Information Extraction from Undersampled and Asynchronous Vehicle Sensor Data

Submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Andrew S. Fox

B.Sc., Engineering Physics, Queen's University
M.S., Electrical and Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA 15213

May, 2016

Thesis Committee Members

Prof. Vijayakumar Bhagavatula (Advisor)
Department of Electrical and Computer Engineering
Carnegie Mellon University

Prof. Hae Young Noh
Department of Civil and Environmental Engineering
Carnegie Mellon University

Prof. Anthony Rowe
Department of Electrical and Computer Engineering
Carnegie Mellon University

Dr. Fan Bai
Electrical and Control Integration Laboratory, Research and Development and Planning
General Motors Corporation

Abstract

As smart-vehicles, capable of advanced sensing and automated control procedures, become more prevalent in the Intelligent Transportation System (ITS), there is a need for next-generation road maps that contain all relevant environmental information that may assist drivers, passengers, and other stakeholders connected to the ITS. The proliferation of sensor-equipped consumer vehicles with dedicated communications systems has provided a valuable resource for continuous mobile data collection from which to extract such information.

There are however, a number of challenges associated with such an information extraction process. Since vehicles and road environments are heterogeneous, there are many different possible sensor signals that could indicate an event. Sensor measurements may have significant error, which is often correlated within vehicles, particularly for measurements indexed by the Global Positioning System (GPS). Due to functional constraints of sensors, the measurements are asynchronously collected, and the signal-of-interest is often undersampled, requiring data to be aggregated from multiple vehicles to acquire a sufficient data set.

In this thesis, we develop a vehicle-Cloud detection framework to extract environmental information from such aggregated, undersampled, and asynchronous vehicle data. We introduce the noisy multi-source, variable-rate (MSVR) sampling model with correlated errors in variables, and derive error models based on the vehicle and GPS sampling conditions.

Road environmental information extraction algorithms are developed for the vehicle MSVR sampling conditions, specifically for the detection of continuous and binary types of road information. Within the overall vehicle-Cloud detection framework, algorithm adaptations are developed to detect events in multi-lane environments, filter data to reduce the required network bandwidth, account for temporally changing information, apply side information from other events, and use data-driven metrics to optimize the algorithm parameters.

This framework is applied to detect road incline and bank angle information, and pothole locations on multi-lane roads. These algorithms are developed specifically for the MSVR sampling environment using only GPS and accelerometer data. Results are analyzed for sets of both simulated and real-world data, examining the tradeoffs between the number of aggregating vehicles and detection accuracy, in addition to the effects of the data filters and parameter optimizations developed in the overall detection framework.

Acknowledgments

I first want to acknowledge all the work of my advisor, Prof. Vijayakumar Bhagavatula, without whom this document certainly would not exist. I do not know how many Ph.D students' graduate school tenures follow any sort of an initial plan, however the pathway I took was certainly not the one I envisioned taking six years ago. I am not sure what it was that convinced you to initially take me on as a student, but I am forever grateful for the opportunity. Your ability to take something I had been working on for months and upon hearing the idea for the first time, reiterate the concept with perfect clarity and detail, is a talent unmatched by anyone. Your patience for disjointed explanations and grammatical errors has certainly been appreciated over these years. I have no doubt that if you wrote this acknowledgment instead of me that it would much more eloquently and honestly express my feelings. Your insight and guidance have served me greatly, and I again want to truly express my gratitude for everything you have done.

Our collaborator at General Motors, Dr. Fan Bai, is one of the most dedicated people I have ever had the opportunity of meeting. Fan, you have been extremely generous with your time and your ability to spin a phrase to perfectly capture a reviewer's attention is nothing short of amazing. I want to sincerely thank you for all the time and resources you have devoted to helping to progress this research.

I want to thank the other members of my thesis committee, Prof. Hae Young Noh and Prof. Anthony Rowe, for their interest in my research and helping to guide this document.

The funding for this work was provided by General Motors (GM) through the GM-

CMU Vehicle Information Technology (VIT) Collaborative Research Lab (CRL) and by the Technologies for Safe and Efficient Transportation (T-SET) University Transportation Center (UTC). I would also like to thank the Carnegie Mellon University Electrical and Computer Engineering (ECE) department for the support they have provided.

I want to thank Dr. Jinzhu Chen at General Motors and Dr. Christoph Mertz at the CMU T-SET UTC for providing us with real-world data sets.

I want to acknowledge two group members specifically. Dr. Joseph Fernandez, you helped introduce me to both the softball team and Prof. Kumar. Those were without a doubt two of the most important things to have happened to me in the last six years (in some order). I also want to thank Stephen Siena, who always knew exactly how to trigger a rant to break up a monotonous day.

A mere acknowledgment is not enough for former group members Dr. Andrew Turner and John Payne. Your friendship helped keep me somewhat maybe almost close to sane throughout all the work we accomplished.

I want to acknowledge the other group members whose genius I have had the fortune of witnessing up close over these past six years: Dr. Akkarit Sangpetch and Dr. Orathai Sukwong, who managed to keep their laughter to a minimum as they taught me how computers work. Dr. Jonathon Smereka, Dr. Katharine Brigham, and Eric He, who helped welcome me to Prof. Kumar's group. I want to also thank research group members Dr. Kyriaki Levanti, Dr. Huiguang Liang, Tiago Carvalho, Chen Wang, Senbo Fu, Yongjune Kim, Zhiding Yu, Dr. Hyunggi Cho, and Rick Chang. Your presence and curiosity have provided me with constant inspiration, and it has been an honor to work in some capacity with all of you.

I want to thank all the members of the ECE intramural teams, especially Gigahurtz and the Fast Floorier Transforms. The time spent playing sports with you has been one of my most cherished memories of my time at CMU.

I particularly need to thank all my friends in Toronto, Pittsburgh, and beyond. You have never failed to provide me with a distraction any time I wanted one.

I want to acknowledge my siblings, Mitchell, Shannon, and Morgan, all of whom I have not seen nearly enough of over the past ten years. Any opportunity that we can gather more than one of us into a room together is a true pleasure.

Finally, I would like to thank my parents, Rick and Anna Fox. It has been an interesting journey getting to this point, and your love and support have been the true constants the entire time. You have provided every opportunity possible for Mitchell, Shannon, Morgan, and myself, and I hope that this in some small way acknowledges how truly grateful I am.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

As automated vehicle control systems become more prevalent in consumer vehicles, there is a need for next-generation road maps that contain an inventory of all relevant environmental information that may affect driving behavior. Existing maps currently display road location information and often relative traffic densities or predicted travel times, however there is further information such as potholes, traffic lights, road inclination angles, crosswalks, speedbumps, etc., whose knowledge could be used for the benefit of drivers, automated systems, and eventually fully autonomous vehicles.

In the past, detecting such environmental information relied on either driver self-reporting or specialized event-specific detection equipment. These are expensive approaches that lack the wide coverage required to keep pace with dynamic conditions. However, recent years have witnessed significant proliferation in the number of sensor-equipped, Internet-connected vehicles in the Intelligent Transportation System (ITS). The embedded sensors and dedicated communications systems in consumer vehicles provide a resource for continuous mobile data collection during drivers' ordinary commutes. With consumer vehicles acting as mobile sensor agents, the capability exists to enable wide-spread and current

sensing of road conditions. By aggregating the sensed data from these vehicles, a system can be implemented to extract any relevant environmental information.

However, there are a number of challenges that exist when attempting to extract information from aggregated vehicle data:

- Vehicles do not directly sense most types of road information, e.g., there is no explicit "pothole" sensor within the vehicle sensor suite. For most types of information, a detector or mapping system needs to be directly designed and trained using the data from conventional sensors monitoring the vehicles' dynamics (e.g., accelerometers). Although training procedures may be generalized for an entire detection framework, each event-specific information extraction procedure will require its own unique set of algorithms and application of existing system knowledge in order to design the best-performing detector.
- Vehicles, sensors, and detection scenarios are heterogeneous. Therefore the measured sensor data in each vehicle varies when experiencing a shared event. Any detection process needs to account for the various possible sensor signals that may exist to signify an event. Since it is difficult to design a customized detector for every unique scenario, the detection processes therefore need to be designed to be generally applicable.
- Environmental information falls into a number of different categorizations. Some information, such as the physical location of the road, is relatively static or unchanging, while other information such as traffic density is highly dynamic. Similarly, some information could be denoted by a binary variable indicating whether or not an event occurs at that location (e.g., is there a pothole at a given location?), while other information is better represented by continuous functions where any value could exist for a given location (e.g., road inclination angle). These factors influence how a detection procedure needs to be designed and in what form the results should be

output.

- There may be significant noise or error in the sensor output from different vehicles. Due to the nature of the sensor measurement systems, the error on consecutive measurements may not be independent. This is particularly important when considering Global Positioning System (GPS) measurements. Measured GPS locations spatially index the measurement samples from other vehicle sensors. However, GPS may have significant error associated with it and it is often strongly correlated over relatively short times and distances. This error needs to be properly accounted for in the detection systems. Many traditional event detection or signal processing algorithms are ill-equipped to account for correlated location error. Therefore any information extraction algorithm designed for vehicle sensor data as an input needs to be constructed to consider these sampling conditions.

1.2 Problem Statement and Goal

As the number of sensor-equipped, Internet-connected, smart vehicles have increased in the Intelligent Transportation System, the vehicles' sensor data has become a valuable resource from which to extract useful environmental information. This information is essential for the development of next-generation road maps, rich with current and vehicle-relevant features.

Extracting the environmental information necessitates the creation of a vehicle-Cloud framework to aggregate the vehicles' sensor data and extract the required information. In this thesis, we develop this framework and investigate detecting both binary and continuous types of road event information. The framework and associated detection algorithms are specifically designed to handle the undersampled and asynchronous sensing conditions that exist for embedded vehicle sensors. Within this context, we address problems of vehicle and environment heterogeneity, and correlated GPS error measurements that corrupt event

detection and signal reconstruction.

We evaluate our detection framework both from a theoretical aspect on the signal reconstruction methods, and on specific example event detection systems, those being pothole and road topography information. We analyze our detection capabilities in terms of the number of vehicles involved in the detection process, and how bandwidth may be conserved for the ITS conditions.

1.3 Contributions of the Thesis

1.3.1 Detection Framework and Signal Reconstruction

In Chapter 4 we develop a framework to extract information regarding both continuous and binary types of road events from asynchronous and undersampled vehicle sensor data. The framework is designed to specifically account for the unique sampling conditions of the road environment and the interaction of the various types of events or information within the Intelligent Transportation System. These unique sampling conditions are introduced in Chapter 3, where error models are derived for the types of correlated error inherent to the road network sampling conditions. We derive the information extraction algorithms, investigate their properties, develop adaptations for temporally changing information, and analyze their applicability to different types of vehicle sensor data and desired road information. Although the framework is applied to the Intelligent Transportation System, the detection framework and derived signal reconstruction algorithms are applicable to more general situations where undersampled or asynchronous sampling conditions necessitate aggregating sensor data from multiple sources to detect or reconstruct information.

1.3.2 Road Angle Information

In Chapter 5, an algorithm is developed to determine the bank and incline angles of the road from noisy and undersampled vehicle acceleration measurements. The algorithm estimates the angles for each individual data sample before transmitting the samples to the Cloud for data aggregation where the road angle information signal for the entire road is reconstructed. We demonstrate how the various aspects of the detection algorithm fit within the context of the overall information extraction framework.

1.3.3 Pothole Detection

In Chapter 5, we demonstrate how the information extraction framework is applied to detecting potholes from asynchronous and undersampled vehicle accelerometer data. We apply the detection algorithm to multi-lane roads to attempt to finely localize the pothole positions while developing filters to minimize the required transmission bandwidth of the system. We evaluate the framework's detection capabilities under these constraints.

1.3.4 Publications

Work in this dissertation has resulted in the following peer-reviewed publications:

1. A. Fox, B. V. K. Vijaya Kumar, J. Chen, and F. Bai, "Crowdsourcing undersampled vehicular sensor data for pothole detection," in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (IEEE SECON 2015)*, Seattle, USA, Jun. 2015, pp. 515–523.
2. A. Fox, B. V. K. Vijaya Kumar, and F. Bai, "Multi-source variable-rate sampled signal reconstructions in vehicular CPS," in *The 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016)*, San Francisco, USA, Apr. 2016, pp. 946–954

3. A. Fox, V. Sharma, and B. V. K. Vijaya Kumar. "Signal Reconstruction for Multi-source Variable-rate Samples with Autocorrelated Errors in Variables", in *2016 IEEE Statistical Signal Processing Workshop (SSP 2016)*, Palma de Mallorca, Spain, Jun. 2016.

At the time of publication, the following work is also under review:

1. A. Fox, B. V. K. Vijaya Kumar, J. Chen, and F. Bai, "Multi-lane Pothole Detection from Crowdsourced Undersampled Vehicle Sensor Data," in *Mobile Computing, IEEE Transactions on (TMC), Under Review.*

1.4 Outline of the Document

The remainder of the dissertation is organized as follows: Chapter 2 presents a high-level overview of the vehicle-Cloud information extraction framework. Important parameters and interactions within the framework are introduced, and relevant variables are discussed with regard to their effect on the framework design. Chapter 3 presents the noisy multi-source variable-rate (MSVR) sampling model appropriate for vehicle sensing systems. Error models for vehicle and GPS measurements are presented along with their effects on the measurements. Chapter 4 presents detailed frameworks for how to detect continuous and binary types of environmental information within the vehicle-Cloud framework. Chapter 5 applies these frameworks to the detection of specific types of information. Conclusions for this thesis research are provided in Chapter 6.

Chapter 2

Vehicle-Cloud Detection Framework

2.1 Introduction

The state of technology in consumer vehicles has advanced such that vehicles are now equipped with a wide range of sensors to measure the surrounding road environment and the vehicles' physical states. All vehicles sold in the United States after 2008 are required to implement the controller area network (CAN) standard, and vehicles can have up to 70 electronic control units (ECUs) that communicate using CAN protocols [1]. The data from sensors on the CAN bus can be accessed using the On-Board Diagnostic (OBD-II) port, which has been mandatory in vehicles since 1996 [2]. These sensors provide a mechanism to obtain an instantaneous snapshot of the vehicle state and surrounding environment.

Keeping pace with other advances, the developments in vehicular communications systems and Cloud technology have allowed for data to be shared between vehicles and aggregated for centralized processing. Since consumer vehicles continually cover a wide range of the road network during ordinary commutes, data aggregation allows for analyses of wide-ranging data for the entire Intelligent Transportation System (ITS). These technologies allow one to use the vehicles' sensor data regarding their physical states to reverse engineer a system to extract information about the road environmental conditions that caused

the vehicles to be in the respective measured states. Designing such a system effectively crowdsources the environmental information from the existing fleet of sensors in mobile consumer vehicles. Once determined, the extracted environmental information could subsequently be transmitted back to the vehicles for the benefit of drivers or automated control systems.

In this thesis work, we develop a general framework to extract road environmental information from vehicle sensor data. In this chapter, we provide a general overview of the information extraction process and highlight how the various aspects of the detection framework are designed to handle heterogeneous vehicle and road conditions. This will be approached from a high-level, justifying the reasons for the framework. Subsequent chapters will detail specific algorithms and their adherence to the framework (Chapter 4) and how those algorithms are applied to detecting specific types of environmental information (Chapter 5).

In this thesis, the term "information" will refer to the high-level concept that is attempted to be determined, e.g., information about a traffic light. The term "event" refers to the process of a vehicle or multiple vehicles engaging or becoming under the influence of the desired road information. For example, a vehicle experiences a traffic light event as it slows to a stop at a red light. Often the terms "information extraction" or "event detection" will be used interchangeably to describe the process of determining the desired information. The term "data" is used to refer to the input to any event detection algorithm. Depending on the algorithm, this data could include raw sensor measurements, or processed sensor measurements used as an input to a secondary algorithm. For example, accelerometer data is an input to a road incline angle detection algorithm, however the output angles could be used as the input data to a pothole detection algorithm.

A block diagram of the overall information extraction framework is shown in Figure 2.1. This framework features continual feedback and exchange of data and information between

vehicles and the Cloud so that the raw data and any extracted information can be best applied to the appropriate algorithmic element. The following sections expand on the different components of the framework. For different types of events or information (Section 2.2), vehicles first collect their own sensor data and potentially perform some initial processing (Section 2.3) where they either attempt to fully detect the event, or perform some form of filtering or data manipulation so that the data is in an appropriate format for aggregation. Depending on the event of interest, either the raw vehicle sensor data, preprocessed data, or vehicle-extracted information is transmitted to the Cloud, thereby aggregating data from multiple vehicles (Section 2.4). The Cloud offers significantly more computational and storage capacities than individual vehicles, as well as access to a broader set of data. These capabilities allow for algorithms to be deployed in the Cloud to extract environmental information from the aggregated data by methods that individual vehicles are incapable of executing (Section 2.5). Within the detection process, both in vehicles and on the Cloud, it is important to consider how the information extracted from one type of detection algorithm may be used as side or additional information for detecting other events (Section 2.6). It is also important to consider how any algorithmic parameters are determined (Section 2.7) or how the detection algorithm is trained and developed (Section 2.8) for both the vehicle and Cloud detection processes. Any information detected within the Cloud should then subsequently be transmitted back to the vehicles, completing the feedback loop. Similar to the side detection process, this Cloud information can be used in the vehicles as an aid for detecting other types of information.

