\ &= \frac{P(Z^{k+1}|Z^k, S, \Phi^{k+1})P(S|\Phi^k, Z^k)P(\Phi^{k+1}, Z^k)}{P(\Phi^{k+1}, Z^{k+1})} \\ &= P(Z_{k+1}|Z^k, S, \Phi^{k+1})P_{S,k}/\Delta_k \ , \end{aligned} \tag{7.4}$$

where Δ_k is the normalizing factor $P(\Phi^{k+1}, Z^{k+1})/P(\Phi^{k+1}, Z^k)$, and is equal to $\sum_S P(Z_{k+1}|Z^k, S, \Phi^{k+1})P_{S,k}$.

Note that $P(Z_{k+1}|Z^k, S, \Phi^{k+1}) = P(Z_{k+1}|\Phi_{k+1}S)$ as Z_{k+1} only depends on the error-free test result $\Phi_{k+1}S$; recall the two type of errors are given by $P(Z_{k+1} = 1|\Phi_{k+1}S = 0) = \alpha$ and $P(Z_{k+1} = 0|\Phi_{k+1}S = 1) = \beta$.

To update the sensor state probabilities using (7.4) for each $S \in \mathcal{S}$ can be computationally prohibitive for large N ($|\mathcal{S}| = 2^N$). Below we show that using the conditional independence assumption we can instead update $P_{i,k+1}$ directly without calculating $P_{S,k+1}$, thus reducing the complexities from $\mathcal{O}(2^N)$ to $\mathcal{O}(N)$. We first calculate the normalization constant, and then update $P_{i,k+1}$ accordingly.

Given a test pool Φ_{k+1}, we will refer to the set of sensor states $\{S : \Phi_{k+1}S = 1\}$ as the *positive set*, and $\{S : \Phi_{k+1}S = 0\}$ as the *negative set*. Note that by definition, we have $\sum_{S:\Phi_{k+1}S=0} P_{S,k} = \Omega_k$ and $\sum_{S:\Phi_{k+1}S=1} P_{S,k} = 1 - \Omega_k$. By separating $\mathcal{S}$ into these two sets, Δ_k can be calculated as follows:

$$\begin{aligned} \Delta_k =& \sum_S P(Z_{k+1}|\Phi_{k+1}S)P_{S,k} \\ =& \sum_{S:\Phi_{k+1}S=1} P(Z_{k+1}|\Phi_{k+1}S)P_{S,k} + \sum_{S:\Phi_{k+1}S=0} P(Z_{k+1}|\Phi_{k+1}S)P_{S,k} \\ =& P(Z_{k+1}|\Phi_{k+1}S = 1)(1 - \Omega_k) - P(Z_{k+1}|\Phi_{k+1}S = 0)\Omega_k \end{aligned} \tag{7.5}$$

Therefore, if the test result is positive, $Z_{k+1} = 1$, then $\Delta_k = (1-\beta)(1-\Omega_k) - \alpha\Omega_k$;

if the test result is negative, $Z_{k+1} = 0$, then $\Delta_k = (\beta)(1-\Omega_k) - (1-\alpha)\Omega_k$.

We next show how $P_{i,k+1}$ is updated. If sensor $i \in \Phi_{k+1}$, then using (7.4) we have

$$\begin{aligned}
P_{i,k+1} &= \sum_{S:S_i=0} P_{S,k+1} = 1 - \sum_{S:S_i=1} P_{S,k+1} \\
&= 1 - \sum_{S:S_i=1} P_{S,k} P(Z_{k+1}|\Phi_{k+1}S = 1)/\Delta_k \\
&= 1 - (1-P_{i,k})P(Z_{k+1}|\Phi_{k+1}S = 1)/\Delta_k \\
&= \begin{cases} 1-(1-P_{i,k})(1-\alpha)/\Delta_k & \text{if } Z_{k+1} = 1, \\ 1-(1-P_{i,k})(1-\beta)/\Delta_k & \text{if } Z_{k+1} = 0. \end{cases}
\end{aligned} \tag{7.6}$$

If sensor $i \notin \Phi_{k+1}$, then using (7.5) we have:

$$\begin{aligned}
P_{i,k+1} = \sum_{S:S_i=0} P_{S,k+1} &= \sum_{S:S_i=0,\Phi_{k+1}S=1} P_{S,k+1} + \sum_{S:S_i=0,\Phi_{k+1}S=0} P_{S,k+1} \\
&= \sum_{S:S_i=0,\Phi_{k+1}S=1} P_{S,k}P(Z_{k+1}|\Phi_{k+1}S)/\Delta_k + \sum_{S:S_i=0,\Phi_{k+1}S=0} P_{S,k}P(Z_{k+1}|\Phi_{k+1}S)/\Delta_k \\
&= P_{i,k}(1-\Omega_k)P(Z_{k+1}|\Phi_{k+1}S = 1)/\Delta_k + P_{i,k}\Omega_k P(Z_{k+1}|\Phi_{k+1}S = 0))/\Delta_k \\
&= P_{i,k}\Delta_k/\Delta_k = P_{i,k}
\end{aligned} \tag{7.7}$$

where the fourth equality is due to the independence assumption. As a result, when $i \notin \Phi_{k+1}$, the corresponding $P_{i,k+1}$ remains unchanged.

	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$\mathcal{S}_5$	$\mathcal{S}_6$	$\mathcal{S}_7$	$\mathcal{S}_8$
S_1	0	0	0	0	1	1	1	1
S_2	0	0	1	1	0	0	1	1
S_3	0	1	0	1	0	1	0	1

Table 7.1: Possible sensor states

For example, if the network has 3 sensors, S_1, S_2 and S_3, then there are 8 possible states $\mathcal{S}_i, i = 1, \ldots, 8$ as shown in Table 7.1. assume the initial prior $P_{\mathcal{S}_i,0} = 1/8, \forall i$ (*i.e.*, $P_{i,0} = 1/2, \forall i$) and $\alpha = \beta = 0$ (i.e., the group tests are noiseless). Suppose the first test has only S_1 and S_2 participated in the test and the test result is positive (i.e., $\Phi_1 = [1,1,0]$ and $Z_1 = 1$). A straight forward method to calculate the sensor state probability is update the state prior by Eq. (7.4). Therefore, $P_{\mathcal{S}_{\infty},\prime}, P_{\mathcal{S}_{\in},\prime}$ are first multiplied by $P(Z_{k+1} = 1|W_{k+1} = 0) = \alpha = 0$ and $P_{\mathcal{S}_{\infty},\prime}, \ldots, P_{\mathcal{S}_{\in},\prime}$ are first multiplied by $P(Z_{k+1} = 1|W_{k+1} = 1) = 1 - \beta = 1$. Therefore, $\{P_{\mathcal{S}_{\infty},\prime}, \ldots, P_{\mathcal{S}_{\forall},\prime}\}$ becomes $\{0, 0, 1/8, \ldots, 1/8\}$. Normalizing these values by $\Delta_k = (1/8) \times 6 = 3/4$, we get $\{P_{\mathcal{S}_{\infty},\infty}, \ldots, P_{\mathcal{S}_{\forall},\infty}\} = \{0, 0, 1/6, \ldots, 1/6\}$. As a result, $P_{i=1,1} = 1/3$, $P_{i=2,1} = 1/3$ and $P_{i=3,1} = 1/2$.