2.2 Types of Environmental Information

The nature of the environmental information to be detected determines how various aspects of the detection framework need to be adapted to function properly in the applicable domain. In order to customize the detection framework configuration to a specific type

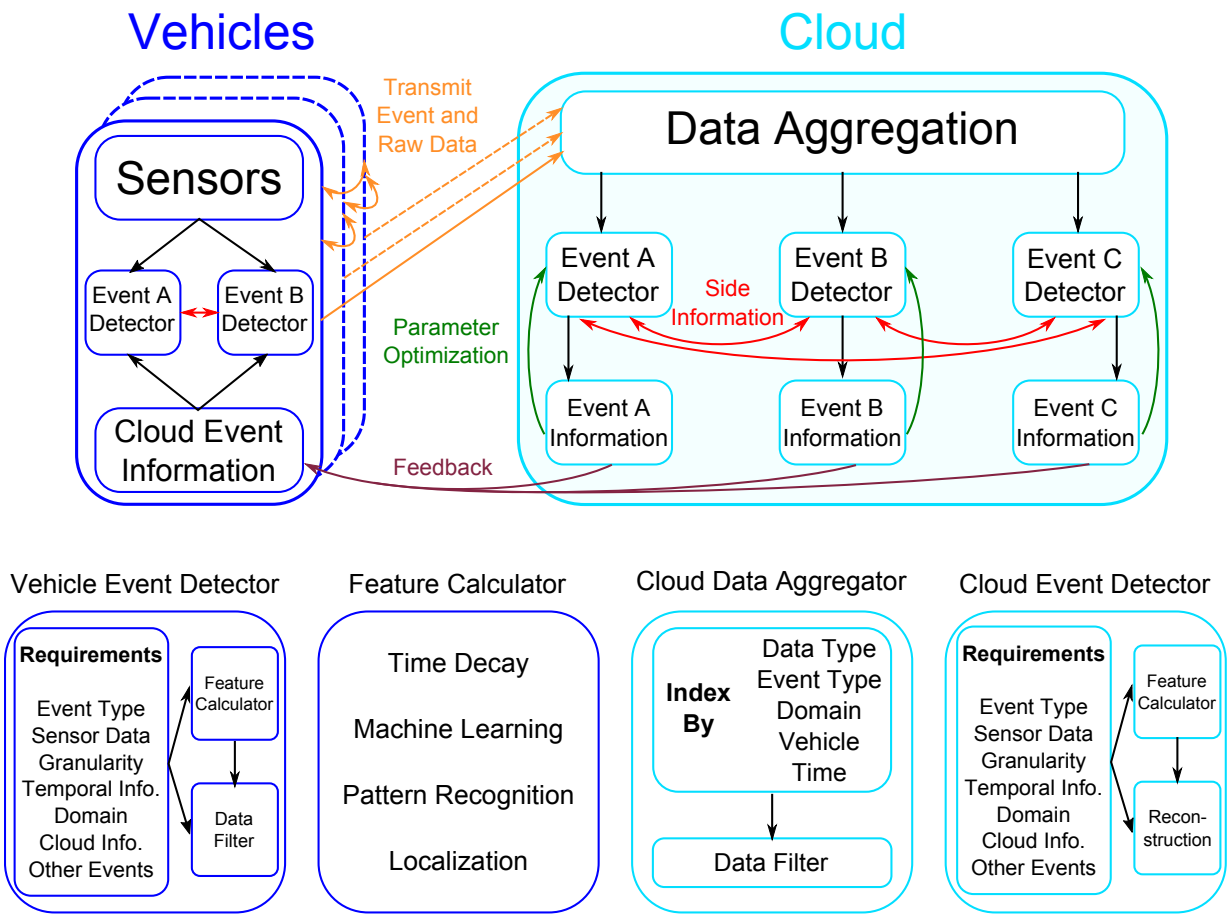


Figure 2.1: Framework of Vehicle-Cloud information extraction system

Table 2.1: Example types of environmental information, classified by spatial and temporal granularity

	Static or Slowly-changing Information	Dynamic or Quickly-changing Information
Binary Information	• Pothole [3] • Speedbump [4] • Train crossing • Stop sign and traffic lights [6] • Crosswalks [7] • Traffic sign recognition [8]	• Slippery or icy road • Construction zone • Poor visibility [5] • Rain • Roadkill • Accidents [9] • Road-side parking availability [10]
Continuous Information	• Road incline and bank angle [11] • Number of lanes [12] • Speed limit [13]	• Traffic density • Ambient temperature • Pedestrian density • Fuel Efficiency Map [14]

of environmental information, information regarding the following elements needs to be specified:

Event Type

The *event type* is the specific event or piece of environmental information to be detected. This high level concept acts as the ground-truth for comparison and provides the reference to determine features to be used in any algorithm. Examples of different types of events to be detected are shown in Table 2.1, indexed by spatial and temporal granularity (to be discussed in the following). Parameters for the subsequent component requirements depend on the physical nature of the event types and their influence on driving behavior.

Sensor Data

It is important to identify the types of sensor data that are most appropriate for extracting the desired environmental information. The *sensor data* specifies the data source for the detection algorithms. The data could be chosen to fit into a machine learning framework, or there may be a more fundamental, perhaps physics-based reason, for utilizing the sensor data. We demonstrate examples of both situations with pothole and road angle detection, respectively, in Chapter 5. The chosen data also specifies how the detection process for a specified event may interact with other event detection processes. If two different detection processes require the same raw data to be uploaded to the Cloud, the processes should be indexed so that data is transmitted only once and there is no redundancy within the system (unless required for security or error-checking purposes). The type of data also specifies what types of machine learning or signal processing techniques may be applied to the data. Due to controller area network (CAN) bus limitations [15], different sensors operate asynchronously and at different frequencies. Many sensors operate at only approximately 1 Hz, which undersamples many road signals. Any undersampled sensor data would

therefore limit the capabilities of detecting events within the vehicle and may necessitate aggregating data from multiple vehicles onto the Cloud to increase the effective sampling rate to an adequate level for detection.

Spatial Granularity

The *spatial granularity* of a road feature can be examined at multiple scales. At a high level we can separate events into two categories: binary features and continuous features. Binary features are features that could be described by a binary value as either existing or not existing at a given location. A speedbump is one such example. Continuous features are those best described by continuous functions, often as a function of the road location. These features have varying values. One such example of a continuous feature is the road incline angle, which would be a continuous function along a road. Spatial classifications of other example road events are shown in Table 2.1. There are also finer-grained specifications of events beyond just binary or continuous classification. For binary road events for example, localizing the event to an exact location is important. Most road events occupy some region of space (e.g., $\sim$1 m for a pothole), however since the detection is not necessarily done in a continuous domain, the event may be detected only in some specified window(s) depending on the nature of the detection algorithm. It is beneficial to more precisely localize an event. The desired granularity affects the parameters of the detection process. For example, the parameters or amount of data required to localize a speedbump to within 0.1 m vastly differ from being able to localize a speedbump to within a 10 m range. Specifying the required accuracy is important to the detection process. Algorithms for the detection of continuous information are given in Section 4.1 and for binary events in Section 4.2.

Temporal Information

Different types of road information are associated with different timescales, which

dictates how a detection system should value dynamically changing *temporal information*. Events could abruptly change (e.g., a pothole being filled), dynamically change on short timescales (e.g., traffic density), or remain static for long periods of time (e.g., road incline angle information). The lifespan or time constant associated with an event dictates how the information extraction process collects new information and factors in old information. For static processes, sensor data may not need to be continuously collected after the event has been confidently detected as the new data can not provide any updated information. In comparison, highly dynamic processes need to continually collect new data to reexamine the current environment, as well as using previous data to build up a history and sample size to predict how the environment evolves. There are intermediate procedures as well where after an event is confidently detected, the sampling rate could be significantly reduced and new data is used to verify the existing information and possibly trigger a change. Depending on the spatial granularity or event type, and therefore the machine learning or signal processing algorithm used in detection, the method by which timing information is accounted for may vary. An algorithm for discounting old information is presented in Section 4.3.

Domain

The information extracted from the vehicles can fit into a number of different *domains*. These domains could include road structure information (e.g., road incline angle, speedbump locations), driving regulations (e.g., speed limits, stop signs), driver or vehicle specific information (e.g., drunk or erratic driver detection), or anomaly information (e.g., poor visibility, potholes). Identifying the domain of the information dictates how it can be used by interested parties, such as drivers or automated vehicle control systems, and what process the information should undergo after it has been detected.

Cloud and Other Event Information

Information from previously detected road features can be critical to helping in the detection of another event. For example, in Section 5.1.3, we demonstrate how known road curvature information can be used to determine the incline and bank angles of the road. Similarly, in Section 5.2.4.1, we demonstrate how these incline and bank angles are used in a pothole detection process. This shared information framework means that the amount of data, or the difficulty of any detection process, can be reduced since information from other sources may assist, or dictate some mutual compatibility in the detection process. Specifying what other information is required is critical to determining the control and data flow paths of the detection framework.

All of these requirements are critical to determining the appropriate features to calculate in order to properly filter the vehicle sensor data and design the final event detection algorithm.

2.3 In-vehicle Event Detection

It would be desirable, if possible, to detect environmental features from the raw sensor data within the vehicle itself. This reduces the need to use bandwidth transmitting raw data to the Cloud, where a large database system would be required to store such extensive data. Performing detection within the vehicle can also mitigate the security and privacy risks of transmitting raw vehicle sensor data.

However, as has been previously stated, the computational capabilities within each vehicle are often restricted due to the limited amount of data that can be collected and the computational infrastructure. These restrictions limit the suite of mathematical algorithms that can be implemented to compute event-discriminating features or the detection of the events themselves. Many road features occupy a small region of space, both in terms of the size of the feature and the range of space over which the feature has measurable

influence on vehicles. A pothole, for example, may only be a meter long, with noticeable aftershocks in the vehicle for about one second. With sensor measurements acquired at only 1 Hz, features such as signal energy or zero crossings are either meaningless or undefined, since there is likely to only be one data sample available for calculations in the region-of-interest. Similarly, processes such as Kalman filters to estimate the vehicle's dynamics may be intractable as the dynamics change at a much higher rate than what the filter is able to estimate.

Therefore any detection in the vehicle is likely to perform worse than a detector that uses aggregated data from multiple vehicles. It was demonstrated in [3] how aggregating detections from the weak detectors in the individual vehicles produced a worse result than designing a detector for all the raw data. This is why it is often beneficial, from the perspective of detection, to aggregate the raw data directly instead of relying on weak detectors in individual vehicles that only have access to their own limited data set. However a certain amount of pre-processing or filtering of the data within the vehicle may be feasible and appropriate. In Section 5.2.4.1, we demonstrate how vehicles can isolate anomalous components of their sensed acceleration data, so that only data that deviates from normal road conditions and could possibly indicate an anomalous road event is transmitted to the Cloud.

In this work, we do not consider the technical details of designing the control and data flow in the vehicular detection system. However this is an important component that dictates many of the constraints of the detection process. Other works have designed specific architectures to consider and compensate for the environmental restrictions and demands [16].

2.4 Data Aggregation

There are two important components in aggregating data from multiple vehicles. The first component describes the communication process by which the data is transmitted to the Cloud, and what form the data takes for a given process (e.g., raw sensor data, preprocessed data, detected event information). This is particularly important given the intermittent connectivity periods [17] and high packet loss rates [18] in vehicular networks. The second component in data aggregation involves how the data is stored and indexed on the Cloud so that it can be used appropriately for any detection algorithm.

Any raw data sent to the Cloud should be indexed by a number of features. Beyond labeling the sensor data by type, there are important additional details to index the data by, including the vehicle from which the samples were obtained, the time the samples were obtained, and the spatial location at which the data was acquired. The samples in the aggregated data set should not be treated independently and ignorant of the knowledge of which vehicle produced them. Vehicles have their own unique behaviors that affect the errors or noise in any measurement. As will be described in Chapter 3, the measuring vehicle is important when considering the GPS error for samples. The GPS errors in the samples from the same vehicle tend to be highly correlated across both time and location. Therefore, using the knowledge that two measurements came from the same vehicle can be used to better estimate and mitigate the impact of error in the detection system.

Timing data is also critical information to be incorporated into the detection process. As described in Section 2.2, different types of information have different time scales associated with them that dictate how the road event evolves over time. It is important to know the age of any data sample to determine its relevance, or how much it should be weighted, for extracting a certain piece of information.

Given the mobile conditions of the vehicular networking environment, there are new protocols developed for Dedicated Short Range Communications (DSRC) [19] or for how

to best store and disseminate event information [20]. However, the exact transmission and storage mechanisms are not the focus of this work. We instead consider their impact on the detection algorithms. Reducing the required bandwidth of a detection protocol is explicitly built into our algorithms, for instance by reducing the amount of transmitted raw data in pothole detection using the data filter described in Section 5.2.4.1. Also, the knowledge of which vehicle each data sample was obtained from is critical in the signal processing algorithms described in Chapter 4, which compensate for and try to estimate the effects of GPS errors. Timing information is also critically considered as part of the detection framework in Section 4.3.

2.5 Cloud Event Detection

Detecting events from aggregated data on the Cloud is beneficial when the individual vehicles are not capable of performing satisfactory detection themselves. As detailed in Section 2.3, this is generally a result of either lacking the computational capabilities to perform the detection task, or lacking the necessary and sufficient data to build up a model extensive enough to detect events. Aggregating data from multiple sources therefore builds up a large enough sample size for training and testing a more comprehensive suite of algorithms. Some examples include inferring road locations and directions from GPS traces [21, 22] or trying to forecast travel times [23] where the behavior of a single vehicle may not be indicative of the true nature of the event or the sensor data may be too noisy for any reliable information extraction.

Performing the information extraction process on the Cloud generally takes two forms, depending on the type of data transmitted from the vehicles. If raw sensor data is transmitted from the vehicles, then the Cloud detection algorithm operates on an aggregated raw data set where the full breadth of measured data is considered. This has the advantage of increasing the effective sampling rate so that if there are certain infrequent behaviors in the

data that can best capture the desired information, the effective sampling rate is increased, resulting in a sample size where it is probable that such behavior can be discovered within the data set. However such a breadth of data means that the heterogeneous behavior of the different vehicles and road conditions may confuse the nature of the features. It may be difficult to determine event-specific behavior if the behavior is not consistent among vehicles. This method also requires transmitting some or all of the raw data, which may be too bandwidth intensive for the vehicular network.

The other example of Cloud detection is where the Cloud aggregates detections made from the vehicles themselves for a final determination. This has the advantage of saving bandwidth since only a location and binary value need to be transmitted from the vehicles to the Cloud. However if the classifiers on the individual vehicles are weak, due to low sampling rates for example, this method is aggregating unreliable detection results. This could result in inaccurate detections over the aggregated data set.

The Cloud framework is particularly helpful for mitigating the effects of GPS noise. Data from different vehicles may conflict due to GPS noise identifying the same event at multiple locations, or by blending the information from different regions onto the same location. However by aggregating the data from multiple vehicles, the detection of the feature itself can be used as a checkpoint to redetermine the likely true position of the measured GPS values. This joint process uses the crowdsourced group knowledge to enhance both the detection process and the quality of the measured data.

The algorithms for Cloud detection also vary depending on the nature of the type of information to extract. A signal processing approach is outlined in Section 4.1 for detecting continuous information, and a machine learning framework is outlined in Section 4.2 for detecting binary road information.

2.6 Side Information

Detection of each individual type of environmental information should not be performed in isolation. The road network is a complex environment with many interacting events. The detection of an event at a certain location could significantly increase the probability of a second event occurring, and exclude a third event altogether. As an example, detecting a 4-way stop at an intersection increases the probability that each of the roads is a bidirectional single lane road, and excludes the possibility that a traffic light would exist at the intersection. There are many similarities in vehicle behavior between stop signs and traffic lights that could be used in a detector for both systems. If the detections are done independently, it could falsely suggest that both a traffic light and a stop sign exist at an intersection. However by considering the existing environmental information in conjunction with the current detection process, a single more confident and correct detection possibility can be determined.

In this thesis work, we focus on applying the detection framework to two specific types of road events, namely detecting the road incline and bank angles, and detecting potholes. In Section 5.2.4.1 and the ensuing results we describe a process for how the incline and bank angles can be used as side information to the pothole detection process to significantly enhance the accuracy of the detection system while also reducing the required transmission bandwidth.

2.7 Parameter Optimization

Although many machine learning or signal processing frameworks attempt to be nonparametric, there are inevitably choices of customizable event-specific features or model-tunable parameters that are necessary in the design of any event detection algorithm. Such parameters could include:

- The number of vehicles or sensor samples required to be aggregated for the detection algorithm
- Function domain length or bandlimit for signal reconstruction
- Time constants or average lifespans of an event
- Thresholds on filters to determine the inclusion of data in an algorithm
- Thresholds on algorithms to determine if data should be transmitted from vehicles to the Cloud
- Inherent algorithmic training parameters (e.g., penalty costs in Support Vector Machine training)
- Expected covariance of error terms

Many of these parameters can be estimated and tuned through repeated trial-and-error processes. However, it is preferable if an automated process is able to select or tune the appropriate parameters. In Section 4.1.4 we discuss methods to determine parameters for our continuous signal reconstruction algorithm, and Section 4.3.1.1 discusses an automated method for determining a temporal decay factor.

2.8 Training Framework

To create extensive detection models that represent a variety of circumstances and environmental conditions, data is required from different vehicles experiencing events in diverse scenarios, in addition to ground-truth data. Such data can be obtained from either real-world driving or from simulations.

Obtaining extensive data through real-world driving is prohibitively labor-intensive and expensive. With simulations, although the experiments can be extended beyond the available equipment and distances can be simulated that would be too costly to drive manually, the environment is often restricted to the variables of interest and the conditions

may not be perfectly representative of real-world environments.

To address this problem we use a simulator to synthesize training data for our classifiers, and we then evaluate our systems using both simulated and a more limited, but feasibly obtainable real-world data set. This process also demonstrates how models derived from simulated data can be applied to real-world scenarios.

We use the CarSim® [24] program to simulate vehicles in various driving conditions. CarSim® is a highly customizable vehicle simulation kit, which accurately simulates vehicle component (e.g., tires, suspension, steering, etc.) responses to given input environments and stimuli. It is the default tool for kinematic and controls simulation testing in the vehicle community [25]. With CarSim®, we simulated vehicles driving distances that vastly exceed what could be collected manually. Using simulations also provided us the knowledge of ground-truth information, since it was a simulation design parameter. The exact details of the CarSim® simulations are discussed in Chapter 5 as to how the simulations are constructed for testing the different types of road events.

2.9 Chapter Summary

In this chapter, we outlined the general vehicle-Cloud detection framework to extract environmental information from vehicle sensor data. The interactive cycle of initially acquiring the sensor data in individual vehicles, pre-processing the data, transmitting it to the Cloud for aggregation, running detection algorithms on the aggregated data, and transmitting the extracted information back to the vehicle for use in other products is described, along with appropriate modifications and adaptations that may be required to cope with varying types of environments, types of information, and algorithmic details.

The following chapters provide details for these high-level concepts, describe how the detection processes fit into the presented general vehicle-Cloud detection framework, and demonstrate examples of detecting specific types of environmental information.

Chapter 3

Noisy Multi-source Variable-rate Sampling

3.1 Introduction

Due to controller area network (CAN) bus and cellular bandwidth constraints, many vehicle embedded sensors operate at low sampling frequencies (e.g., 1 Hz). The low operating frequencies undersample road information signals in the vehicle sensor data, thus making aggregating sensor data a critical requirement to increase the effective sampling rate to a sufficient level so that environmental road information can be adequately detected. However, aggregating data from multiple vehicles or mobile sensors results in unique sampling conditions for the aggregated data set. The following paragraphs describe those sampling conditions, with examples of how they manifest themselves on an example signal shown in Figure 3.1.

Embedded sensors sample data as functions in the temporal domain, e.g., acquiring samples periodically at 1 Hz. However road information is generally indexed by spatial location instead of time. Speed and acceleration values are therefore required to map the sensor sampling rate from the temporal domain to the spatial domain. Since vehicles have

varying speeds and accelerations, even if the sensors were sampling perfectly periodically in the time domain, the sampling rates in the spatial domain would still vary. These varying sampling rates therefore produce nonuniform spacings between the sample locations. To further complicate the sampling environment, different sensors usually operate asynchronously and the shared buffers can produce output labeled sampling times that do not necessarily correspond to the true sampling times.

Similar to how individual vehicles produce nonuniformly spaced samples, different vehicles also operate asynchronously due to the lack of coordinated control. Since different vehicles have different acceleration patterns, the aggregated data from all the vehicles is not only nonuniformly spaced, but there may also be a different number of aggregated samples from each source over any given domain. The asynchronous sampling nature also implies that the aggregated samples may be undesirably bunched and the sampling density could vary significantly over the domain.

Aggregating data from multiple vehicles is also complicated by vehicle and sensor heterogeneity. Sensor responses, particularly from sensors measuring vehicle kinematics, are affected by the physical features of the vehicle as well as its driving behavior. For example, when measuring the accelerometer response from a vehicle driving over a pothole, factors such as the size, weight, and length of the vehicle, quality of the suspension system, and speed and turning behavior would all affect the vehicle's response to the pothole and therefore the resulting acceleration measurements. Therefore, when aggregating data from multiple vehicles it is important to consider that the data differs from vehicle to vehicle not just due to noise, but also because the vehicles' responses to certain events vary due to differences in the vehicles' physical characteristics.

These sampling conditions should be taken into account when designing methods for extracting information from networked vehicle systems. We refer to these conditions as noisy *multi-source, variable-rate* (MSVR) sampling conditions. The complexity of these

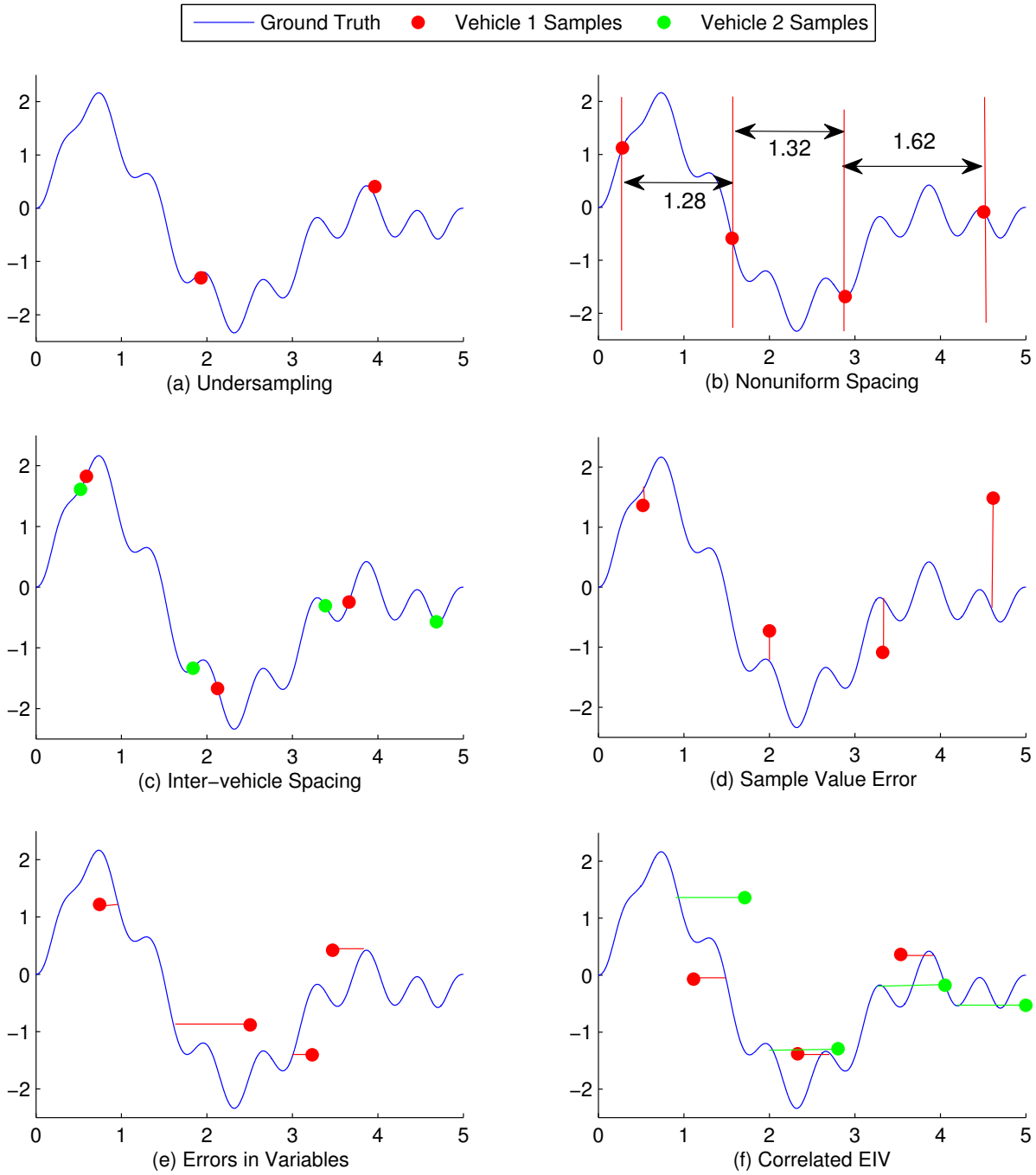


Figure 3.1: Vehicle MSVR sampled signal reconstruction difficulties, (a) Signal undersampling, (b) Nonuniform sample spacing, (c) Uncontrolled sample spacing between sensor sources, (d) Sample value error, (e) Errors in variables, (f) Correlated errors in variables

conditions is an important consideration for trying to accurately extract and localize information from undersampled and asynchronous vehicle sensor data.

There is an additional challenge that further distinguishes vehicle sampling conditions from traditional uniform sampling problems. Sensor measurements are indexed by the GPS locations at which they were measured. However GPS measurements have errors that are not independent and identically distributed (i.i.d.) for each sample, but instead may be spatially and temporally correlated. Consider as an example the GPS location measurements from 16 vehicle traces driving over a 300 m stretch of a two lane road (E 13 Mile Rd., Warren, MI, USA, between Mound Rd. and Ryan Rd.) in Figure 3.2. Note that the GPS locations from each trace do not randomly fluctuate around the lane centerlines from each lane. There are also more traces on the northern side of the road and some traces that would indicate a vehicle is being consistently driven in the middle of the road. The error, or location offsets from the lane centerlines, is strongly correlated for all the sampling locations from each trace. Also note that the spacing between samples in each trace and the total number of samples from each of the different traces varies in accordance with the MSVR sampling model.

A second example of correlated GPS error is shown in Figure 3.3. The arrows are from GPS measurements from 115 vehicle traces on Overlook Dr., Pittsburgh, Pennsylvania. All the vehicles purportedly follow the same route (the road hidden under the winding cluster of traces). However, note the purple arrows from a single vehicle trace that indicates measured locations that cut through the field rather than the road. This highlighted trace is in the same shape as the surrounding road, indicating a large and strongly correlated GPS error for that particular vehicle. Although this trace is an extreme example, other traces have similar GPS location biases that create difficulties when aggregating the erroneous sampling locations.

These location errors extend the MSVR sampling model to a correlated *errors in vari-*

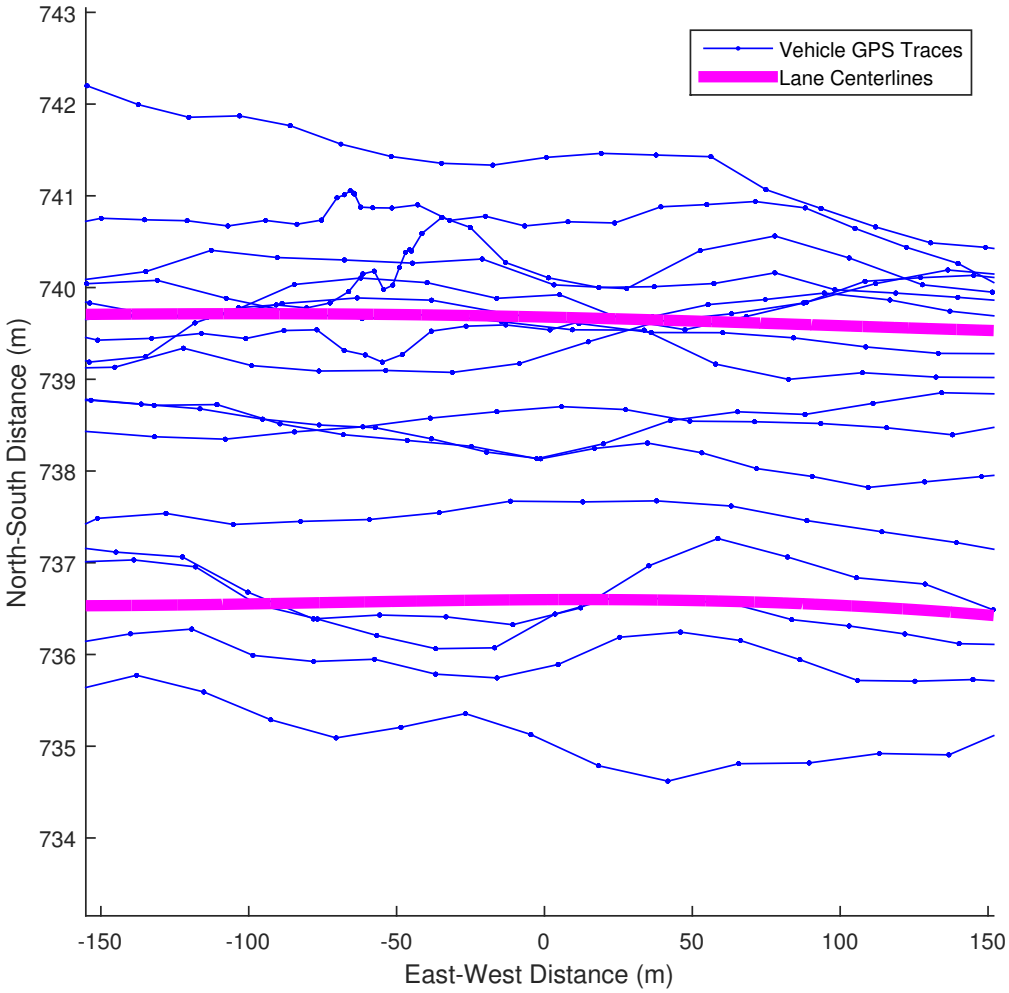


Figure 3.2: GPS traces from multiple vehicles on a two-lane road. The location error from the samples in individual traces is correlated.

Figure 3.3: Map of measured GPS locations on Overlook Dr., Pittsburgh, PA, from 115 vehicle traces

ables (EIV) sampling model where there is error in both the sampling value being measured (e.g., acceleration), and the sampling location. Although there has been ressearch investigating signal reconstructions with error in variables models, we extend the work to MSVR sampling conditions with particular consideration to correlations in the sampling location errors.

In this chapter, Section 3.2 provides background on how the GPS errors are correlated. Section 3.3 shows how we translate these error sources into mathematical models for use in the information extraction algorithms. Section 3.4 summarizes the chapter.

3.2 Global Positioning System (GPS) Background

The Global Positioning System (GPS) is a satellite-based navigation system that provides location estimates for, in this thesis work, vehicle-embedded GPS receivers. The GPS system was originally launched in 1995 with 24 satellites, and currently functions with 31 operational satellites. A receiver needs to have line-of-sight interactions with a minimum of 4 satellites to be able to determine its position.

Each satellite continually broadcasts its time of transmission, ephemeris data (satellite position), and a pseudorandom code signal. A GPS receiver generates a replica code signal for the respective monitored satellite, and by correlating the receiver-generated signal with the one received from the satellite, a receiver is able to estimate the signal propagation time [26]. This propagation time is converted into a distance measurement from the satellite, known as the pseudorange. The term "pseudorange" is used rather than "range" due to errors in the system contaminating the distance calculation. The pseudorange for each satellite corresponds to a sphere of positions where the receiver could be located. The spheres from a minimum of 4 satellites are used, often in a least-squares problem [27], to estimate the true location of the GPS receiver. As an oversimplification, this can be envisioned as the intersection of the spheres, however there are weightings by

Table 3.1: GPS pseudorange error standard deviations [26]

Error Source	Bias (m)	Random (m)	Total (m)
Ephemeris Data	2.1	0.0	2.1
Satellite Clock	2.0	0.7	2.1
Ionosphere (Single Freq.)	4.0	0.5	4.0
Ionosphere (Dual Freq.)	1.0	0.7	1.2
Troposphere	0.5	0.5	0.7
Multipath	1.0	1.0	1.4
Receiver	0.5	0.2	0.5
Total (Single Freq.)	5.1	1.4	5.3
Total (Dual Freq.)	3.3	1.5	3.6

probability distributions, adjustments for overdetermined systems, and nonlinearities in error calculations that shift the final estimated location.

The process of estimating the location of a GPS receiver contains a number of error sources that result in both random, but also spatially and temporally correlated location measurement errors. Although not intended as a thorough investigation of GPS error, the following sections highlight the most significant GPS error sources and how they manifest themselves as both persistently biased and independent random error values. The errors are divided into four categories: satellite errors (Section 3.2.1), atmospheric errors (Section 3.2.2), earth-level errors (Section 3.2.3), and satellite geometry errors (Section 3.2.4).

Each error section highlights how the errors may be spatially or temporally correlated. Table 3.1 summarizes the results of these sections, replicated from [26]. The errors are presented as a constant bias (an error which persists for at least minutes within the range of at least a few kilometers) in addition to a random independent error component, which can be interpreted as i.i.d. for each measurement. Each error component is assumed to be uncorrelated to other error components. These error values are similar to those obtained from other sources [28].

3.2.1 Satellite Errors

3.2.1.1 Ephemeris Error

The satellite ephemeris is its position in space. This position is required as a reference point when determining the pseudorange from a GPS receiver to a satellite. Satellites predict their trajectory based on previous observations and periodically uploaded (often daily) predictions made from ground stations [29]. A satellite's position error therefore undergoes diurnal cycles between uploaded corrections [30]. It was shown in [31] that the error autocorrelation function decreased approximately linearly over 30 minute time intervals, reaching an autocorrelation value of 0.5 at about 20 minutes. Any GPS receiver using a satellite to determine its location will continually receive the same correlated erroneous ephemeris data. For receivers relatively close to each other, the error is therefore strongly correlated as the pseudorange between devices is similarly affected in localized regions.

3.2.1.2 Satellite Clock Error

The satellite clock value is used to estimate the signal propagation time from the satellite to the receiver, and therefore to estimate the pseudorange. Atomic clocks can drift approximately 10^{-8} s per day, equivalent to about 3.5 m in pseudorange. This error has been shown to grow quadratically with time between updates [32, 33]. This clock error would be equal for any receiver (in any location) using the satellite.

3.2.2 Atmospheric Errors

3.2.2.1 Ionosphere Error

The wave propagation speed of the GPS signal is reduced from the speed of light in the ionosphere (50 km - 1000 km above the Earth's surface) due to the free electron concentration [34]. The delay is proportional to the squared inverse of the carrier frequency

of the signal. There are diurnal models for the electron concentrations which are used as a correction aid, however there are still errors when the true concentration does not match the model. Dual frequency signals or differential GPS receivers can be used to significantly reduce the error since the delay difference between the two signals can be better used to estimate the true delays [34]. These errors are correlated in localized regions but are also dependent on latitude, with the error increasing towards the equator and poles [35].

3.2.2.2 Troposphere Error

The troposphere is the atmospheric level closest to earth (0 km - 50 km above the Earth's surface), which is associated with most day-to-day weather variations. Since the signal propagation speed is affected by changes in temperature, pressure, and humidity, localized weather conditions can significantly delay the propagation time of the signal. These effects vary depending on satellite elevation since there is a longer path through the troposphere for lower elevation signals. However, given current conditions, the tropospheric errors are fairly predictable and can therefore be compensated for when estimating the pseudorange [36]. These errors are also heavily correlated in time and location, however can vary for signals from different satellites for the same receiver as their path lengths through the troposphere are dependent on their geometry relative to the receiver.

3.2.3 Earth-level Errors

3.2.3.1 Multipath Errors

Multipath errors are caused by the satellite signal reflecting off nearby objects to the receiver (e.g., large buildings, trees) creating multiple paths to the receiver and therefore multiple (and delayed) estimates for the signal propagation time. These effects are the most difficult to estimate or compensate for as they are highly localized and highly dependent on which satellites a receiver is using for the pseudorange calculations. Receivers in the

same location may experience similar multi-path error from a satellite if the same objects are interfering with the signal.

3.2.3.2 Receiver Errors

Any coding, clock, or algorithm errors within a device will cause an error in the location estimate. A coding error for example could be a mismatch between the satellite pseudo-random code and the receiver-generated one meant to correlate the signals. Depending on the exact nature of these errors, they may be temporally correlated, however they are unlikely to be correlated between different GPS receivers in close proximity.