First the normalization factor $\Delta_0 = \alpha(P_{i=1,0}P_{i=2,0}) - (1-\beta)(1 - P_{i=1,0}P_{i=2,0}) = (1 - 1/4) = 3/4$. Second, for sensors involved in the test, the update of probability $P_{i,1}$ is based on the state probabilities with $S_i = 1$. Therefore, $1 - P_{i=1,1} = (1 - P_{i=1,0})P(Z_1 = 1|W_1 = 1)/\Delta_0 = 1/2 \times 4/3 = 2/3$, thus $P_{i=1,1} = 1/3$. Similarly, we get $P_{i=2,1} = 1/3$. Third, for sensors not involved in the test, *i.e.*, S_3, the calculation of $P_{i=3,1}$ involves of two sets of states: 1) the states with $S_3 = 0$ and belong to the positive set and 2) the states with $S_3 = 0$ and belong to the negative set. The state probabilities of the first set are multiplied by $(1 - P_{i=1,0}P_{i=1,2})P(Z_{k+1}|W_{k+1} = 1)/\Delta_0$ and the second set are multiplied by $(P_{i=1,0}P_{i=1,2})p(Z_{k+1}|W_{k+1} = 0))/\Delta_k$. In this example, $P_{i=3,1} = 0 \times (1 - 1/4) \times 1 \times 4/3 + 3 \times 1/8 \times 1 \times 4/3 = 1/2$.

The above computational procedure is repeated after each test, starting from some assumed initial prior $P_{i,0}$. After k tests and given Z^k and Φ^k, the sensor fault state S can be recovered in two ways: (1) use the maximum a posteriori probability (MAP) estimator: $\arg\max_S P(Z^k|S, \Phi^k)P_{S,k}$, or (2) declare the i^{th} sensor faulty if $P_{i,k} < \sigma$ for some predefined threshold σ, and normal otherwise. While both are valid, the second method is preferred as $P_{i,k}$ is readily available from the above updating procedure, whereas the MAP estimation is computationally much more complex. The performance of these two methods is similar as we show in Section 7.3.

7.3 Performance of the Bayesian Group Testing Method

The fault detection performance of the Bayesian group testing (BGT) method is evaluated by two experiments. The first experiment evaluates the the performance of the BGT method by using the same bridge sensor data that are used in the evaluation of the CGT method. Hence, the CGT and the BGT methods can be directly compared. The experimental setup, sensor data collection and sensor faults details can be found in Chapter 6.4. The second experiment evaluates the performance of the BGT method under large scale systems (with 1000 sensors). The BGT method will be compared to a well-known divide-and-conquer based adaptive group testing method proposed by Hwang (*Hwang*, 1972). The influence of the initial prior, $P_{i,0}$, on the fault detection performance is also addressed.

7.3.1 Performance of the BGT Method on the New Carquinez Bridge Sensors

As mentioned in Section 7.2, the BGT method and the CGT method have the same Kalman filter based group test method, therefore the BGT method adopts the same group testing threshold levels as used in the CGT method. The two decoding methods, MAP-based decoder and $P_{i,k}$-based decoder, introduced in Section 7.2 are used and compared. Both decoding methods do not require the knowledge of d (the maximum number of faulty sensors). This is a significant benefit compared to the CGT method when the d is difficult to estimate. The CGT method is not able to get correct result if d is underestimated, due to the d-disjunct measurement matrix requirement. Consequently, d is usually larger than the true value in the CGT methods when d is unknown and thus resources could be wasted.

Fig. 7.1 shows the performance comparison between BGT (with the $P_{i,k}$ decoder) and CGT. Fig. 7.2 compares the two decoders for BGT: the MAP and the $P_{i,k}$ decoder. Both comparisons are evaluated on the same New Carquinz Bridge data,

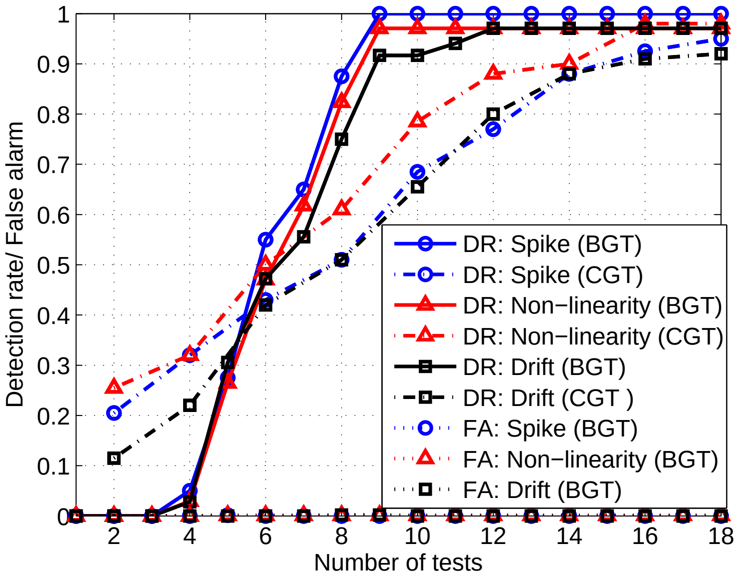


Figure 7.1: The fault detection performance of the CGT method and the BGT method.

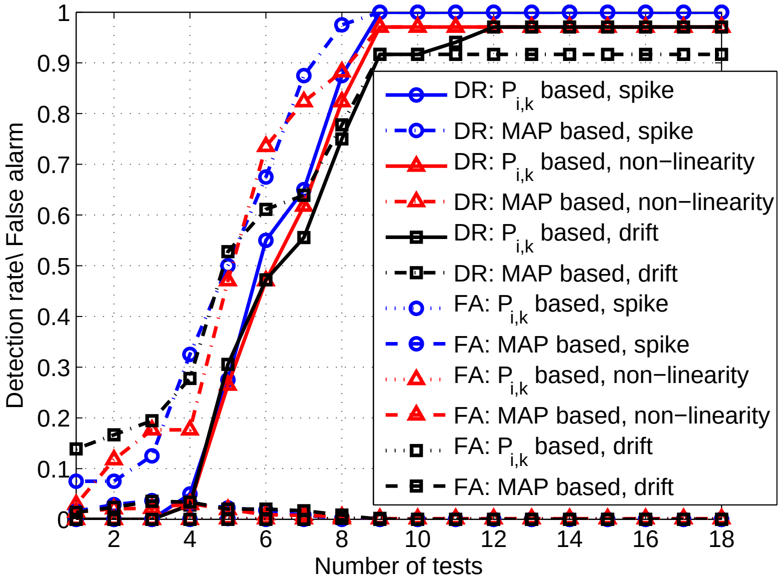


Figure 7.2: The fault detection performance of MAP decoder and the $P_{i,k}$ based decoder.

where 2 out of 18 sensors are faulty. For BGT the initial prior $P_{i,0}$ is set to 2/18 for each sensor. The first test pool is randomly generated with each sensor having probability 1/2 of being selected. The group test error α and β are set to 0.01. When the $P_{i,k}$ decoder is used, sensor i is regarded as faulty when the corresponding $P_{i,k}$ is smaller than 0.2. The results are obtained from 50 random runs using the same setup as in the CGT evaluation in Section 6.5.