3.2.4 Geometric Dilution of Precision

The values expressed in Table 3.1 only express the errors in the pseudorange calculation. It is important to also determine how those errors are reflected in localization accuracy on an earth-level plane. This projection is done using the Geometric Dilution of Precision (GDOP) which uses the satellite geometry to determine how the range errors are expressed as localization errors. The GDOP is the ratio of the size of the location errors to the range errors [37]. The GDOP is derived from the covariance matrix, $\mathbf{Q}$, of the east (E), north (N), elevation (Z), and time (t) errors,

$$\mathbf{Q} = \begin{bmatrix} \sigma_E^2 & \sigma_{EN}^2 & \sigma_{EZ}^2 & \sigma_{Et}^2 \\ \sigma_{EN}^2 & \sigma_N^2 & \sigma_{NZ}^2 & \sigma_{Nt}^2 \\ \sigma_{EZ}^2 & \sigma_{NZ}^2 & \sigma_Z^2 & \sigma_{Zt}^2 \\ \sigma_{Et}^2 & \sigma_{Nt}^2 & \sigma_{Zt}^2 & \sigma_t^2 \end{bmatrix}. \tag{3.1}$$

The matrix $\mathbf{Q}$ is determined by

$$\mathbf{Q} = (\mathbf{G}^T\mathbf{G})^{-1}, \tag{3.2}$$

where $\mathbf{G}$ is derived from the satellite geometry for the S satellites used in determining the location from their azimuth, α, and elevation, β, angles, as illustrated in Figure 3.4, as the

transformation between spherical and Cartesian coordinate systems such that,

$$\mathbf{G} = \begin{bmatrix} \cos\beta_1 \sin\alpha_1 & \cos\beta_1 \cos\alpha_1 & \sin\beta_1 & 1 \\ \cos\beta_2 \sin\alpha_2 & \cos\beta_2 \cos\alpha_2 & \sin\beta_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \cos\beta_S \sin\alpha_S & \cos\beta_S \cos\alpha_S & \sin\beta_S & 1 \end{bmatrix}. \tag{3.3}$$

Particular interest is paid to the horizontal dilution of precision (HDOP), where

$$\text{HDOP} = \sqrt{\sigma_E^2 + \sigma_N^2}, \tag{3.4}$$

as that expresses the location error multiplicative factor in the latitudinal and longitudinal plane. The HDOP is traditionally used to indicate the change in error on the ground plane, however particular attention should be paid to the off-diagonal terms of $\mathbf{Q}$ as those indicate how errors are correlated between the latitudinal and longitudinal directions. Typical HDOP values are between 1.0-2.0 [37]. The elevation errors are typically larger than the horizontal earth-plane errors since all the satellite measurements are biased to one side along the vertical axis from the receiver (all satellites are above the receiver) while the lateral and longitudinal coordinates are typically determined from satellites surrounding the receiver [37]. This disparity increases as the latitudes approach the poles since the satellite elevations are more limited.

To illustrate the difference in satellite geometries, consider a situation with four satellites spread evenly around the sky with elevation and azimuth angles

$$\begin{bmatrix} \beta_1 & \alpha_1 \\ \beta_2 & \alpha_2 \\ \beta_3 & \alpha_3 \\ \beta_4 & \alpha_4 \end{bmatrix} = \begin{bmatrix} 5^\circ & 0^\circ \\ 5^\circ & 120^\circ \\ 5^\circ & 240^\circ \\ 0^\circ & 90^\circ \end{bmatrix}. \tag{3.5}$$

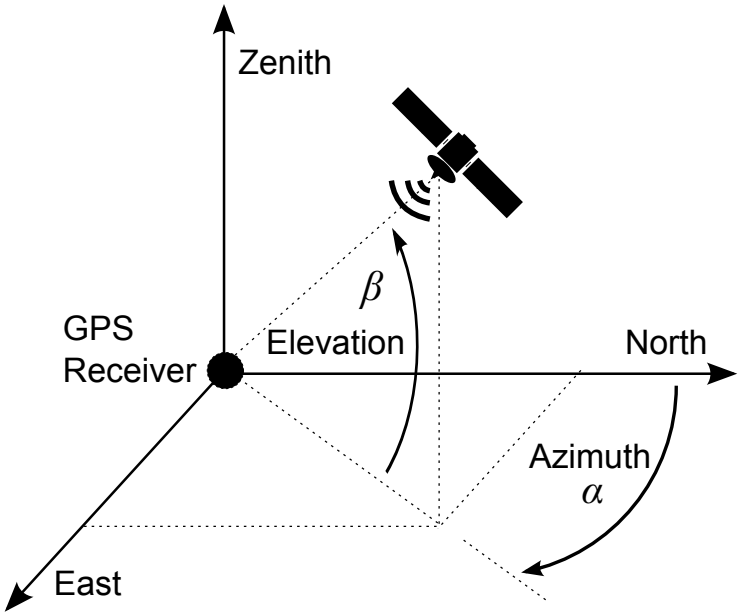


Figure 3.4: Illustration of the satellite azimuth (α) and elevation (β) angles as related to a GPS receiver

The resulting GDOP matrix is

$$\mathbf{Q} = \begin{bmatrix} 0.67 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.67 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.60 & -0.50 \\ 0.00 & 0.00 & -0.50 & 0.41 \end{bmatrix}. \tag{3.6}$$

The resulting HDOP value is only 1.16 and the errors in the latitude and longitude are uncorrelated. Compare this with a second situation where the satellites are not as evenly distributed around the sky, with elevation and azimuth angles,

$$\begin{bmatrix} \beta_1 & \alpha_1 \\ \beta_2 & \alpha_2 \\ \beta_3 & \alpha_3 \\ \beta_4 & \alpha_4 \end{bmatrix} = \begin{bmatrix} 15^\circ & 0^\circ \\ 20^\circ & 90^\circ \\ 15^\circ & 240^\circ \\ 60^\circ & 120^\circ \end{bmatrix}. \tag{3.7}$$

Table 3.2: MSVR sampling notation

N	Total number of aggregated samples
V	Total number of sampling sources
n_v	Total number of samples from source v
$\mathbf{x}_j^v$	Measured location of data sample j from source v
$\mathbf{q}_j^v$	True location of data sample j from source v
$\mathbf{\Delta}_j^v$	Location error of data sample j from source v
$\boldsymbol{y}_j^v$	Measured sensor value for measured location $\boldsymbol{x}_j^v$
t_j^v	Measurement time of data sample j from source v
σ_x^2	Variance for sampling location error for each data sample
σ_v^2	Variance for sampling location error for each source
σ_a^2	Variance for sampling location error for autocorrelated error model
σ_y^2	Variance for sample value error
T	Exponential autocorrelation time constant
ρ_{jk}^v	Correlation factor for sampling location error for samples j and k from source v
$\mathbf{C}_{xx}$	Covariance matrix for sample location errors from aggregated data
$\mathbf{C}_{xx}^v$	Sample location error covariance matrix for source v
$\mathbf{C}_{vv}$	Condensed sample location error covariance matrix for all sources

The resulting GDOP matrix is

$$\mathbf{Q} = \begin{bmatrix} 0.79 & -0.33 & -1.01 & 0.35 \\ -0.33 & 1.07 & 1.12 & -0.50 \\ -1.01 & 1.12 & 5.72 & -2.40 \\ 0.35 & -0.50 & 2.40 & 1.27 \end{bmatrix}. \tag{3.8}$$

In this satellite geometry, the HDOP value is 1.36, and the variances for the latitude and longitude are now different, with their components correlated.

3.3 Sample Location Error Models

Refer to Table 3.2 for a notation reference for this section. As outlined in the previous section, GPS measurements have associated errors inherent in the measurements that are

both spatially and temporally correlated. As related to the Intelligent Transportation System (ITS), consider a system of V vehicles collecting data samples on a road or road network. The j^{th} sample location from the v^{th} vehicle, where $v \in \{1, \ldots, V\}$ and $j \in \{1, \ldots, n_v\}$, is measured at the true (but initially unknown) location $\mathbf{q}_j^v$, however reported to be at $\mathbf{x}_j^v$ due to GPS positioning error. The measured location is offset from the true location by the sampling location error, $\boldsymbol{\Delta}_j^v$, such that

$$\mathbf{x}_j^v = \mathbf{q}_j^v + \boldsymbol{\Delta}_j^v. \tag{3.9}$$

The measured and true location variables could be vectors in in $\mathbb{R}^3$ if considering a (latitude, longitude, elevation) coordinate system, in $\mathbb{R}^2$ if one is only considering the (latitude, longitude) system, or in $\mathbb{R}^1$ if one is only considering a total distance coordinate system, for example, the location along a single road. This is an *errors in variables* (EIV) [38] sampling model due to the error $\boldsymbol{\Delta}_j^v$.

There is also a signal of interest generally being measured with reference to the GPS locations, for example, accelerometer or gyroscope readings. Let the ground-truth signal be represented by the function $f(\mathbf{q})$. Therefore the measurement output, $\mathbf{y}_j^v$ corresponding to the measured location $\mathbf{x}_j^v$ is given by

$$\mathbf{y}_j^v = f(\mathbf{q}_j^v) + \boldsymbol{\nu}_j^v, \tag{3.10}$$

where $\boldsymbol{\nu}_j^v$ is the sample value error from the measurement. Therefore the true coupled location and signal values, $(\mathbf{q}_j^v, f(\mathbf{q}_j^v))$, are output as the reported measured values $(\mathbf{x}_j^v, \mathbf{y}_j^v)$. Figure 3.5 presents an illustration of the relationship between these variables for an example one-dimensional coordinate system.

When attempting to extract environmental information from the sensed vehicle data, it is imperative that the error terms are properly taken into account to try and determine the true location of the sensed data. Accounting for such behavior requires an error model that well approximates the correlated location errors for all the GPS measurements in an aggregated data set that includes data from multiple vehicles. To accomplish this goal, we

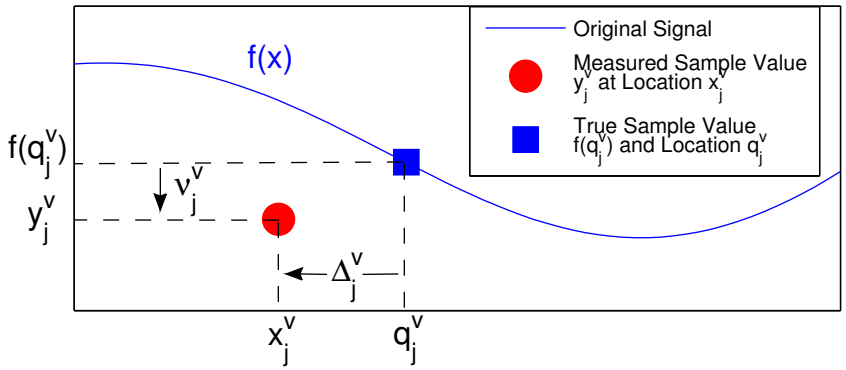


Figure 3.5: Relationship of measured sampled location, x_j^v, sampled value, y_j^v, true sampling location, q_j^v, and true sample value, $f(q_j^v)$. The difference between x_j^v and q_j^v is the sampling location error and the difference between y_j^v and $f(q_j^v)$ is the sampling value error.

require the covariance matrix $\mathbf{C}_{xx}$ that relates the variances of all the location error terms for the sampled locations of the aggregated data, where

$$\mathbf{C}_{xx} = E\left[\begin{bmatrix} \Delta_1^1 \\ \vdots \\ \Delta_{n_1}^1 \\ \vdots \\ \Delta_1^V \\ \vdots \\ \Delta_{n_V}^V \end{bmatrix} \begin{bmatrix} \Delta_1^1 & \dots & \Delta_{n_1}^1 & \dots & \Delta_1^V & \dots & \Delta_{n_V}^V \end{bmatrix}\right]. \tag{3.11}$$

Let us also assume that $\mathbf{x}, \mathbf{q} \in \mathbb{R}^1$.

Predetermining a covariance matrix that is generally applicable is difficult given all the variables that may affect the spatial and temporal correlations in error from data samples from multiple vehicles. We therefore make a few qualifying assumptions. The spatial and temporal correlations for a single receiver are quite evident, as described by the error sources in Section 3.2, however the correlation in location error between different GPS receivers is more difficult to quantify since they may be tracking different satellites. We therefore construct the covariance matrix such that the location error from samples from

different vehicles is assumed to be independent. Therefore $\mathbf{C}_{xx}$ is a block diagonal matrix, such that

$$\mathbf{C}_{xx} = \begin{bmatrix} \mathbf{C}_{xx}^1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{xx}^2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_{xx}^V \end{bmatrix}, \tag{3.12}$$

where $\mathbf{C}_{xx}^v$ is the location error covariance matrix for the data samples collected from vehicle v.

In the following sections, we present error models for $\mathbf{\Delta}_j^v$ inspired by the background information on GPS error.

3.3.1 Autocorrelated Location Error

Although the errors in Table 3.1 were presented as a persistent bias plus random error, it is useful to quantify time scales over which such assumptions are applicable. An exponential autocorrelation function has been found to well approximate the GPS error as it evolves in time [39, 40]. Given the j^{th} and k^{th} measurements from vehicle v at times t_j^v and t_j^v, respectively, the covariance between the location error of the two measurements is given by

$$\sigma^2_{\Delta_j^v, \Delta_k^v} = \sigma_a^2 \exp\left(-\frac{\left|t_j^v - t_k^v\right|}{T}\right), \tag{3.13}$$

where T is the time constant of the exponential (found to vary between 200-500 seconds for GPS applications [39]), and σ_a^2 is the variance of the sample location error. Denote the autocorrelation component of the variance as ρ_{jk}^v, where

$$\rho_{jk}^v = \exp\left(-\frac{\left|t_j^v - t_k^v\right|}{T}\right). \tag{3.14}$$

By expanding the autocorrelation model to all the samples from each individual vehicle, the covariance matrix $\mathbf{C}_{xx}^{v}$ is determined as

$$\mathbf{C}_{xx}^{v} = \sigma_a^2 \begin{bmatrix} 1 & \rho_{12}^{v} & \rho_{13}^{v} & \dots & \rho_{1n_v}^{v} \\ \rho_{12}^{v} & 1 & \rho_{23}^{v} & \dots & \rho_{2n_v}^{v} \\ \rho_{13}^{v} & \rho_{23}^{v} & 1 & \dots & \rho_{3n_v}^{v} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{1n_v}^{v} & \rho_{2n_v}^{v} & \rho_{3n_v}^{v} & \dots & 1 \end{bmatrix}. \tag{3.15}$$

If a periodic sampling model is assumed where

$$\tau = |t_{j+1}^{v} - t_{j}^{v}|, \quad \forall j, \tag{3.16}$$

then denote ρ_τ as

$$\rho_\tau = \exp\left(-\frac{\tau}{T}\right), \tag{3.17}$$

which simplifies the covariance matrix $\mathbf{C}_{xx}^{v}$ to

$$\mathbf{C}_{xx}^{v} = \sigma_a^2 \begin{bmatrix} 1 & \rho_\tau & \rho_\tau^2 & \dots & \rho_\tau^{n_v-1} \\ \rho_\tau & 1 & \rho_\tau & \dots & \rho_\tau^{n_v-2} \\ \rho_\tau^2 & \rho_\tau & 1 & \dots & \rho_\tau^{n_v-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_\tau^{n_v-1} & \rho_\tau^{n_v-2} & \rho_\tau^{n_v-3} & \dots & 1 \end{bmatrix}. \tag{3.18}$$

However due to missing data, and asynchronous and irregular sampling in embedded sensors, the periodic condition is unlikely to occur in practice. To demonstrate these properties, Table 3.3 shows speed measurements from a vehicle driving indexed by the epoch timestamp of the measurement. Note that the time difference between measurements (shown in the last column) is inconsistent.

Table 3.3: Example real-world vehicle speed measurements

Timestamp (ms) UNIX Epoch	Speed (km/h)	Change in Time from Previous Sample (ms)
1443637460637	79.39	
1443637460752	79.39	115
1443637460863	79.39	111
1443637460940	79.42	77
1443637461056	79.45	116
1443637461140	79.46	84
1443637461220	79.48	80
1443637461333	79.49	113
1443637461423	79.53	90
1443637461555	79.53	132
1443637461657	79.54	102
1443637461734	79.56	77
1443637461840	79.60	106

The inverse of $\mathbf{C}_{xx}$ will be required for optimization problems over the aggregated data set in Section 4.1. Due to its block diagonal structure,

$$(\mathbf{C}_{xx})^{-1} = \begin{bmatrix} (\mathbf{C}_{xx}^{1})^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & (\mathbf{C}_{xx}^{2})^{-1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & (\mathbf{C}_{xx}^{V})^{-1} \end{bmatrix}. \tag{3.19}$$

The matrix inverse for the vehicle sample error covariance matrix given by Equation 3.15 is,

$$(\mathbf{C}_{xx}^{v})^{-1} = \frac{1}{\sigma_a^2} \begin{bmatrix} \frac{1}{1-(\rho_{12}^{v})^2} & \frac{-\rho_{12}^{v}}{1-(\rho_{12}^{v})^2} & 0 & 0 & \dots & 0 \\ \frac{-\rho_{12}^{v}}{1-(\rho_{12}^{v})^2} & \frac{(\rho_{12}^{v})^2}{1-(\rho_{12}^{v})^2} + \frac{1}{1-(\rho_{23}^{v})^2} & \frac{-\rho_{23}^{v}}{1-(\rho_{23}^{v})^2} & 0 & \dots & 0 \\ 0 & \frac{-\rho_{23}^{v}}{1-(\rho_{23}^{v})^2} & \frac{\rho_{23}^{v\,2}}{1-(\rho_{23}^{v})^2} + \frac{1}{1-(\rho_{34}^{v})^2} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \frac{-\rho_{n_v-2,n_v-1}^{v}}{1-(\rho_{n_v-2,n_v-1}^{v})^2} & 0 \\ 0 & \dots & 0 & \frac{-\rho_{n_v-2,n_v-1}^{v}}{1-(\rho_{n_v-2,n_v-1}^{v})^2} & \frac{(\rho_{n_v-2,n_v-1}^{v})^2}{1-(\rho_{n_v-2,n_v-1}^{v})^2} + \frac{1}{1-(\rho_{n_v-1,n_v}^{v})^2} & \frac{-\rho_{n_v-1,n_v}^{v}}{1-(\rho_{n_v-1,n_v}^{v})^2} \\ 0 & 0 & \dots & 0 & \frac{-\rho_{n_v-1,n_v}^{v}}{1-(\rho_{n_v-1,n_v}^{v})^2} & \frac{1}{1-(\rho_{n_v-1,n_v}^{v})^2} \end{bmatrix}. \tag{3.20}$$

3.3.2 Correlated Source Error

For some time-scales or distances, it may be more convenient to approximate the autocorrelation error model as a constant error bias for each vehicle in addition to an independent component, indicated in the style of Table 3.1. We follow a similar construction to [41] and construct the vehicle location error covariance matrix as the sum of independent and correlated components,

$$\mathbf{C}_{xx}^{v} = (\mathbf{C}_{xx}^{v})^{\text{ind}} + (\mathbf{C}_{xx}^{v})^{\text{corr}}. \tag{3.21}$$

Following this structure, we assume that the independent location error component is distributed as $\mathcal{N}(0, \sigma_x^2)$ and i.i.d. for all samples. The correlated error is a fixed bias term for each vehicle, with each vehicle's bias term distributed as $\mathcal{N}(0, \sigma_v^2)$. Following this model,

$$\mathbf{C}_{xx}^{v} = \sigma_x^2 \mathbb{I}_{n_v \times n_v} + \sigma_v^2 \mathbf{1}_{n_v \times n_v}, \tag{3.22}$$

$$= \begin{bmatrix} \sigma_v^2 + \sigma_x^2 & \sigma_v^2 & \sigma_v^2 & \cdots \\ \sigma_v^2 & \sigma_v^2 + \sigma_x^2 & \sigma_v^2 & \cdots \\ \sigma_v^2 & \sigma_v^2 & \sigma_v^2 + \sigma_x^2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{3.23}$$

where $\mathbb{I}_{n \times n}$ is the identity matrix of size $n \times n$, and $\mathbf{1}_{n \times m}$ is a $n \times m$ matrix of ones. This covariance matrix is similar to that of Equation 3.15 in the Autocorrelated error model. However, while this model maintains a constant correlation for any two different sample location errors from the same vehicle, the autocorrelation model reduces the correlation as the measurements are further separated in time.