As shown in Fig. 7.1, BGT with the $P_{i,k}$ decoder outperforms CGT on detecting all types of faults (when the number of tests > 6). BGT generally requires 3-4 fewer tests than the non-adaptive CGT for 80% detection rate. Moreover, BGT uses 8 fewer tests to reach the saturation accuracy which is about 50% improvement over CGT. The false alarm rates are similar ($< 1\%$) for detecting different type of faults. The improvement is primarily due to two sources: BGT uses previous test results to design the next test, which leads to more effective tests; BGT is more conservative in deciding the sensor state (normal vs. faulty) and thus more robust when the group test is incorrect.

For detecting the faulty sensor 11 shown in Fig. 6.9, the initial prior $P_{i,0}$ is set to 1/18 for all i. By selecting the first test pool Φ_1 randomly, the BGT method is able to achieve 56% detection rate (0% false alarm) when 5 tests are used and 100% detection rate (0% false alarm) when 6 tests are used. The BGT algorithm saves 2 tests compared to the CGT algorithm for the same data set.

Fig. 7.2 compares the two state recover methods introduced in Section 6.3. On average the MAP method is able to save one test for achieving the same accuracy as the $P_{i,k}$-based method. However, the MAP method has higher false alarm when the number of tests falls below 7. Also, the $P_{i,k}$-based method is preferred for large scale networks due to its low complexity. Note that neither decoding method requires the knowledge of d, the maximum number of faulty sensors. This is a significant benefit over CGT if d is difficult to estimate. CGT is not able to get correct result if d is underestimated, due to the d-disjunct matrix requirement. Consequently, if d is unknown then an overestimate is recommended for CGT, which then leads to an over-provisioning of the number of tests.

7.3.2 Performance of BGT Method on Larger-Scale Systems

We next evaluate the performance of BGT in a large scale network (1000 sensors) and examine how it varies with the number of faulty sensors and group test error probabilities. A comparison between BGT and the divide-and-conquer adaptive group testing method proposed in (*Hwang*, 1972) is presented. We note that Hwang's method is designed for noiseless group test systems so it is not expected to work well with noisy group tests. Nevertheless, it is meaningful to compare the two and quantify the difference under both noisy and noiseless conditions. We also address the common prior initialization problem in Bayesian inference which also applies to BGT.

For lack of real data on large networks, the experiments and results presented in this section are simulation based. Out of the 1000 sensors, d are randomly chosen and labeled as faulty. A group test result is first determined by whether the test pool contains any faulty sensors and then randomized according to the error model $\alpha = \beta$, i.e., with probability α, the test result is flipped. In other words, we do not actually perform Kalman filtering based detection in this set of experiments, but its effect is simulated via this error model.

Hwang's method is based on the well-known binary search (*Hwang*, 1972), whereby the network is first divided into 2 groups of equal size, and each is subject to the same group test process. If the result is negative, then all sensors in that group are declared normal removed from further testing; if the result is positive, then the group is further divided into two smaller groups of equal size and the same process repeats until all faulty sensors have been identified. Hwang's method has the following improvement compared to the standard binary search. It assumes knowledge on the number of faulty sensors d (or an upper bound on d), and uses d to determine the size of a group. Specifically, when d is small compared to the total number of uncertain sensors, a large test pool is used. The idea is that upon a negative result a large number of sensors may be declared normal, and a new test pool can be selected from the remaining uncertain sensors; if the result is positive, the next test pool is generated randomly from the entire set of uncertain sensors, including the pool just tested positive. Finally, when the number of remaining faulty sensors (d minus the number of detected faulty sensors) is larger than half of the number of remaining uncertain sensors, the test is performed on an individual basis.

Clearly as mentioned, Hwang's method is designed for error-free group tests, so it does not handle errors well. In particular, if a positive group is mistakenly detected as negative, this method will declare all faulty sensors in this group as normal and no further tests will be performed on them. By contrast, BGT only decreases the probability of each tested sensor being normal, and they may be tested again in the future. The comparison study here thus mainly serves to quantify the improvement we can achieve when taking test errors into account.

Figs. 7.3-7.5 show the performance of BGT and Hwang's method ($d = \{4, 10, 50\}$ respectively) under various group test error rates (α). When group tests are error-free (Fig. 7.3), Hwang's method is able to achieve accurate results with fewer tests than BGT. As expected, when group tests are noisy ($\alpha = 0.03$ in Fig. 7.4 and $\alpha = 0.05$ in Fig. 7.5), BGT performs better while Hwang's method deteriorates rapidly.

A common challenge to most Bayesian inference based methods is the selection of the prior on the hypothesis. Under BGT, the prior probability $P_{i,0}$ is required for designing a test pool. Fig. 7.6 shows the result of using different priors ($P_{i,0} = \{0.3, 0.5, 0.7, 0.9, 0.96\}, \forall i$) when $d = 4$ in a 1000-sensor network with $\alpha = 0$. The case $P_{i,0} = 0.96$ represents the correct prior. The figure shows that the performance is highly sensitive to the selection of the initial prior. However, this effect can be alleviated by choosing the first set of test pools randomly. We see that when the first 25 test pools are randomly selected (each sensor has probability 1/2 to be selected), the difference in performance between different initial priors are significantly reduced

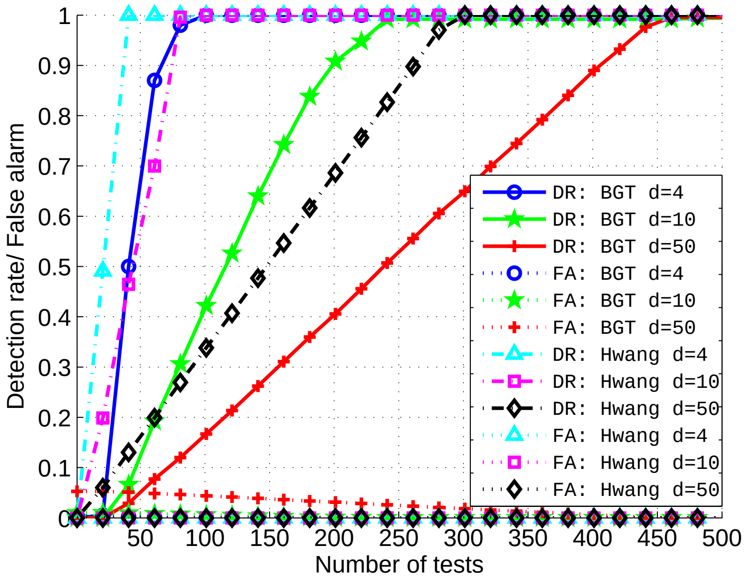


Figure 7.3: The comparison of the BGT and Hwang's methods when group test error $\alpha = 0$.