As in the previous section, the inverse of $\mathbf{C}_{xx}$, and therefore $\mathbf{C}^v_{xx}$, is required for optimization problems. Due to the Toeplitz structure of $\mathbf{C}^v_{xx}$, its inverse is,

$$(\mathbf{C}^v_{xx})^{-1} = \begin{bmatrix} A_v & B_v & B_v & \dots \\ B_v & A_v & B_v & \cdots \\ B_v & B_v & A_v & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{3.24}$$

where

$$A_v = \frac{(n_v - 1)\sigma_v^2 + \sigma_x^2}{n_v\sigma_v^2\sigma_x^2 + (\sigma_x^2)^2}, \tag{3.25}$$

$$B_v = \frac{-\sigma_v^2}{n_v\sigma_v^2\sigma_x^2 + (\sigma_x^2)^2}. \tag{3.26}$$

This considers the covariance between all of the sample location errors. However as shown in Table 3.1, the source bias error (σ_v^2) is often much larger than the independent sample location error (σ_x^2).

When constructing or comparing the correlated and autocorrelated noise models, the covariance matrices of the two algorithms can be designed to approximate each other if necessary. This can be accomplished by minimizing the Frobenius norm between the two covariance matrices for a given vehicle, $\mathbf{C}^v_{xx}$. For a given location variance σ_a^2 and time constant T for the autocorrelated noise model, the required location variance terms for the correlated source error model, σ_v^2 and σ_x^2, are therefore solved to be,

$$\sigma_v^2 = \sigma_a^2 \left(\frac{\sum_{i=1}^{n_v} \sum_{j=i+1}^{n_v} \rho_{ij}^v}{n_v^2 - n_v} \right), \tag{3.27}$$

$$\sigma_x^2 = \sigma_a^2 \left(\frac{n_v^2 - \sum_{i=1}^{n_v} \sum_{j=1}^{n_v} \rho_{ij}^v}{n_v^2 - n_v} \right). \tag{3.28}$$

3.3.3 Fixed Source Error

For a sufficiently small distance and time, if the correlated location error component dominates the independent noise component, then a simplifying assumption could be made to neglect the independent noise component. The covariance matrix for the sample location error from each vehicle could therefore be approximated as

$$\mathbf{C}_{xx}^{v} = \sigma_v^2 \mathbf{1}_{n_v \times n_v}. \tag{3.29}$$

This effectively means that there is a fixed (but random and independent for each vehicle) offset for all sample locations from individual vehicles. This formulation reduces the number of error model parameters from the two in the previous two models (σ_a^2 and T for the autocorrelation model, and σ_v^2 and σ_x^2 for the correlated error model) to just one (σ_v^2) for this fixed source error model.

The covariance matrix $\mathbf{C}_{xx}^{v}$ has all elements identical, and is therefore singular and cannot be inverted. However since the location errors from all samples from each individual vehicle are equal, the dimension of the covariance matrix $\mathbf{C}_{xx}$ can be reduced so that instead of expressing the covariance between all sample location error terms as in Equation 3.11, it expresses only the error for the fixed vehicle offset error, Δ^v, where

$$\Delta^v = \Delta_1^v = \Delta_2^v = \cdots = \Delta_{n_v}^v. \tag{3.30}$$

We therefore replace the sample location covariance matrix $\mathbf{C}_{xx}$ with a vehicle location error covariance matrix, $\mathbf{C}_{vv}$, where

$$\mathbf{C}_{vv} = E\left[\begin{bmatrix}\Delta^1\\ \vdots \\ \Delta^V\end{bmatrix}\begin{bmatrix}\Delta^1 & \dots & \Delta^V\end{bmatrix}\right]. \tag{3.31}$$

This formulation simplifies to

$$\mathbf{C}_{vv} = \sigma_v^2 \mathbb{I}_{V \times V}. \tag{3.32}$$

Therefore the matrix inverse is clearly given by

$$\mathbf{C}_{vv}^{-1} = \frac{1}{\sigma_v^2}\mathbb{I}_{V \times V}. \tag{3.33}$$

3.4 Chapter Summary

In this chapter we introduced the noisy *multi-source variable-rate* (MSVR) sampling model that results from vehicular sensing conditions in the Intelligent Transportation System. We demonstrated how the MSVR sampling model was an errors in variables model, specifically with correlated errors due to the nature of GPS location estimates. Three different error models were introduced for the correlated error model, due to varying assumptions on the sampling conditions. In the following chapter we demonstrate how these noisy MSVR sampling conditions are incorporated into machine learning and signal processing algorithms for extracting information from vehicle sensor data.

Chapter 4

Aggregated Data Detection Algorithms

There are many factors that influence how a detection algorithm needs to be constructed to properly extract the desired environmental information from undersampled and asynchronous vehicle sensor data. These factors were outlined in Section 2.2 and include the event type, and spatial and temporal information, among others. The information extraction algorithms presented in this section are designed to handle the noisy multi-source variable-rate (MSVR) sampling conditions described in Chapter 3 within the overall information extraction framework.

In this chapter, we describe specific algorithms for event detection with regards to the overall vehicle-Cloud detection framework presented in Chapter 2. Specifically, the framework for reconstructing continuous event data is presented in Section 4.1 and binary event data is presented in Section 4.2. Section 4.3 describes adaptations to the framework to handle dynamic (i.e., temporally changing) information. These algorithms are analyzed with respect to general signals and data in this chapter and their application to specific types of environmental information will be presented in Chapter 5.

4.1 Continuous Signal Reconstruction

Many types of road environmental information are best represented by continuous signals as functions of location. Example continuous information includes static road topography (elevation, road inclination, bank angle, curvature), dynamic real-time conditions (traffic density, road friction, ambient temperature), or others shown in Table 2.1. This information is essential for route planning, predictive steering, and speed control, among other functions. These algorithms are beneficial for adaptive driver assistance, automated processes, city information databases, and eventually autonomous vehicles.

There are, however, a number of significant challenges associated with reconstructing continuous signals from vehicle sensor data. One of the more significant challenges is the noisy MSVR sampling conditions detailed in Chapter 3. These MSVR sampling conditions prove to be difficult for signal reconstruction due to the following reasons:

- Due to the functional constraints of the vehicle sensor technology and limited network bandwidth, the signals of interest are mostly *undersampled* in each vehicle. At low sampling frequencies, crowdsourcing data is critical to obtain sufficient samples to robustly reconstruct the desired signals.
- The various embedded vehicle sensors operate asynchronously and often at different sampling rates. Different vehicles are also asynchronous as there is no coordinated control between the sampling instances on different vehicles. Furthermore, most of the signals of interest are functions in the spatial domain. Since the sensors sample as a function of time, different speeds and accelerations, both within a single vehicle and when compared vehicle to vehicle, cause the spacing of sample locations in the aggregated data sets to be *nonuniformly spaced*. Additionally, the number of samples collected from each source over a given domain will vary. These conditions must be considered in the signal reconstruction algorithms.
- Sensor output varies from vehicle to vehicle. Factors such as *noise*, vehicle *hetero-*

geneity (i.e., varying vehicle size, weight, sensor placement), sensor heterogeneity, speed, and other driver-specific behavior affect the measurements different vehicles produce.

- Data samples collected from each individual vehicle have *correlated* location errors as described in Section 3.2. These location errors create an *errors in variables* model, that unlike standard regression, creates an ill-posed problem for signal reconstruction.

The following sections describe how to reconstruct continuous signals given these sampling conditions. Section 4.1.1 describes related work that reconstructs signals while accounting for some subset of the sampling constraints. Section 4.1.2 details our algorithm for signal reconstruction under the noisy and correlated MSVR vehicle sampling conditions. Section 4.1.3 extends these algorithms to joint signal reconstructions. Section 4.1.4 provides details on how to determine or optimize the algorithmic parameters.

4.1.1 Related Signal Reconstruction Work

It should be noted that the language and notation in this chapter refers to reconstructing a signal, $f(x)$, as a function of spatial coordinate x. However, the methods are equally applicable to temporal signals or any situation with noisy MSVR sampling conditions. Other applications with MSVR sampling conditions include:

- Several low-frequency analog-to-digital converters (ADCs) interleaved in time to achieve periodic sampling at higher frequencies [42].
- Astronomy applications, for example measuring star luminosity, where samples are obtained by geo-distributed devices in an irregular manner due to weather conditions, diurnal cycles, or equipment malfunctions [43]. The biases from individual equipment units, the difficulties in syncing globally or orbitally distributed devices, and the varied conditions at different sites produce correlated errors in the time and distance measurements from individual sensors.

Other works have attempted to reconstruct signals in the presence of a subset of the noisy MSVR sampling conditions outlined in Chapter 3, however there are limitations to these algorithms that prevent them from adequately reconstructing signals under the full noisy and correlated MSVR conditions.

4.1.1.1 Paley-Wiener-Levinson Theorem

There are a number of signal reconstruction algorithms that are specifically designed for nonuniform sample spacing [44]. These are more traditional sampling algorithms that generally follow the Paley-Wiener-Levinson (PWL) theorem [45] for signal reconstruction. Under PWL, the function $f(x)$ is interpolated as

$$\hat{f}(x) = \sum_{j=-\infty}^{\infty} y_j \frac{G(x)}{\frac{\partial G(x_j)}{\partial x}(x - x_j)}, \tag{4.1}$$

where $G(x)$ is an interpolated Lagrange polynomial [46],

$$G(x) = \prod_{j=-\infty}^{\infty} \left(1 - \frac{x}{x_j}\right). \tag{4.2}$$

This interpolation matches all the input samples ($\hat{f}(x_j) = y_j, \forall j$) in the reconstruction.

The interpolation in Equation 4.1 is a generalized model for nonuniform sampling. If x were uniformly sampled such that $x_j = \chi + jW$, then $G(x) = \sin\left(\pi\left(\frac{x-\chi}{W}\right)\right)$ and the PWL interpolation in Equation 4.1 reduces to the traditional Whittaker-Shannon-Kotelnikov (WSK) theorem [47, 48, 49] for uniform sampling, where

$$\hat{f}(x) = \sum_{j=-\infty}^{\infty} y_j \cdot \mathrm{sinc}\left(\frac{x - \chi - jW}{W}\right), \tag{4.3}$$

and $\mathrm{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$. The PWL theorem can be adapted to a class of reconstructions for multiple sampling sources, each sampling periodically. In this formulation, the different sampling sources have specified offsets such that

$$x_j^v = jW_v + \chi_v, \quad v \in \{1, \ldots, V\}, \tag{4.4}$$

where χ_v and W_v are the phase offset and the sampling period, respectively, for sampling source v [46]. This results in the following reconstruction function,

$$\hat{f}(x) = \sum_{j=-\infty}^{\infty} \sum_{k=1}^{V} \frac{\prod_{v=1}^{V} \sin\left(\frac{\pi}{W_v}(x-\chi_v)\right)}{(-1)^k \frac{\pi}{W_v} \left[\prod_{v=1, v\neq k}^{V} \sin\left(\frac{\pi}{W_v}(x_j^k - W_v)\right)\right](x - x_j^k)}. \tag{4.5}$$

There are a number of limitations associated with these reconstructions. The Lagrange polynomial is constructed based on the unrealistic assumption of infinite number of samples. Truncation error is introduced when the number of samples is finite, particularly when the sources sample below the Nyquist rate. There are functions which can be substituted to increase the rate of truncation error decay in the reconstruction [50], however these are also less effective below the Nyquist rate as aliasing still occurs.

These approaches are based on the requirement that the reconstructed signal matches all the samples perfectly (i.e., $\hat{f}(x_j^v) = y_j^v$, $\forall v, j$). This requirement is evident from the Lagrange interpolation formula that ensures that $f(x_j^v) = y_j^v$ and therefore does not account for the unique types of noise in MSVR samples. Matching samples on the estimated function exactly is not a necessary or good requirement in the presence of noise, and in fact often has negative consequences on the reconstruction due to overfitting.

As an example of this overfitting, consider the simple function $y = \sin(\pi x)$. Thirty samples were drawn from a uniform distribution over the domain $[-4, 4]$. Reconstructions using the PWL theorem were attempted under two sampling conditions. In the first sampling condition, there was no additional noise. In the second sampling condition, zero-mean white noise with standard deviation 0.001 was added to each of the sampling values. The reconstructions for each of these sampling conditions are shown in Figure 4.1. Note that even though the added sampling noise was too small to even be visible in the figure, it necessitated complicated dynamics and higher frequency signal components to reconstruct a signal that exactly matches with each sample. This poor reconstruction is an example of how the PWL reconstructions fail in the presence of noise. Our signal reconstruction

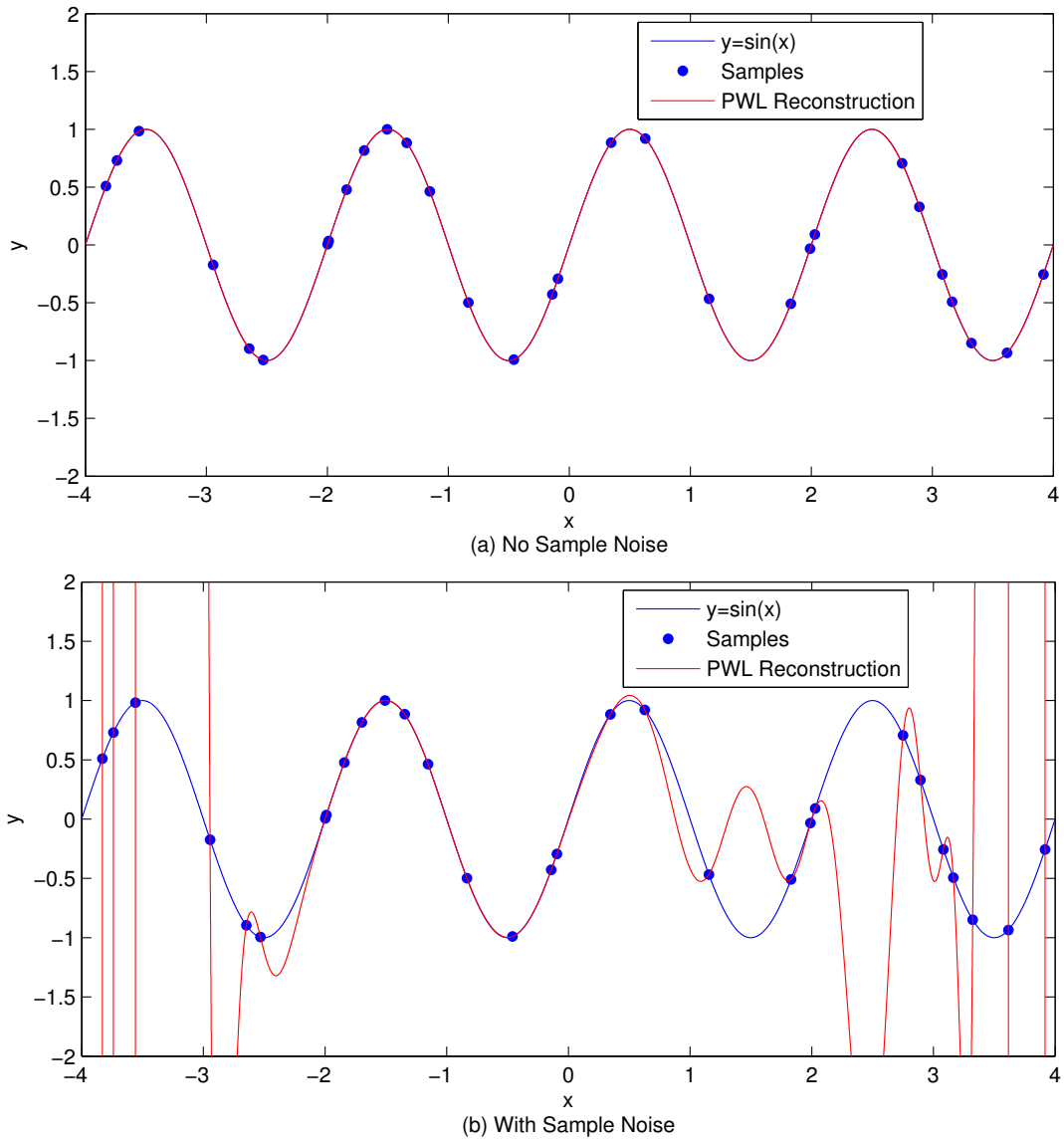


Figure 4.1: Signal reconstruction from nonuniformly spaced samples using the PWL theorem for both (a) non-noisy, and (b) noisy samples. The degree of the polynomial $G(x)$ is 30 due to 30 samples being used.

solution does not have the requirement that the reconstruction fits the samples exactly.