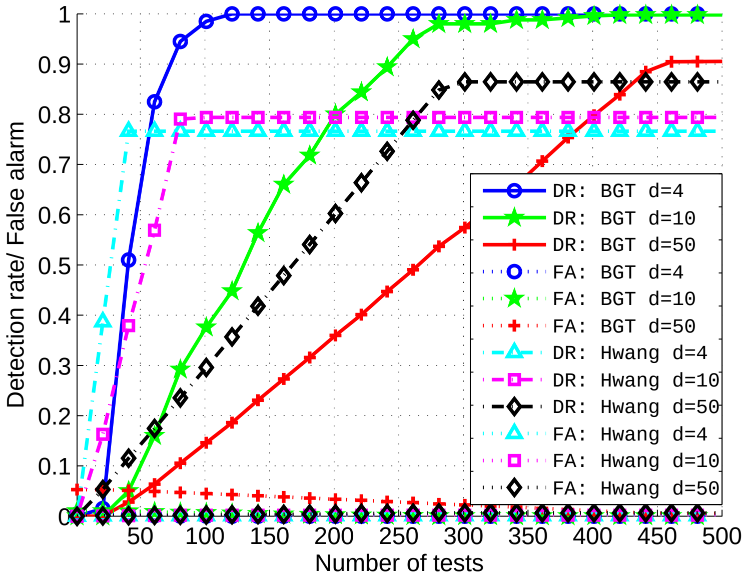


Figure 7.4: The comparison of the BGT and Hwang's methods when group test error $\alpha = 0.03$.

(Fig. 7.7); when we increase this number to 50 tests (Fig. 7.7), this difference is largely eliminated. Thus this random selection at the beginning serves as a very simple yet effective way to counter possible bad priors. It may be seen as a form of

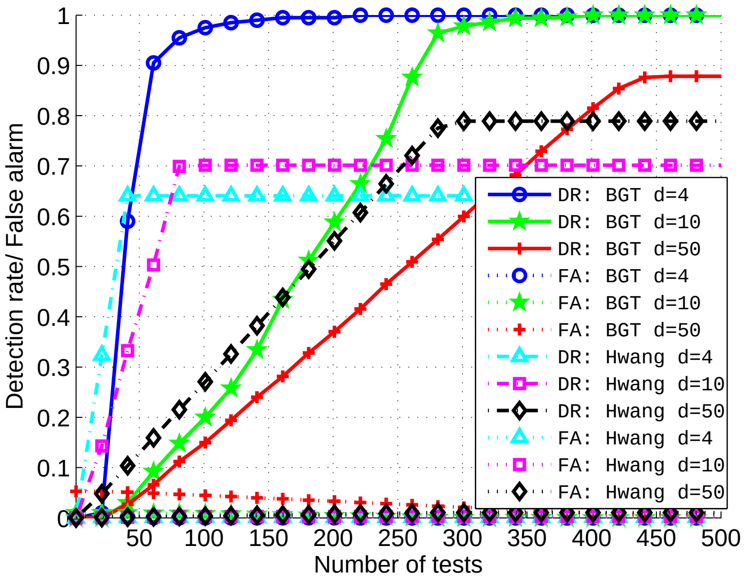


Figure 7.5: The comparison of the BGT and Hwang's methods when group test error $\alpha = 0.05$.

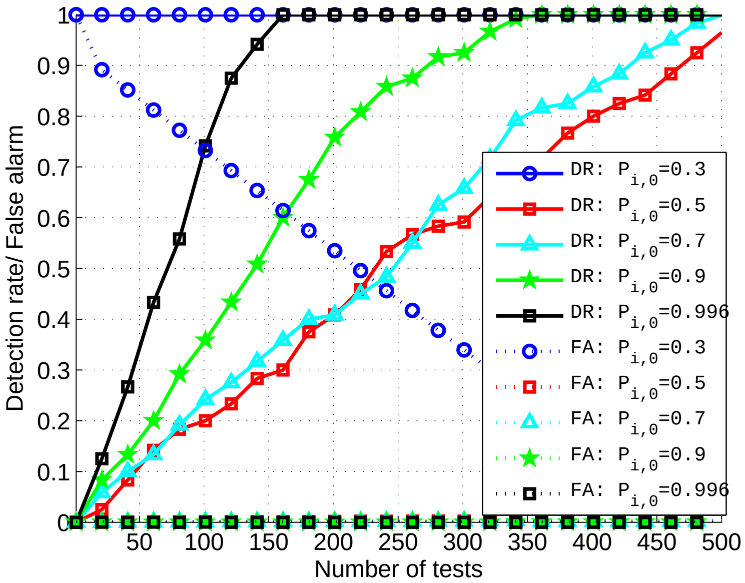


Figure 7.6: The performance of the BGT method under different initial priors $P_{i,0}$ with the first test pool being selected randomly.

exploration (random sampling) prior to *exploitation* (adaptive selection).

To summarize, BGT is able to achieve the same performance as CGT with fewer tests, and is well suited for noisy group tests. Furthermore, it does not require knowl-

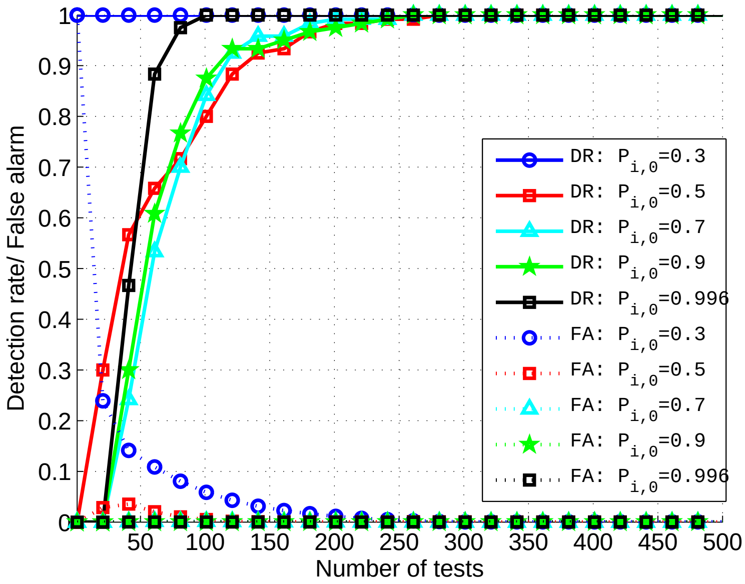


Figure 7.7: The performance of the BGT method under different initial priors $P_{i,0}$ with the first 25 test pools being selected randomly.