4.1.1.2 Trigonometric Polynomials

Sample value noise can be tolerated in the trigonometric polynomial interpolation approaches of [51, 52, 53]. However, these works are only concerned with the nonuniform

spacing between the samples. We extend the trigonometric reconstructions in Section 4.1.2 to incorporate MSVR sampling conditions and errors in variables conditions. There is particular focus given to how source information can be used to reconstruct the signal in the presence of correlated location error biases from the samples from each sensor agent.

4.1.1.3 Nonparametric Regression Algorithms

There is a class of nonparametric regression algorithms that attempt to estimate $f(x)$ by fitting kernel density functions to the sample probability density functions. The work in [54], for example, estimates $f(x)$ as

$$\hat{f}_n(x) = \frac{\sum_{i=1}^{n} K_n(\frac{x-x_i}{h_n}) y_i}{\sum_{i=1}^{n} K_n(\frac{x-x_i}{h_n})}, \tag{4.6}$$

where for a Gaussian kernel function,

$$K_n(x) = \frac{h_n}{\sqrt{2\pi(h_n^2 - \sigma_x^2)}} \exp\left(-\frac{h_n^2 x^2}{2(h_n^2 - \sigma_x^2)}\right), \;\; h_n > \sigma_x, \tag{4.7}$$

where h_n is a preselected bandwidth parameter, and σ_x^2 is the variance on the sampling location error of x_j. The regression is extremely sensitive to this parameter and the algorithm does not account for any correlation in the sample location error.

Another nonparametric curve estimation model [55] attempts to minimize the mean integrated square error in the reconstruction,

$$\min E\left\{\int_0^1 (\hat{f}_n(x) - f(x))^2 \, dx\right\}, \tag{4.8}$$

by estimating Fourier coefficients for a truncated series. This is done with the following data driven estimator

$$\hat{f}_n(x) = \sum_{j=0}^{J_n} (1 - \hat{\theta}_j^{-2} \hat{\sigma}^2 N^{-1}) \tilde{\theta}_j I_{\{|h_j^{\Delta}| > c_H \hat{\sigma} n^{-\frac{1}{2}+b_n}\}} \phi_j(x), \tag{4.9}$$

where ϕ_j are orthogonal basis functions, taken here as the cosine basis, $\tilde{\theta}$ and $\hat{\theta}$ are estimated Fourier coefficients, $\hat{\sigma}$ is a normed sample median, J_n is the truncation parameter, h_j^{Δ} is

Table 4.1: MSVR sampling and signal reconstruction notation

$p(x)$	Reconstructed signal
$\mathbf{a}$	Complex coefficients for trigonometric polynomials
w_j^v	Weighting value for reconstruction algorithm for x_j^v
M	Signal reconstruction bandlimit
L	Domain length over which signal is to be reconstructed
$\mathbf{C}$	Covariance matrix for sampling error terms in aggregated data

the value of the characteristic function of Δ at $j\pi$, and the remaining parameters are data-driven coefficients with values further expanded on in [55].

However, this algorithm still requires a number of arbitrarily selected parameters that the algorithm is particularly sensitive to. This selection often results in inadequate reconstructions, especially when using the parameters in the data-driven estimator to parametrize the model. This model also does not explicitly account for any correlation in the error in variables.

4.1.2 MSVR Sampled Continuous Signal Reconstructions

Refer to Table 4.1 for the signal reconstruction notation used in this section and to Table 3.2 for the previously introduced MSVR sampling notation.

We attempt to reconstruct the signal $f(x)$ as $p(x)$, a sum of trigonometric polynomials [51] with bandlimit M, over a domain of length L. The reconstruction is formulated as

$$p(x) = \sum_{k=-ML}^{ML} a_k e^{\frac{2\pi i k x}{L}}, \tag{4.10}$$

where a_k is the complex coefficient of the complex exponential at frequency $\frac{k}{L}$. The vector of a_k variables, $\mathbf{a}$, is the $(2ML+1)$-dimensional vector which needs to be determined to reconstruct the signal. The continuous signal reconstruction of Equation 4.10 allows us to obtain values of $p(x)$ at any desired location, not just where the sensors carry out sampling.

We determine $\mathbf{a}$ by minimizing a cost function that includes both the sample value error

and sample location error between the samples and the reconstructed signal, weighted by the inverse of the covariance matrix of all the sampling errors. The cost function is also used to estimate the true measurement locations by estimating the sample location errors, $\mathbf{\Delta}$, for each sample. The general framework for the cost function is therefore,

$$\min_{\mathbf{a},\mathbf{\Delta}} \begin{bmatrix} \sqrt{w_1^1}\left(p(q_1^1) - y_1^1\right) \\ \vdots \\ \sqrt{w_{n_V}^V}\left(p(q_{n_V}^V) - y_{n_V}^V\right) \\ \sqrt{w_1^1}\left(x_1^1 - q_1^1\right) \\ \vdots \\ \sqrt{w_{n_V}^V}\left(x_{n_V}^V - q_{n_V}^V\right) \end{bmatrix}^T \mathbf{C}^{-1} \begin{bmatrix} \sqrt{w_1^1}\left(p(q_1^1) - y_1^1\right) \\ \vdots \\ \sqrt{w_{n_V}^V}\left(p(q_{n_V}^V) - y_{n_V}^V\right) \\ \sqrt{w_1^1}\left(x_1^1 - q_1^1\right) \\ \vdots \\ \sqrt{w_{n_V}^V}\left(x_{n_V}^V - q_{n_V}^V\right) \end{bmatrix}, \tag{4.11}$$

where w_j^v is a weighting value based on sampling density for each sample (to be discussed in Section 4.1.2.5) and $\mathbf{C}$ is the covariance matrix which includes the covariance between all the error terms. The matrix $\mathbf{C}$ can be expressed as block matrices,

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{yy} & \mathbf{C}_{yx} \\ \mathbf{C}_{yx}^T & \mathbf{C}_{xx} \end{bmatrix}, \tag{4.12}$$

where $\mathbf{C}_{yy}$ is the covariance matrix of sample value errors, $\mathbf{C}_{yx}$ is the matrix of covariances between sample value errors and sample location errors, and $\mathbf{C}_{xx}$ is the covariance matrix of the sample location errors for the entire data set as given by the MSVR sampling models in Section 3.3.

Empirically determining the complete covariance matrix is impractical and leads to a complicated minimization problem. We therefore make a number of simplifying assumptions on the covariance matrix. These assumptions are based on the MSVR sampling model for vehicles and reducing the cost function in Equation 4.11 to a least-squares model. First, we assume that the sample value errors are all independent from each other. Therefore,

$$\mathbf{C}_{yy} = \sigma_y^2 \mathbb{I}_{N\times N}, \tag{4.13}$$

where $\mathbb{I}_{N\times N}$ is an $N \times N$ identity matrix.

We further assume that the errors between the sample values and sample locations are uncorrelated. Therefore,

$$\mathbf{C}_{yx} = \mathbf{0}. \tag{4.14}$$

This assumption is not always valid. We demonstrate in Section 5.1.5.2 that the measured location values affect how errors are propagated in estimating road angle values. However, without prior knowledge of the final function or how the final data was obtained from the raw data, it is difficult to make any further assumptions on $\mathbf{C}_{yx}$.

The covariance matrix $\mathbf{C}_{xx}$ expressing the covariance between the sampling location errors is adapted from the different MSVR error models given in Section 3.3. The following sections detail how the signal reconstruction algorithms are designed to handle these various MSVR vehicle sampling models.

4.1.2.1 Algorithm 1 - Autocorrelated Source Error

Using the assumptions about the covariance matrix given in Equation 4.12, the general cost function of Equation 4.11 is reduced to the following,

$$\min_{\mathbf{a},\boldsymbol{\Delta}} \sum_{v=1}^{V}\sum_{j=1}^{n_v} w_j^v |p(q_j^v) - y_j^v|^2 + \sigma_y^2 \sum_{v=1}^{V} \begin{bmatrix} \sqrt{w_1^v}(x_1^v - q_1^v) \\ \vdots \\ \sqrt{w_{n_v}^v}(x_{n_v}^v - q_{n_v}^v) \end{bmatrix}^T (\mathbf{C}_{xx}^v)^{-1} \begin{bmatrix} \sqrt{w_1^v}(x_1^v - q_1^v) \\ \vdots \\ \sqrt{w_{n_v}^v}(x_{n_v}^v - q_{n_v}^v) \end{bmatrix}. \tag{4.15}$$

The inverse of the vehicle source error covariance matrix, $(\mathbf{C}_{xx}^v)^{-1}$, is given by the MSVR autocorrelated error model in Equation 3.20. By expanding the covariance matrix further, the minimization target cost function becomes,

$$\begin{aligned} &\min_{\mathbf{a},\boldsymbol{\Delta}} \sum_{v=1}^{V}\sum_{j=1}^{n_v} w_j^v \left|p(q_j^v) - y_j^v\right|^2 \\ &\quad + \sum_{v=1}^{V}\sum_{j=1}^{n_v-1} \frac{\sigma_y^2}{\sigma_a^2} \frac{1}{1-\left(\rho_{j,j+1}^v\right)^2} \left(\sqrt{w_j^v}\Delta_j^v - \sqrt{w_{j+1}^v}\Delta_{j+1}^v\right)^2 + \sum_{v=1}^{V} \frac{\sigma_y^2}{\sigma_a^2} w_{n_v}^v \left(\Delta_{n_v}^v\right)^2. \end{aligned} \tag{4.16}$$

The first term of the cost function represented by Equation 4.16 represents the sample value error. The second term represents the difference in sample location error between consecutive measurements from each individual vehicle. Due to the autocorrelation error model, the error between consecutive terms should be relatively close (expressed by the exponential autocorrelation function). The third term minimizes the sample location error of the final measurement from each vehicle. This acts as a reference for all the other sample location errors from that vehicle. Without this term, the location error from all the other samples may be close to equal in accordance with the autocorrelation model, but far exceed the expected magnitude of the errors. The final term helps to regularize the estimated sample location error.

The cost function represented by Equation 4.16 is minimized with respect to coefficient vector $\mathbf{a}$ and sampling location offset vector $\boldsymbol{\Delta}$. We use the Levenberg-Marquardt algorithm (LMA) [56, 57] with damping and stopping conditions from [58] for this task. The LMA iteratively minimizes $S(\boldsymbol{\beta})$, the sum of squares of m functions, $\mathbf{r} = \begin{bmatrix} r_1 & \dots & r_m \end{bmatrix}^T$ of n variables $\boldsymbol{\beta} = \begin{bmatrix} \beta_1 & \dots & \beta_n \end{bmatrix}^T$, where

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{m} r_i(\boldsymbol{\beta})^2. \tag{4.17}$$

The LMA update involves solving the following linear equation,

$$\left(\mathbf{J}^T\mathbf{J} + \lambda \text{diag}(\mathbf{J}^T\mathbf{J})\right)(\boldsymbol{\beta}^{(s)} - \boldsymbol{\beta}^{(s+1)}) = \mathbf{J}^T\mathbf{r}(\boldsymbol{\beta}^{(s)}), \tag{4.18}$$

where the superscript s denotes the iteration index and $\mathbf{J}$ is the Jacobian matrix for $\mathbf{r}(\boldsymbol{\beta})$. The LMA is a standard method of solving nonlinear least-squares problems, and is an adaptation of the Gauss-Newton algorithm with damping parameter $\lambda \text{diag}(\mathbf{J}^T\mathbf{J})$. For the minimization of the cost function in Equation 4.16,

$$\boldsymbol{\beta} = \begin{bmatrix} \Delta_1^1 & \dots & \Delta_{n_1}^1 & \dots & \Delta_1^V & \dots & \Delta_{n_V}^V & \mathbf{a}^T \end{bmatrix}^T \tag{4.19}$$

is the $(N + 2ML + 1) \times 1$ vector of variables to be solved for, where

$$N = \sum_{v=1}^{V} n_v. \tag{4.20}$$

The $2N$-dimensional vector of cost functions is

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1^{1T} & \dots & \mathbf{r}_1^{VT} & \mathbf{r}_2^{1T} & \dots & \mathbf{r}_2^{VT} & r_3^1 & \dots & r_3^V \end{bmatrix}^T. \tag{4.21}$$

The vector $\mathbf{r}_1^v$ is the $n_v \times 1$ vector of cost functions for the sample value error from source v consisting of elements,

$$r_{1\,j}^v = \sqrt{w_j^v} \left(\sum_{k=-ML}^{ML} a_k e^{\frac{2\pi i k \left(x_j^v - \Delta_j^v \right)}{L}} - y_j^v \right). \tag{4.22}$$

The vector $\mathbf{r}_2^v$ is the $(n_v - 1) \times 1$ vector of cost functions for the difference in consecutive location error terms for source v, where

$$r_{2\,j}^v = \sqrt{\frac{\sigma_y^2}{\sigma_a^2 (1 - (\rho_{j,j+1}^v)^2)}} \left(\sqrt{w_j^v} \Delta_j^v - \sqrt{w_{j+1}^v} \Delta_{j+1}^v \right). \tag{4.23}$$

The scalar cost functions r_3^v for the final location error term for source v are equal to

$$r_3^v = \sqrt{\frac{\sigma_y^2}{\sigma_a^2} w_{n_v}^v} \Delta_{n_v}^v. \tag{4.24}$$

The Jacobian matrix $\mathbf{J}$ is a $(2N) \times (N + 2ML + 1)$ sized matrix composed of the following block matrices,

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{A} \\ \mathbf{J}_2 & \mathbf{0} \\ \mathbf{J}_3 & \mathbf{0} \end{bmatrix}. \tag{4.25}$$

The matrix $\mathbf{J}_1$ is the $N \times N$ block diagonal Jacobian matrix for the sample value error cost functions with respect to the $\mathbf{\Delta}$ location error variables such that

$$\mathbf{J}_1 = \begin{bmatrix} \mathbf{J}_1^1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_1^2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}_1^V \end{bmatrix}, \tag{4.26}$$

where $\mathbf{J}_1^v$ is a $n_v \times n_v$ diagonal Jacobian matrix for source v such that

$$J^v_{1\,jj} = -\sqrt{w_j^v}\left(\sum_{k=-ML}^{ML} a_k \frac{2\pi i k}{L} e^{\frac{2\pi i k\left(x_j^v - \Delta_j^v\right)}{L}}\right). \tag{4.27}$$

The Jacobian matrix $\mathbf{J}_2$ is a $(N-V)\times N$ block diagonal matrix such that,

$$\mathbf{J}_2 = \begin{bmatrix} \mathbf{J}_2^1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_2^2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}_2^V \end{bmatrix}, \tag{4.28}$$

where $\mathbf{J}_2^v$ is an $(n_v - 1)\times n_v$ sized matrix with non-zero entries on only the main diagonal and superdiagonal, given by

$$J^v_{2\,j,j} = \sqrt{\frac{\sigma_y^2 w_j^v}{\sigma_a^2(1-(\rho^v_{j,j+1})^2)}}, \tag{4.29}$$

$$J^v_{2\,j,j+1} = -\sqrt{\frac{\sigma_y^2 w_{j+1}^v}{\sigma_a^2(1-(\rho^v_{j,j+1})^2)}}. \tag{4.30}$$

Finally, $\mathbf{J}_3$ is also a block diagonal matrix such that,

$$\mathbf{J}_3 = \begin{bmatrix} \mathbf{J}_3^1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_3^2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}_3^V \end{bmatrix}, \tag{4.31}$$

where the block matrices $\mathbf{J}_3^v$ are $1 \times n_v$ sized matrices with all zero entries except for

$$J^v_{3\,1,n_v} = \sqrt{\frac{\sigma_y^2}{\sigma_a^2} w^v_{n_v}}. \tag{4.32}$$

The matrix $\mathbf{A}$ below is the $N \times (2ML+1)$ Jacobian matrix for the sample value cost functions with respect to the coefficient vectors $\mathbf{a}$, composed of the following block matrices,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^1 \\ \vdots \\ \mathbf{A}^V \end{bmatrix}, \tag{4.33}$$

such that $\mathbf{A}^v$ is the $n_v \times (2ML+1)$ Jacobian matrix for source v composed of elements

$$A^v{}_{jl} = \sqrt{w_j^v} e^{\frac{2\pi i(l-ML)\left(x_j^v - \Delta_j^v\right)}{L}}. \tag{4.34}$$

The initial estimate for the sampling location offsets and coefficient variables, $\boldsymbol{\beta}^{(0)}$, is

$$\boldsymbol{\beta}^{(0)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{a}^{(0)} \end{bmatrix}, \tag{4.35}$$

where $\mathbf{a}^{(0)}$ is the closed-form solution vector to the signal reconstruction model in Section 4.1.2.4 that is derived from ignoring the errors in variables.

4.1.2.2 Algorithm 2 - Correlated Source Error

For the Correlated Source Error signal reconstruction mode, the $\mathbf{C}_{xx}$ covariance matrix for the cost function in Equation 4.12 is constructed using the MSVR correlated source error sampling model in Section 3.3.2. Similar to the autocorrelated source error model, the general cost function in Equation 4.11 is expanded to produce the form of Equation 4.15. However for this error model, the matrix $\mathbf{C}_{xx}^v$ is given by the correlated source error model in Equation 3.24. The covariance matrices allow the minimization of Equation 4.15 to be reformulated as

$$\begin{aligned} \min_{\mathbf{a},\boldsymbol{\Delta}} \sum_{v=1}^{V} \sum_{j=1}^{n_v} \left[w_j^v \left| p(q_j^v) - y_j^v \right|^2 + \sigma_y^2 (A_v - B_v) w_j^v (\Delta_j^v)^2 \right] \\ + \sum_{v=1}^{V} \left[\sigma_y^2 B_v \left(\sum_{j=1}^{n_v} \sqrt{w_j^v} \Delta_j^v \right)^2 \right]. \end{aligned} \tag{4.36}$$

The first term of the cost function represents the sample value error. The second term represents the sample value error. The third term in the cost function is a corrective term that is related to the sum of the location errors from all the samples from each vehicle, since due to the correlated error model, all sample location error terms from a vehicle are expected to be similar.

The LMA iterative solution is used to minimize the cost function in Equation 4.36. The vector $\boldsymbol{\beta}$ of variables used in the minimization is the same as that in Equation 4.19 for the autocorrelated error model. The $(2N + V)$-dimensional vector of cost functions is composed as,

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1^{1T} & \dots & \mathbf{r}_1^{VT} & \mathbf{r}_2^{1T} & \dots & \mathbf{r}_2^{VT} & r_3^1 & \dots & r_3^V \end{bmatrix}^T. \tag{4.37}$$

The vector $\mathbf{r}_1^v$ is the $n_v \times 1$ vector of cost functions for the sample value error from source v, with elements given by Equation 4.22.

The $n_v \times 1$ sized $\mathbf{r}_2^v$ vector for the second term of Equation 4.36 is defined as having the following components,

$$r_{2j}^v = \sqrt{\sigma_y^2 (A_v - B_v) w_j^v}\ \Delta_j^v. \tag{4.38}$$

The scalar cost functions r_3^v for the third term of Eq. 4.36 are,

$$r_3^v = \sqrt{\sigma_y^2 B_v} \sum_{j=1}^{n_v} \sqrt{w_j^v}\ \Delta_j^v. \tag{4.39}$$

Similar to the autocorrelated error model signal reconstruction, the required Jacobian matrix from the LMA algorithm, $\mathbf{J}$, is composed of the following block matrices,

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{A} \\ \mathbf{J}_2 & \mathbf{0} \\ \mathbf{J}_3 & \mathbf{0} \end{bmatrix}. \tag{4.40}$$

The matrices $\mathbf{J}_1$ and $\mathbf{A}$ are defined (identically to the Autocorrelated Source Error algorithm) by Equations 4.26 and 4.33 respectively.

The matrix $\mathbf{J}_2$ is constructed from V $N \times N$ block diagonal matrices similar to Equation 4.28, however in the correlated source error model $\mathbf{J}_2^v$ is a diagonal matrix with diagonal elements given by,

$$J_{2\,jj}^v = \sqrt{\sigma_y^2 (A_v - B_v) w_j^v}. \tag{4.41}$$

The matrix $\mathbf{J}_3$ is a $V \times N$ block diagonal matrix similar to Equation 4.31, however the composition block matrices $\mathbf{J}_3^v$ are of size $1 \times n_v$ and composed of elements

$$J_{3\,1j}^v = \sqrt{\sigma_y^2 B_v w_j^v}. \tag{4.42}$$

4.1.2.3 Algorithm 3 - Fixed Source Error

To simplify the error model and reduce the dimensionality of the problem space, we use the MSVR sampling assumptions outlined in Section 3.3.3 where the sample location errors from all samples from a single sensor source are assumed to be equal, but the errors are i.i.d. for the different sources. This assumption reduces the number of location error variables to estimate from N to V, which can be computationally advantageous.

In this formulation, the covariance matrix $\mathbf{C}$ in Equation 4.12 is replaced by

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{yy} & \mathbf{C}_{yv} \\ \mathbf{C}_{yv} & \mathbf{C}_{vv} \end{bmatrix}, \tag{4.43}$$

where $\mathbf{C}_{vv}$ is given by Equation 3.31 from the MSVR model. We still assume that the sample errors and location errors are independent, so similar to Equation 4.14, $\mathbf{C}_{yv} = \mathbf{0}$. By expanding the covariance matrix, we arrive at the following least-squares problem,

$$\min_{\mathbf{a},\boldsymbol{\Delta}} \sum_{v=1}^{V} \sum_{j=1}^{n_v} w_j^v \left| p(q_j^v) - y_j^v \right|^2 + \sum_{v=1}^{V} \left[\left(\sum_{j=1}^{n_v} w_j^v \right) \frac{\sigma_y^2}{\sigma_v^2} \left(\Delta^v \right)^2 \right]. \tag{4.44}$$

To apply the Levenberg-Marquardt algorithm, we define the $(V + 2ML + 1) \times 1$ vector of variables as

$$\boldsymbol{\beta} = \begin{bmatrix} \Delta^1 & \dots & \Delta^V & \mathbf{a}^T \end{bmatrix}^T, \tag{4.45}$$

and we define $\mathbf{r}$ as the following $(N+V)\times 1$ vector of cost functions in terms of the $(V+2ML+1)$ $\boldsymbol{\beta}$ variables,

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1^{1T} & \dots & \mathbf{r}_1^{VT} & r_2^1 & \dots & r_2^V \end{bmatrix}^T. \tag{4.46}$$

In this formulation, $\mathbf{r}_1^v$ is a $n_v \times 1$ vector for the first term of Equation 4.44 with components,

$$r_{1j}^v = \sqrt{w_j^v}\left(\sum_{k=-ML}^{ML} a_k e^{\frac{2\pi ik\left(x_j^v-\Delta^v\right)}{L}} - y_j^v\right). \tag{4.47}$$

The scalar cost functions r_2^v for the second term of Equation 4.44 are defined as

$$r_2^v = \sqrt{\frac{\sigma_y^2}{\sigma_v^2}\sum_{j=1}^{n_v} w_j^v}\ \Delta^v. \tag{4.48}$$

The Jacobian $\mathbf{J}$ is a $(N+V)\times(V+2ML+1)$ matrix composed of the following block matrices,

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{A} \\ \mathbf{J}_2 & \mathbf{0} \end{bmatrix}, \tag{4.49}$$

where $\mathbf{J}_1$ is the $N \times V$ Jacobian matrix for the $\mathbf{r}_1$ cost functions with respect to the $\boldsymbol{\Delta}$ variables, composed of block matrices similar to Equation 4.26, however where $\mathbf{J}_1^v$ are $n_v \times 1$ Jacobian matrices for each source composed of elements,

$$J_{1\,j1}^v = -\sqrt{w_j^v}\left(\sum_{k=-ML}^{ML} a_k \frac{2\pi ik}{L} e^{\frac{2\pi ik\left(x_j^v-\Delta^v\right)}{L}}\right). \tag{4.50}$$

$\mathbf{J}_2$ is a $V \times V$ diagonal matrix where,

$$J_{2vv} = \sqrt{\frac{\sigma_y^2}{\sigma_v^2}\sum_{j=1}^{n_v} w_j^v}. \tag{4.51}$$

The Jacobian matrix $\mathbf{A}$ is defined by Equation 4.33.

Although this significantly reduces the problem dimension compared to the previous two models, this remains an iterative solution. By ignoring the sampling location error, we can derive a closed-form solution (and initial estimate for the iterative solutions) for the frequency coefficient vector $\mathbf{a}$.

4.1.2.4 Algorithm 4 - Sample Value Error Only

The sample location error is neglected in this signal reconstruction model. This assumption ignores a major component of the reconstruction, however it allows us to derive a closed form solution that can be used as an initial estimate for the iterative solutions of the more complete models. With no sample location error,

$$\mathbf{C}_{xx} = \mathbf{0}. \tag{4.52}$$

This eliminates the requirement of solving for location offsets, $\mathbf{\Delta}$, in the optimization, and simplifies the cost function to a least-squares summation over just the sample value error terms,

$$\min_{\mathbf{a}} \sum_{v=1}^{V} \sum_{j=1}^{n_v} w_j^v \left| p(x_j^v) - y_j^v \right|^2. \tag{4.53}$$

This is a version of the cost functions in [51], generalized to multiple sources. We follow a similar derivation to their closed form solution. The least-squares problem of Equation 4.53 is a standard quadratic minimization and can be shown to lead to the following closed-form solution. First the cost function is expanded as,

$$\min_{\mathbf{a}} \sum_{v=1}^{V} \sum_{j=1}^{n_v} \left[\left(\sum_{k=-ML}^{ML} a_k e^{\frac{2\pi i k x_j^v}{L}} - y_j^v \right) \left(\sum_{k=-ML}^{ML} a_k^* e^{\frac{-2\pi i k x_j^v}{L}} - y_j^v \right) w_j^v \right] \tag{4.54}$$

$$\begin{aligned} = \min_{\mathbf{a}} \sum_{v=1}^{V} \sum_{j=1}^{n_v} w_j^v \Bigg[& \sum_{k=-ML}^{ML} \sum_{l=-ML}^{ML} a_k a_l^* e^{\frac{2\pi i (k-l) x_j^v}{L}} - \\ & y_j^v \left(\sum_{k=-ML}^{ML} \left(a_k e^{\frac{2\pi i k x_j^v}{L}} + a_k^* e^{\frac{-2\pi i k x_j^v}{L}} \right) \right) + \left(y_j^v \right)^2 \Bigg]. \end{aligned} \tag{4.55}$$

Differentiating with respect to a_k, equating to zero, and taking the complex conjugate results in the following equations,

$$\sum_{v=1}^{V} \sum_{j=1}^{n_v} w_j^v \left[\sum_{l=-ML}^{ML} a_l^* e^{\frac{2\pi i (k-l) x_j^v}{L}} - y_j^v e^{\frac{2\pi i k x_j^v}{L}} \right] = 0, \tag{4.56}$$

$$\sum_{v=1}^{V}\sum_{j=1}^{n_v} w_j^v y_j^v e^{\frac{2\pi i k x_j^v}{L}} = \sum_{v=1}^{V}\sum_{j=1}^{n_v} w_j^v \sum_{l=-ML}^{ML} a_l^* e^{\frac{2\pi i (k-l) x_j^v}{L}}, \tag{4.57}$$

$$\sum_{v=1}^{V}\sum_{j=1}^{n_v} w_j^v y_j^v e^{\frac{-2\pi i k x_j^v}{L}} = \sum_{l=-ML}^{ML} a_l \sum_{v=1}^{V}\sum_{j=1}^{n_v} w_j^v e^{\frac{2\pi i (l-k) x_j^v}{L}}. \tag{4.58}$$

For the terms on the left side of Equation 4.58, define a $(2ML+1) \times 1$ vector, $\mathbf{b}$, with components,

$$b_k = \sum_{v=1}^{V}\sum_{j=1}^{n_v} w_j^v y_j^v e^{\frac{-2\pi i k x_j^v}{L}}, \tag{4.59}$$

and define a $(2ML+1) \times (2ML+1)$ Toeplitz matrix, $\mathbf{T}$, for the terms on the right side of Equation 4.58, with the following components,

$$T_{lk} = \sum_{v=1}^{V}\sum_{j=1}^{n_v} w_j^v e^{\frac{2\pi i (l-k) x_j^v}{L}}. \tag{4.60}$$

Then the solution for the signal reconstruction coefficient vector $\mathbf{a}$ satisfies the linear equation,

$$b_k = \sum_{l=-ML}^{ML} a_l T_{lk}, \tag{4.61}$$

$$\mathbf{b} = \mathbf{T}\hat{\mathbf{a}}. \tag{4.62}$$

Toeplitz matrix system solutions [59] are available as efficient methods to solve Equation 4.62 for $\hat{\mathbf{a}}$. The solution to Equation 4.62 is used as the initial estimate for $\mathbf{a}^{(0)}$ in Equation 4.35 for the previous signal reconstruction algorithms.

4.1.2.5 Sample Weighting

The weight w_j^v corresponding to the sample at x_j^v for the reconstruction algorithms is defined as the size of the Voronoi region [60, 61] over all N aggregated samples in the data set. Let x_k, $k \in \{1, \ldots, N\}$, correspond to the sample x_j^v in the sorted data set. For a

one-dimensional domain, the Voronoi weight, w_j^v, is therefore,

$$w_j^v = w_k = \frac{x_{k+1} - x_{k-1}}{2}. \tag{4.63}$$

The weights compensate for the local variations in sampling density caused by MSVR sampling conditions by assigning higher weights to samples in lower density regions [52]. In lower density regions, there are fewer samples in the region that provide information for the reconstruction. Therefore it is important that the reconstruction is closer to those respective samples than for those in higher density regions where the noise characteristics are more clearly manifested, and matching two close noisy samples exactly can produce poor results.

For reconstruction Algorithms 1-3 that require an iterative solution, we interrupt the algorithm between each iteration to recalculate the sample weights based on the newly estimated $\hat{q}_j^v$ locations. This is done under the assumption that the estimated locations are closer to the ground-truth than the original x_j^v measurements, so the weighting scheme should reflect the ground-truth locations as accurately as possible. We demonstrate the importance of reweighting the samples between LMA iterations by reconstructing the following example signal,

$$\begin{aligned} f(x) = &\sin(2\pi(0.132695)x + 2.1) + 0.3\sin(2\pi(0.21)x + 0.5) \\ &+ 1.1\sin(2\pi(0.325869)x + 0.3) + 0.4\sin(2\pi(1.612)x + 0.1). \end{aligned} \tag{4.64}$$

The sampling noise and reconstruction parameters are given in Table 4.2. We use Algorithm 3 for the signal reconstructions by both reweighting samples between iterations and by using only the originally calculated weights based on the x_j^v locations. The resulting reconstructions are shown in Figure 4.2. This figure demonstrates that the iteratively reweighted scheme produces the better signal reconstruction. This is supported by the RMS errors for the two reconstructions, which are 0.19 for the reweighting scheme and 0.55 for the non-reweighting scheme, which is a relatively large difference given that the RMS signal value is 1.14.

Table 4.2: Reconstruction and sampling parameters for the sample reweighting experiment with results in Figure 4.2

Parameter	Value
M	$\frac{40}{L}$
σ_v	0.7
σ_x	0.1
σ_y	0.05
V	100
N	2015
Source sample spacing	Drawn from Rayleigh dist., mean 1

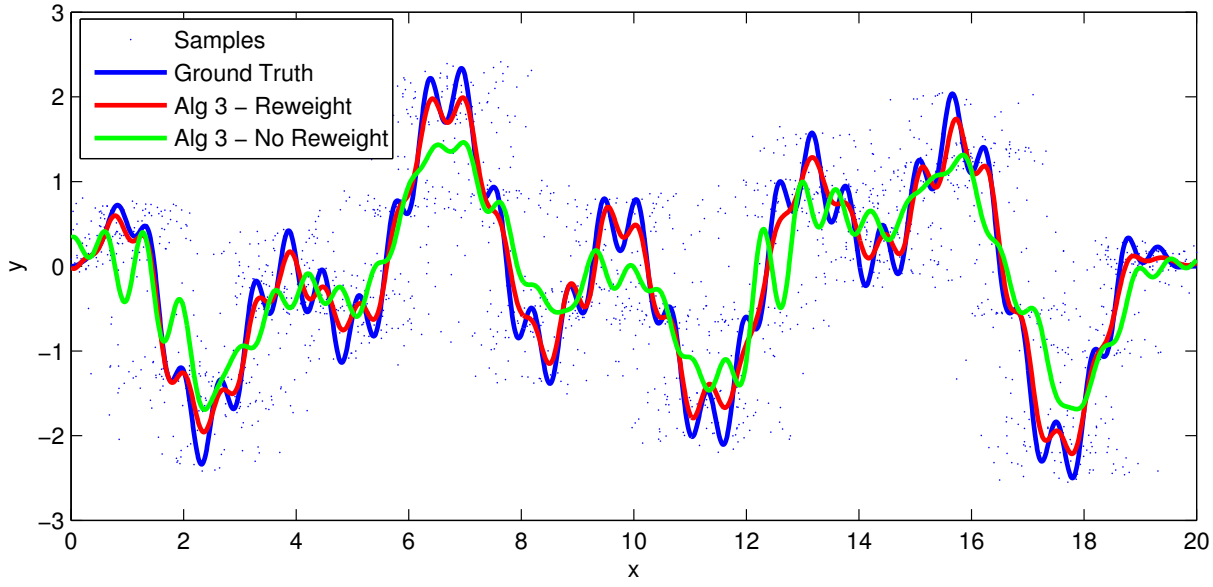


Figure 4.2: Example signal reconstructions, using the conditions in Table 4.2, for both recalculating the signal reconstruction sample weights and not recaulculating the sample weights between LMA iterations

4.1.2.6 Reconstruction Analysis

We first compare the MSVR signal reconstruction algorithms on an example road bank angle signal shown in Figure 4.3. The signal will be further explained in Section 5.1.4 with sampling details in Section 5.1.3. However at this stage it is sufficient to assume that samples are drawn from the signal by vehicles traveling at an average of 50 km/h, with error statistics to be specified.

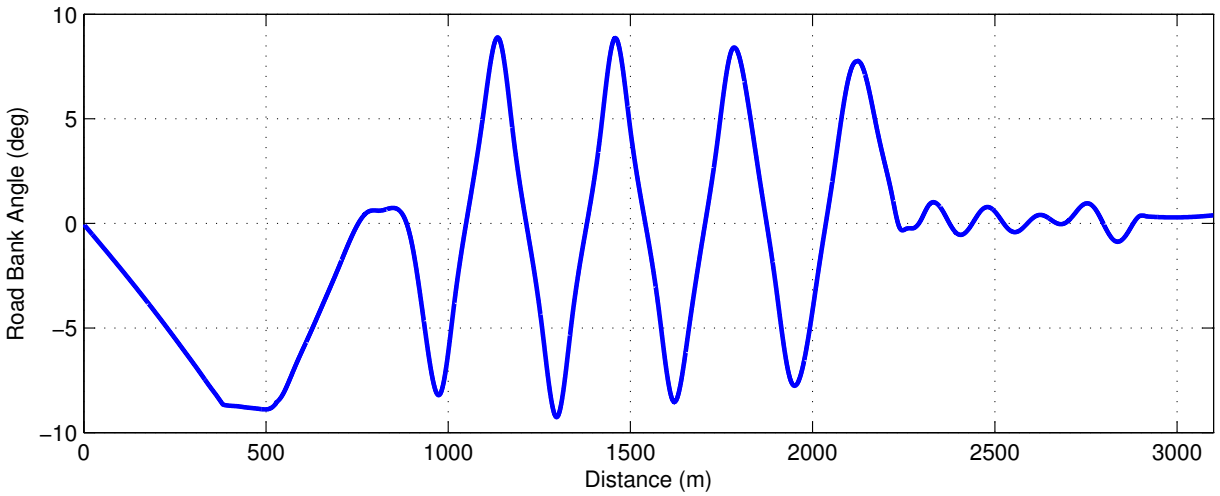


Figure 4.3: Ground-truth simulated road bank angle signal for reconstruction comparisons

One example set of signal reconstructions for $V = 50$, $M = \frac{30}{L}$, $\sigma_y = 0.005°$, $\sigma_v = 20$ m, and $\sigma_x = 40$ m is shown in Figure 4.4. The RMSE values along with those for reconstructions from other MSVR sampling conditions are given in Table 4.3. The effectiveness of each signal reconstruction algorithm depends on the MSVR sampling conditions. For the reconstruction in Figure 4.4, $\sigma_x^2 \gg \sigma_v^2$. Therefore the reconstruction from Algorithm 2 is able to best account for the noise conditions as the independent error for each sample location dominates the source location bias. However, the reconstruction from Algorithm 3 still outperforms Algorithm 4 as the location from each source is still partially corrected.