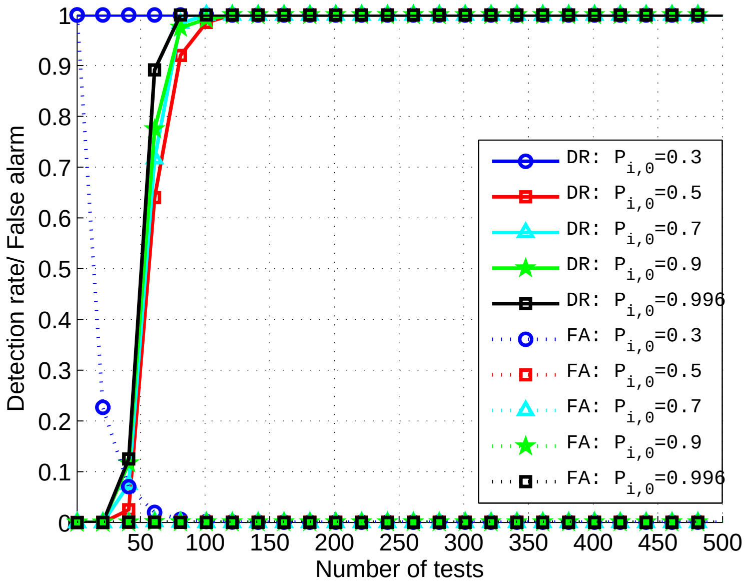


Figure 7.8: The performance of the BGT method under different initial priors $P_{i,0}$ with the first 50 test pools being selected randomly.

edge on d when compared to CGT and Hwang's method. However, the adaptive design process prevents the use of parallel computing, which is viable for CGT. Therefore, CGT may actually have shorter run time if parallel computing is used.

7.4 The Design and Performance of KF-BGT Method

Standard group tests are modeled as boolean operations. While both CGT and BGT work with noisy group tests by modeling it as boolean operations with an error probability, they do not take into account other possible features of the group tests. In our case, the group tests are given by the Kalman filtering based detection procedure, whose accuracy depends on not only the system model but also the test pools. This suggests that a better understanding of the relationship between the detection procedure and the test pool design may allow us to further improve the design of the test pools and in turn the accuracy of the method. This is the subject of investigation in this section.

We note that the Kalman filter estimates the state of a system based on the system model and the measurements from the sensors. As system identification method is used to obtain the system model, the model accuracy depends on the model order (the size of the system state, S) used. A higher order model generally gives better model accuracy (before over-fitting occurs) but it also requires more computational resources for the state estimation. The dependence of the state estimate accuracy on the size of the test group is shown in Fig. 7.9. In this experiment, subgroups of different sizes are used to estimates the system state. For each group size, the discrepancy $|S_A - S_B|_\infty$ is recorded between having no faulty sensors in the subgroups and having one faulty sensor in one of the subgroups. When there are no faulty sensors, the discrepancy $|S_A - S_B|_\infty$ is very close to zero. On the other hand, $|S_A - S_B|_\infty$ is significantly larger with the presence of a single faulty sensor and increases with the group size. This means that if a uniform detection threshold is used, then different group sizes will result in significantly different detection error (i.e., group test error) probabilities. This further suggests that it would be desirable to maintain the same group sizes for the state estimate so as to keep the error probability constant and also to facilitate the choice of an optimal detection threshold.

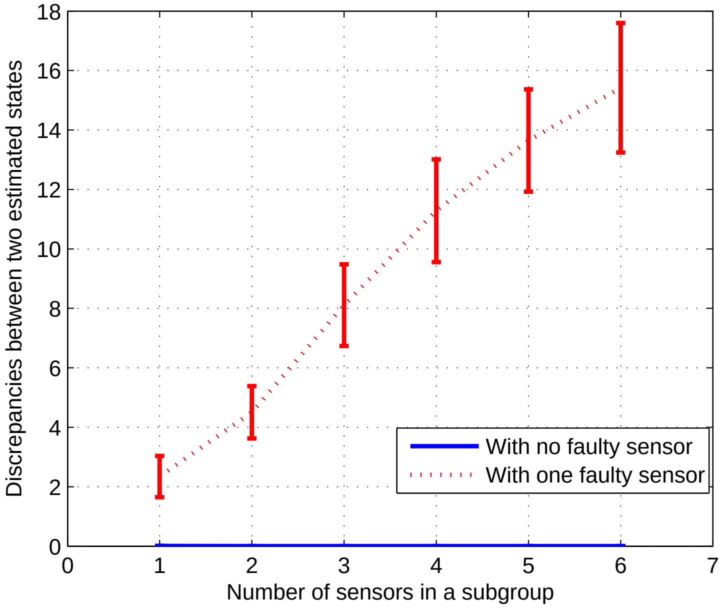


Figure 7.9: The discrepancies of state estimates under different model orders with one and no faulty sensors .

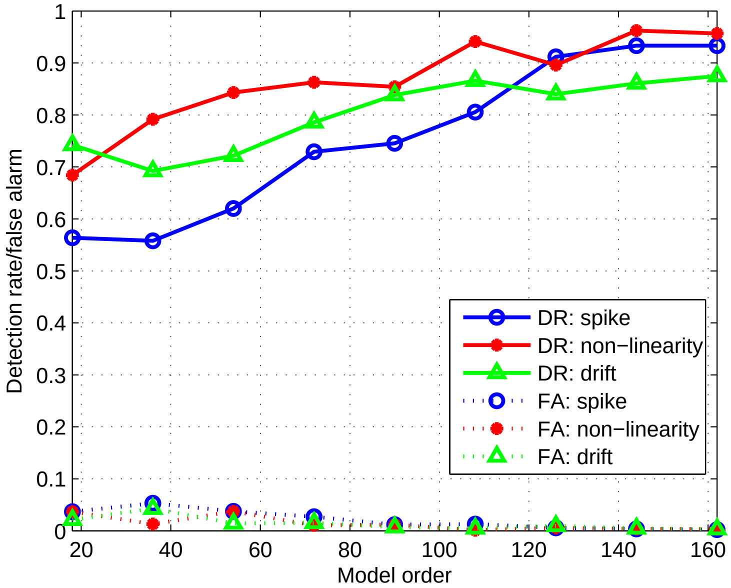


Figure 7.10: The fault detection performance versus different system model orders.

Since only a binary result is required for each group test, a lower order system model may be sufficient for the detection task. Therefore, the performance under model with different model orders is analyzed. Fig. 7.10 shows the detection performance of the CGT method using different bridge model orders from order 18 to order 162. 16 group tests are used in each experiment. The performance of the proposed method deteriorates only slightly with the reduction on the model order. The detection rate of drift faults and non-linearity faults remain at 70% when the model order is 18. The high detection rates is mainly because group testing methods only require binary results from the group tests. The proposed method is able to achieve correct detection as long as a abnormal group leads to a higher discrepancy in state estimate than a normal group. This suggests that the computational complexity can be reduced without sacrificing the performance too much.

Table 7.2: State estimate discrepancy $|S_A - S_B|_\infty$ under various faulty sensor distributions. (G: Number of good sensors, F: Number of faulty sensor)

8 sensors			10 sensors		
Sensor distribution		Discrepancy	Sensor distribution		Discrepancy
A:0G 4F	B:4G 0F	8.29	A:0G 6F	B:6G 0F	10.78
A:1G 3F	B:3G 1F	23.88	A:1G 5F	B:5G 1F	26.73
A:2G 2F	B:2G 2F	41.10	A:2G 4F	B:4G 2F	46.19
			A:3G 3F	B:3G 3F	67.01
A:4G 0F	B:4G 0F	7E-4	A:6G 0F	B:6G 0F	5E-4

We next examine the distribution of faulty sensors between two subgroups used in the filtering detection. Table 7.2 shows the state estimate discrepancy under various faulty sensor distribution in each subgroup. These results show that the discrepancy is highest when faulty sensors are evenly distributed between the two groups, e.g., having a faulty sensor in each subgroup is better than allocating two sensors in one subgroup as the larger discrepancy makes the detection more accurate.

Based on the above empirical observations, we propose the Kalman filtering (KF)-enhanced group test (KF-BGT) that uses the following rule in addition to the operation of BGT: after a new test pool has been selected using BGT, divide it *evenly* into two subgroups. If the there are fewer than 3 sensors in a subgroup, then sensors with high probability of being normal outside the test pool are added to the subgroups before performing Kalman filtering.

The differences in performance with and without the added sensor distribution step is illustrated in the following experiment. Fig. 7.11 shows the detection rate and false alarm for non-linearity fault under BGT and KF-BGT. The performance is evaluated under both order 162 and order 90 system models. The performance of BGT declines significantly under a less accurate system model (smaller model order). In contrast, the performance of KF-BGT only deteriorate slightly, thus it improves upon BGT significantly when the system model is less accurate. This shows that the sensor distribution makes the resulting method highly robust against the quality of the system model.

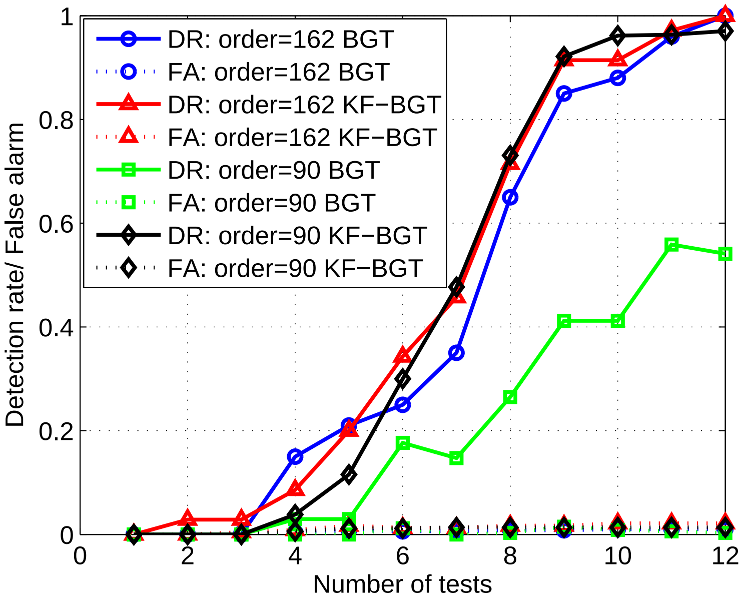


Figure 7.11: The performance of BGT and FK-BGT methods under different model orders.

7.5 Summary

This chapter present a Bayesian based adaptive Group Testing method which is suitable for noisy group test systems. Compared to the Combinatorial Group Testing

(Chapter VI), this adaptive method designs each test pool based on the previous group test result. A probabilistic measure is used to estimate the probability of each sensor being faulty. These probabilities are updated based on the test pool design and the corresponding group test result by Bayesian inference. Based on the updated probabilities, a new test pool is selected such that the test result gives the most information for revealing the faulty sensor(s) in the network. Efficient methods are proposed to perform the probability update, test pool selection and sensor state decoding when the size of the network is large. Results show that the Bayesian Group Testing method uses $40\% - 50\%$ fewer of tests than the Combinatorial Group Testing method in achieving the same high detection rate ($> 80\%$). Moreover, the Bayesian Group Testing method is able to achieve high accuracy when the group test results are noisy but the accuracy of other existing divide-and-conquer-based adaptive group testing methods deteriorate significantly.

CHAPTER VIII

Conclusions

This thesis studies efficient sensor fault diagnosis algorithms because wireless sensor networks are energy scarce and contain a large number of sensors. This study proposes two efficient sensor fault detection and identification frameworks. The first framework is a distributed model-based framework which fills the existent gap between the centralized model-based design and the distributed model-free design. It has benefits of low communication energy consumption, high accuracy and scalable. Two algorithms are presented under this framework for detecting and identifying, without reference sensors, spike and non-linearity faults. This study also conducts a field study in which the spike faults detection and identification algorithm is implemented on a real wireless sensors and deployed on a bridge. This field study not only verifies the fault detection algorithm's performance under real deployment environment, it also investigates related network issues such as power consumption, computation time, and network partition methods. The second framework is a group testing-based sensor fault detection method which is suitable for networks with rare fault sensors, *i.e.*, the number of faulty sensors is much fewer than the size of the network. Two group testing-based algorithms, one non-adaptive the other adaptive, are proposed under this framework for fast detection of faulty sensors.

The proposed distributed model-based framework partitions the sensor network into sensor pairs, and each pair of sensors is modeled by a linear ARX model. The ARX model acts as a reference for fault detection. Any discrepancies between the sensor measurements and this ARX model indicate the existence of sensor faults. By showing that the discrepancies inherit the characteristics of sensor faults, specific algorithms are constructed for identifying spike faults and non-linearity faults. The spike faults are detected and identified by Match filters which are based on ARX coefficients. The non-linearity faults detection and identification problem is solved by converting it to the Largest Empty Rectangle problem.

The second framework, the proposed group testing framework, reduces the number of tests by evaluating an entire group of sensors at a time. The proposed non-adaptive method (CGT method) consists of a Combinatorial Group Testing method (which designs the test pool) and a Kalman filter based group evaluation method (which indicates whether a test pool contains any faulty sensors). This method is further improved by iteratively designing the test pool based on the available test results. The improved method calculates the probability of each sensor being faulty after each group test by Bayesian inference, and thus is called the Bayesian Group Testing (BGT) method. The BGT method requires fewer group tests when compared to the CGT method and is more suitable for noisy group test systems. The BGT method also assumes conditional independence on the Bayesian inference update to reduce computation complexity

Experiments under simulated and real faulty sensor data show that algorithms under the distributed model-based framework are able to obtain over 85% accuracy under typical sensor fault conditions. These distributed methods, with the proposed network partition method, are also shown to save more than 60% of communication energy when compared to centralized methods. For the group testing-based methods, experiment results show that both the CGT and BGT methods are able to achieve similar and sometimes higher accuracy as other Kalman filter-based sensor fault detection methods while using fewer tests. The BGT method performs significantly better under noisy group testing systems when compared to other adaptive group testing methods.

For the distributed model-based framework, although the discrepancies between the sensor measurements and the ARX model consist of the sensor fault characteristics, different algorithms, which may be sophisticated, are needed for different types of faults. Therefore, a further study is suggested to develop identification algorithms for other fault types, such as drift faults. The group testing-based methods are semi-distributed methods. An important future works would develop a fully distributed version such that the communication energy requirement is reduced. This study assumes the physical system under monitoring is functioning normally during the sensor fault diagnosis process, therefore discrepancies are assumed to be caused by the faulty sensors only. Hence, another future work would develop a fault detection algorithm that is able to distinguish between system faults and sensor faults.

BIBLIOGRAPHY

BIBLIOGRAPHY

Akyildiz, I. F., W. Su, Y. Sankarasubramaniam, and E. Cayirci (2002), Wireless sensor networks: a survey, *Computer networks*, *38*(4), 393–422.

Allemann, A. (2003), Improved upper bounds for several variants of group testing, Ph.D. thesis, Universitätsbibliothek.

Atia, G., and V. Saligrama (2012), Boolean compressed sensing and noisy group testing, *Information Theory, IEEE Transactions on*, *58*(3), 1880 –1901, doi:10.1109/TIT.2011.2178156.

Bishop, C. (2006), *Pattern recognition and machine learning*, vol. 4, springer New York.

Blough, D., G. Sullivan, and G. Masson (1989), Fault diagnosis for sparsely interconnected multiprocessor systems, in *Fault-Tolerant Computing, 1989. FTCS-19. Digest of Papers., Nineteenth International Symposium on*, pp. 62–69, IEEE.

Bokareva, T., W. Hu, S. Kanhere, B. Ristic, N. Gordon, T. Bessell, M. Rutten, and S. Jha (2006), Wireless sensor networks for battlefield surveillance, in *Proceedings of the land warfare conference.*

Borner, M., M. Zele, and R. Isermann (2001), Comparison of different fault detection algorithms for active body control components: Automotive suspension system, in *American Control Conference, 2001. Proceedings of the 2001*, vol. 1, pp. 476–481, IEEE.

Bremaud, P. (1999), *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*, vol. 31, springer.

Broen, R. (1974), A nonlinear voter-estimator for redundant systems, in *Decision and Control including the 13th Symposium on Adaptive Processes, 1974 IEEE Conference on*, vol. 13, pp. 743–748, doi:10.1109/CDC.1974.270532.

Chen, J., and R. J. Patton (2012), *Robust model-based fault diagnosis for dynamic systems*, Springer Publishing Company, Incorporated.

Chen, J., S. Kher, and A. Somani (2006), Distributed fault detection of wireless sensor networks, in *DIWANS, Proceedings of*, pp. 65–71.

Cho, S., and A. Chandrakasan (2001), Energy efficient protocols for low duty cycle wireless microsensor networks, in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 4, pp. 2041 –2044 vol.4, doi:10.1109/ICASSP.2001.940392.

Cover, T. M., and J. A. Thomas (2012), *Elements of information theory*, John Wiley & Sons.

Da, R., and C. Lin (1995), Sensor failure detection with a bank of kalman filters, in *American Control Conference, 1995. Proceedings of the*, vol. 2, pp. 1122–1126, IEEE.

Ding, M., D. Chen, K. Xing, and X. Cheng (2005), Localized fault-tolerant event boundary detection in sensor networks, in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 2, pp. 902 – 913 vol. 2, doi:10.1109/INFCOM.2005.1498320.

Dorfman, R. (1943), The detection of defective members of large populations, *The Annals of Mathematical Statistics*, *14*(4), 436–440.

Du, D. Z., and F. Hwang (1993), *Combinatorial group testing and its applications*, World Scientific.

Dunia, R., S. Qin, T. Edgar, and T. McAvoy (2004), Identification of faulty sensors using principal component analysis, *AIChE Journal*, *42*(10), 2797–2812.

Estrin, D., D. Culler, K. Pister, and G. Sukhatme (2002), Connecting the physical world with pervasive networks, *Pervasive Computing, IEEE*, *1*(1), 59–69.

Frank, P. (1990), Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results, *Automatica*, *26*(3), 459–474.

Gilbert, A., B. Hemenway, A. Rudra, M. Strauss, and M. Wootters (2012), Recovering simple signals, in *Information Theory and Applications Workshop (ITA), 2012*, pp. 382 –391, doi:10.1109/ITA.2012.6181772.

Goodrich, M., and D. Hirschberg (2006), Efficient parallel algorithms for dead sensor diagnosis and multiple access channels, in *Proceedings of the eighteenth annual ACM symposium on Parallelism in algorithms and architectures*, pp. 118–127, ACM.

Guo, S., Z. Zhong, and T. He (2009), Find: faulty node detection for wireless sensor networks, in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pp. 253–266, ACM.

Gutiérrez, G., and J. Paramá (2012), Finding the largest empty rectangle containing only a query point in large multidimensional databases, in *Scientific and Statistical Database Management*, pp. 316–333, Springer.

Hwang, F. (1972), A method for detecting all defective members in a population by group testing, *Journal of the American Statistical Association*, *67*(339), 605–608.

Isermann, R. (1997), Supervision, fault-detection and fault-diagnosis methods–an introduction, *Control Engineering Practice*, *5*(5), 639–652.

Ji, S., Y. Xue, and L. Carin (2008), Bayesian compressive sensing, *Signal Processing, IEEE Transactions on*, *56*(6), 2346–2356.

Jiang, L. (2011), Sensor degradation detection and isolation using system dynamics identification techniques, Ph.D. thesis, University of Michigan.

Jiang, L., D. Djurdjanovic, and J. Ni (2007), A new method for sensor degradation detection, isolation and compensation in linear systems, in *ASME 2007 International Mechanical Engineering Congress and Exposition*, pp. 1089–1101, American Society of Mechanical Engineers.

Kaplan, H., and M. Sharir (2011), Finding the maximal empty rectangle containing a query point, *arXiv preprint arXiv:1106.3628*.

Katayama, T. (2005), *Subspace methods for system identification*, Springer Verlag.

Kobayashi, T., and D. Simon (2003), Application of a bank of kalman filters for aircraft engine fault diagnostics, *Tech. rep.*, DTIC Document.

Koushanfar, F., M. Potkonjak, and A. Sangiovanni-Vincentelli (2003), On-line fault detection of sensor measurements, in *Sensors, 2003. Proceedings of IEEE*, vol. 2, pp. 974–979, IEEE.

Kurata, M., J. Kim, J. Lynch, G. v. d. Linden, H. Sedarat, E. Thometz, P. Hipley, and L.-H. Sheng (2012), Internet-enabled wireless structural monitoring systems: Development and permanent deployment at the new carquinez suspension bridge, *Journal of Structural Engineering*, *139*, 1688–1702.

Lennart, L. (1999), System identification: theory for the user, *PTR Prentice Hall, Upper Saddle River, NJ*.

Li, Z., B. Koh, and S. Nagarajaiah (2007), Detecting sensor failure via decoupled error function and inverse input–output model, *Journal of engineering mechanics*, *133*(11), 1222–1228.

Lipták, B. G. (2005), *Instrument Engineers' Handbook, Volume Two: Process Control and Optimization*, vol. 2, CRC press.

Liu, J., D. Djurdjanovic, K. A. Marko, and J. Ni (2009), A divide and conquer approach to anomaly detection, localization and diagnosis, *Mechanical Systems and Signal Processing*, *23*(8), 2488–2499.

Luo, X., M. Dong, and Y. Huang (2006), On distributed fault-tolerant detection in wireless sensor networks, *Computers, IEEE Transactions on*, *55*(1), 58–70.

Lynch, J. (2007), An overview of wireless structural health monitoring for civil structures, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *365*(1851), 345–372.

Malloy, M. L., and R. D. Nowak (2012), Near-optimal adaptive compressed sensing, in *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*, pp. 1935–1939, IEEE.

Marti, S., T. Giuli, K. Lai, M. Baker, et al. (2000), Mitigating routing misbehavior in mobile ad hoc networks, in *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 255–265.

Maybeck, P. (1979), *Stochastic Models, Estimation, and Control: Vol.: 1*, Academic Press.

Monden, Y., M. Yamada, and S. Arimoto (1982a), Fast algorithm for identification of an arx model and its order determination, *Acoustics, Speech and Signal Processing, IEEE Transactions on*, *30*(3), 390–399.

Monden, Y., M. Yamada, and S. Arimoto (1982b), Fast algorithm for identification of an arx model and its order determination, *Acoustics, Speech and Signal Processing, IEEE Transactions on*, *30*(3), 390–399.

Murty, K., and C. Perin (1982), A 1-matching blossom-type algorithm for edge covering problems, *Networks*, *12*(4), 379–391.

Nandy, S., B. Bhattacharya, and S. Ray (1990), Efficient algorithms for identifying all maximal isothetic empty rectangles in vlsi layout design, in *Foundations of Software Technology and Theoretical Computer Science*, pp. 255–269, Springer.

Pan, J., and W. Levine (1990), A fast order-recursive algorithm for toeplitz submatrix systems with applications to estimation of arx systems, in *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*, pp. 2879–2885, IEEE.

Paradis, L., and Q. Han (2007), A survey of fault management in wireless sensor networks, *Journal of Network and Systems Management*, *15*(2), 171–190.

Percival, D. (1993), *Spectral analysis for physical applications*, Cambridge University Press.

Proakis, J. G. (2007), *Digital Communiations 5th edition*, McGraw-Hill.

Ramanathan, N., K. Chang, R. Kapur, L. Girod, E. Kohler, and D. Estrin (2005), Sympathy for the sensor network debugger, in *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05, pp. 255–267, ACM, New York, NY, USA.

Ricquebourg, V., D. Menga, M. Delafosse, B. Marhic, L. Delahoche, and A. Jolly-Desodt (1991), Sensor failure detection within the tbm framework: A markov chain approach, in *Proceedings of IPMU*, vol. 8, p. 323.

Ricquebourg, V., D. Durand, D. Menga, B. Marine, L. Delahoche, C. Loge, and A.-M. Jolly-Desodt (2007), Context inferring in the smart home: An swrl approach, in *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*, vol. 2, pp. 290–295, IEEE.

Sikka, P., P. Corke, and L. Overs (2004), Wireless sensor devices for animal tracking and control, in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pp. 446 – 454, doi:10.1109/LCN.2004.141.

Staddon, J., D. Balfanz, and G. Durfee (2002), Efficient tracing of failed nodes in sensor networks, in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pp. 122–130, ACM, New York, NY, USA.

Straser, E. G., and A. S. Kiremidjian (1998), *A modular, wireless damage monitoring system for structures*, John A. Blume Earthquake Engineering Center Stanford.

Swartz, R., A. Zimmerman, J. Lynch, J. Rosario, T. Brady, L. Salvino, and K. Law (2012), Hybrid wireless hull monitoring system for naval combat vessels, *Structure and Infrastructure Engineering*, *8*(7), 621–638.

Tarjan, R. (1975), Efficiency of a good but not linear set union algorithm, *Journal of the ACM (JACM)*, *22*(2), 215–225.

Tong, L., and S. Perreau (1998), Multichannel blind identification: From subspace to maximum likelihood methods, *Proceedings of the IEEE*, *86*(10), 1951–1968.

Tosic, T., and P. Frossard (2012), Distributed group testing detection in sensor networks, in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 3097–3100, IEEE.

Trefethen, L., and D. Bau III (1997), *Numerical linear algebra*, 50, Society for Industrial Mathematics.

Venkatasubramanian, V., R. Rengaswamy, K. Yin, and S. N. Kavuri (2003), A review of process fault detection and diagnosis: Part i: Quantitative model-based methods, *Computers & chemical engineering*, *27*(3), 293–311.

Wark, T., P. Corke, P. Sikka, L. Klingbeil, Y. Guo, C. Crossman, P. Valencia, D. Swain, and G. Bishop-Hurley (2007), Transforming agriculture through pervasive wireless sensor networks, *Pervasive Computing, IEEE*, *6*(2), 50 –57, doi:10.1109/MPRV.2007.47.

Weiser, M. (1993), Some computer science issues in ubiquitous computing, *Communications of the ACM*, *36*(7), 75–84.

Xu, X., J. Hines, and R. Uhrig (1999), Sensor validation and fault detection using neural networks, in *Proc. Maintenance and Reliability Conference (MARCON 99)*, pp. 10–12, Citeseer.