The gains that the reconstruction from Algorithm 2 have over Algorithm 3 are decreased as the sampling location source bias error becomes a more dominating factor over the independent location error component. In fact for $\sigma_x^2 \ll \sigma_v^2$, Algorithm 3 outperforms Algorithm 2 for the given conditions. Since the number of variables to be estimated in Algorithm 2 is significantly greater than for Algorithm 3, Algorithm 2 can be prone to overfitting the sample noise when the independent error component is not dominant. However, it still outperforms Algorithm 4, which does not correct for location error. When sample value error is the dominant error source, as given by the final example in Table 4.3, correcting for the location error does not offer significant benefits. Considering that Algorithm 2 has

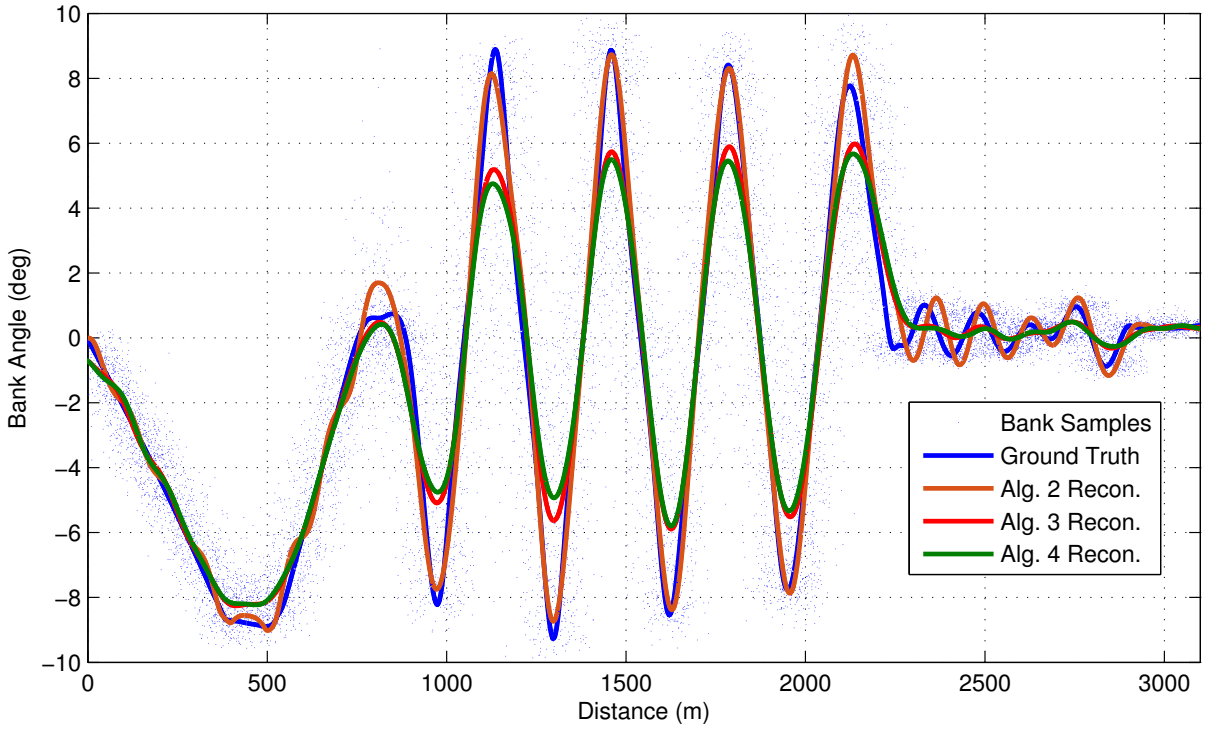


Figure 4.4: Comparison of bank angle signal using three MSVR signal reconstructions for $M = \frac{30}{L}, V = 50, \sigma_y = 0.005, \sigma_v = 20, \sigma_x = 40$

the longest runtime and uses the most memory of these three algorithms (more variables to estimate), the relative sizes of the different error components is an important consideration to determine which algorithm offers the best and most efficient reconstruction.

To further demonstrate the importance of the MSVR error models, and to compare against existing work [62], signal reconstructions are done on the following six synthesized functions, similar to those examples used in [55]. The functions are chosen for their varying smoothness properties, all defined on the domain $x \in [0, 1]$, and note that $P_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-(x-\mu)^2/(2\sigma^2)\right)$:

a) *Gaussian* - $f_a(x) = P_{0.5,0.15}(x)$

b) *Bimodal* - $f_b(x) = 0.5P_{0.4,0.12}(x) + 0.5P_{0.7,0.08}(x)$

c) *Strata* - $f_c(x) = 0.5P_{0.2,0.06}(x) + 0.5P_{0.7,0.08}(x)$

Table 4.3: RMSE values for reconstructing the road incline angle signal for the CarSim® data, $M = \frac{30}{L}, V = 50, N = 15109$

Conditions			**RMSE (deg)**		
σ_y (°)	σ_v (m)	σ_x (m)	**Alg. 2**	**Alg. 3**	**Alg. 4**
0.005	20	40	0.627	1.117	1.251
0.005	30	30	0.678	0.841	1.206
0.005	40	20	0.882	0.656	1.392
0.2	5	5	0.293	0.294	0.295

d) *Full Period Sine* - $f_d(x) = 0.3\sin(2\pi x - 0.654) + 0.7\sin(4\pi x + 2.1)$

e) *Partial Period Sine* - $f_e(x) = 0.3\sin(2\pi(0.912)x - 0.654) + 0.7\sin(2\pi(1.513)x - 0.294) + \sin(2\pi(1.765)x + 2.1)$

f) *Sine and Polynomial* - $f_f(x) = 0.5\sin(6\pi x - 0.215) + 5(x - 0.5)^2 - 1$

For each function, data was simulated to be sampled from 20 mobile sensors with a spacing between samples from each sensor drawn from a Rayleigh distribution with mean 0.05. The time interval between measurements was distributed as $\mathcal{N}(1, 0.05)$. Under these conditions, the aggregated sample distribution approaches a uniform distribution as the number of sensors increases. The sample value error was drawn i.i.d. from a normal distribution with standard deviation $\sigma_y = 0.1$. The location error was drawn from an exponential autocorrelation function independently for each sensor with time constant $T = 50$ and standard deviation $\sigma_a = 0.1$. These errors are relatively large given the domain length and the period of the signals.

We reconstruct these signals using Algorithms 1 and 2 from our MSVR reconstructions, along with the Kernel Regression (KR) algorithm outlined in Section 4.1.1.3 [54], and the Nonparametric Curve Estimation (NPE) algorithm outlined in the same section [55]. The median normalized reconstruction error, $\frac{\|f(x) - \hat{f}(x)\|_2}{\|f(x)\|_2}$, for the different reconstructions for 10000 randomized iterations of these reconstructions is given in Table 4.4. An example of one such set of reconstructions is shown in Figure 4.5.

Table 4.4: Normalized reconstruction errors, $\frac{\|f(x)-\hat{f}(x)\|_2}{\|f(x)\|_2}$

	KR	NCE	Alg. 2	Alg. 1
f_a	0.2116	0.1948	0.0868	0.0843
f_b	0.2936	0.2837	0.1918	0.1476
f_c	0.4299	0.4752	0.5070	0.4280
f_d	0.8443	0.5866	0.2938	0.2882
f_e	0.6612	0.5480	0.5251	0.3563
f_f	0.5861	0.3930	0.4146	0.2870

The reconstructions using Algorithm 1 have the lowest error for each of the reconstructions. This is because the model uses the MSVR error model information in the reconstruction when trying to determine the true location of the sample values. The existing algorithms do not incorporate such information. Algorithm 1 also estimates the true locations of each sample and reduces the median RMS value of the sample location error by between 7.4%-44% for the given signals and associated large error values. Algorithm 2 generally creates the second best reconstruction among the given algorithms since it still corrects for the location error. However, the sample location error covariance matrix for Algorithm 2 only approximates that of the exponential autocorrelation function and overcompensates with too small correlations for closely timed data samples and too large correlations for more distantly timed data samples.

Our MSVR signal reconstruction algorithms use the signal information to estimate the true location of each sample. However for GPS applications, map information (i.e., locations of roads) can be used to directly estimate the true measurement locations [63]. Such map matching algorithms often perform best when the location error is independent, or on roads with known landmark locations (e.g., stop signs) or variations in road topography (i.e., varying curvature) to provide reference locations. However, the use of either map matching algorithms or our signal based algorithm to estimate the true measurement locations does not exclude using the other algorithm, and it is likely that estimations from

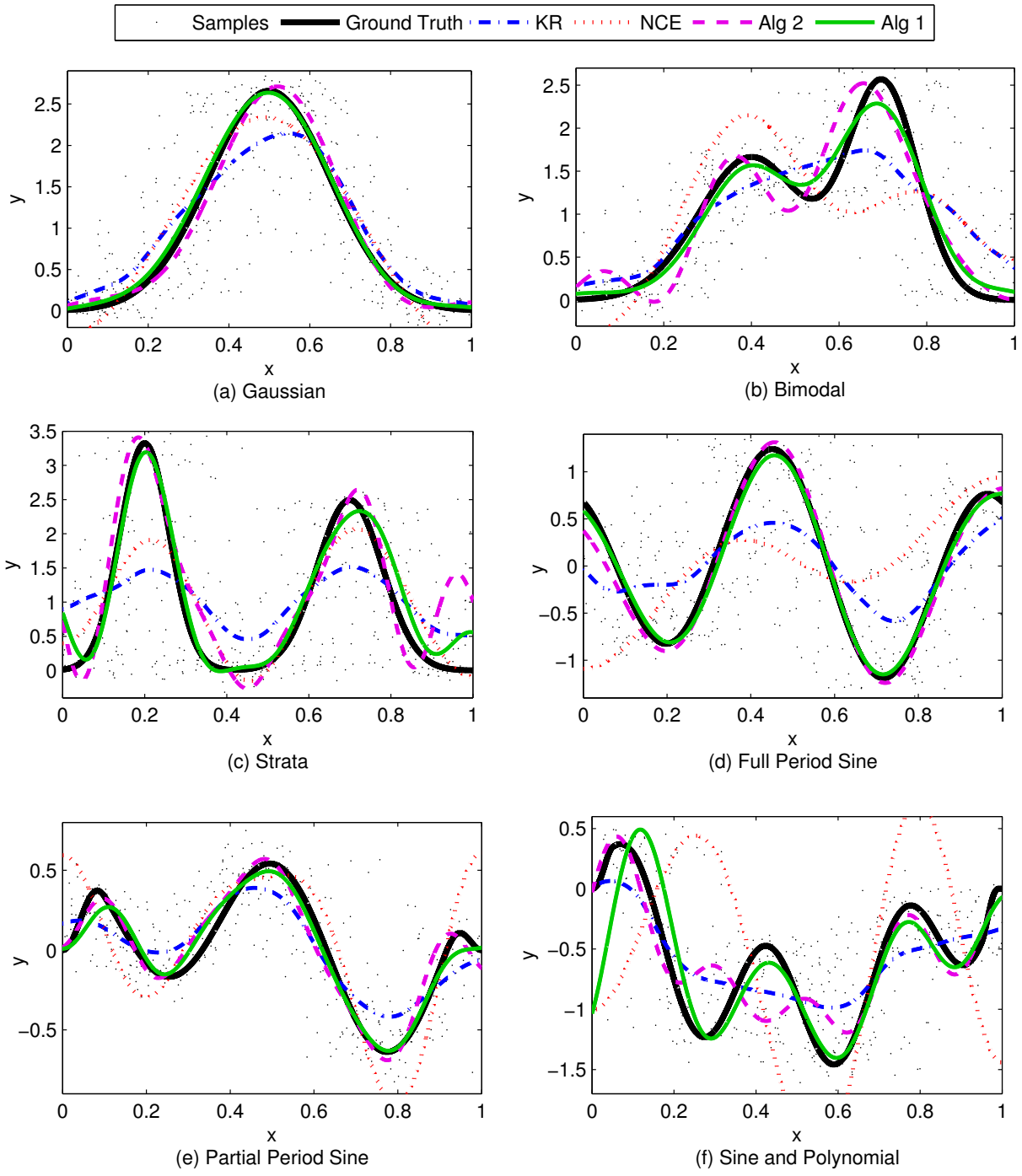


Figure 4.5: Comparison of our Algorithm 1 and Algorithm 2 against existing signal reconstruction algorithms for six example signals

one algorithm could be used to enhance the other.

4.1.3 Joint Reconstructions

Measurements from different sensors in the vehicle are all spatially indexed by the GPS measurements at the respective sampling instances. Data from two different sensors made in the same time interval (which may be relatively long given the low sampling frequency of standard GPS receivers) are therefore indexed to the same location. The signal reconstructions from the previous section (for all but Algorithm 4) estimate the true measurement locations, q_j^v, jointly with the reconstruction parameters, $\mathbf{a}$, for each signal reconstruction. Therefore when reconstructing signals for G different events using data from common vehicles, the G different signal reconstructions each produce independent estimates of the true measurement locations, even if the data samples are indexed by the same GPS measurements. This can result in incompatible location estimations between the two reconstructions.

Since both the true and measured GPS locations are identical for each of the signals, it would be beneficial to jointly reconstruct the two signals, using the source and location information from both signals to create only a single estimate for the true measurement locations. This offers two benefits in the signal reconstructions:

1. The number of variables to estimate is reduced since the true location measurements are only being estimated once for all signals together instead of for each signal independently.

2. The information from one signal reconstruction can be used for the benefit of others. The LMA iterations involve shifting the sampled data locations to minimize both the location and sample value error in the signal reconstruction. This relies on examining correlations between the distances to the signal for the samples from each vehicle according to the MSVR models. With sufficient location noise, the original

Table 4.5: Notation for joint signal reconstructions

G	Number of signals being jointly reconstructed
N_g	Total number of aggregated samples for signal g.
n_{vg}	Total number of samples from source v for signal g
$x^v_{g_j}$	Measured location of data sample j from source v for signal g
$q^v_{g_j}$	True location of data sample j from source v for signal g
$y^v_{g_j}$	Measured sensor value for measured location $x^v_{g_j}$
$p_g(x)$	Reconstructed signal g
$w^v_{g_j}$	Weighting value for reconstruction algorithm for $x^v_{g_j}$
$\sigma^2_{y_g}$	Variance for sample value error for signal g
M_g	Signal reconstruction bandlimit for signal g
$\mathbf{a}_g$	Exponential frequency coefficients for reconstructing signal g

signal reconstruction estimate for the first LMA iteration generally has a flatter slope than the ground truth. Therefore reconstructions from early LMA iterations may not have sufficiently well estimated the higher frequency coefficients for the location error correlations to be truly realized. However by combining the information from multiple signals, the correlations may become more obvious and signals with poor original estimates can be more accurately reconstructed.

In the following subsections we demonstrate how to jointly optimize a set of G signals. Section 4.1.3.1 details the algorithmic steps for the Autocorrelated error model and Section 4.1.3.2 details the algorithm for the Correlated Source Error model. Section 4.1.3.3 explains how to perform the joint reconstruction when only the vehicle error bias is considered. This follows into Section 4.1.3.4, which explains how the joint reconstruction for the fixed error model can be generalized to an asynchronous sampling model where different data sensors in the vehicle index their samples at different locations (due to varying sampling rates and asynchronous sensor operation). The modified notation for these signal reconstructions is shown in Table 4.5. Any omission of the g subscript, where $g \in \{1, \ldots, G\}$ implies that the respective variable is identical for each of the G signals.

4.1.3.1 Autocorrelated Source Error

The cost function for the joint optimization is created using the errors weighted by the inverse of the covariance matrix for the sample and location error for all the signals similar to Equation 4.11. The simplified expansion is similar to the summation of G versions of the cost functions of the form of Equation 4.16, with slight modifications to account for the different possible sample value error variances and frequency coefficient vectors $\mathbf{a}_g$. The components for the cost functions relating to the location error, however, are identical for all the signals since the samples are the same for all signals. This summation results in the following least-squares cost function for jointly reconstructing multiple signals,

$$\begin{aligned}\min_{\mathbf{a}_1,\ldots,\mathbf{a}_G,\boldsymbol{\Delta}} &\sum_{g=1}^{G}\sum_{v=1}^{V}\sum_{j=1}^{n_v}\left[w_j^v\frac{\sigma_a^2}{\sigma_{y_g}^2}\left|p_g(q_j^v)-y_{g_j}^v\right|^2\right] \\ &+G\sum_{v=1}^{V}\sum_{j=1}^{n_v-1}\frac{1}{1-(\rho_{j,j+1}^v)^2}\left(\sqrt{w_j^v}\Delta_j^v-\sqrt{w_{j+1}^v}\Delta_{j+1}^v\right)^2+G\sum_{v=1}^{V}w_{n_v}^v(\Delta_{n_v}^v)^2.\end{aligned} \tag{4.65}$$

This formulation reduces the system from a total of $GN+\sum_{g=1}^{G}(2M_gL+1)$ variables for if the signals were reconstructed independently, to $N+\sum_{g=1}^{G}(2M_gL+1)$ variables for the joint reconstruction, given by

$$\boldsymbol{\beta}=\left[\Delta_1^1\ \ldots\ \Delta_{n_1}^1\ \ldots\ \Delta_1^V\ \ldots\ \Delta_{n_V}^V\ \ \mathbf{a}_1^T\ldots\mathbf{a}_G^T\right]^T. \tag{4.66}$$

The derivation of the cost functions and Jacobian matrices for the LMA algorithm for minimizing the cost function in Equation 4.65 to jointly reconstruct the signals is provided in Appendix A.1.

4.1.3.2 Correlated Source Error

Similar to the autocorrelated source error model, we can sum over G versions of the cost functions of the form of Equation 4.36 to arrive at the following minimization cost function

for jointly reconstructing multiple signals,

$$\begin{aligned}\min_{\mathbf{a}_1,\ldots,\mathbf{a}_G,\boldsymbol{\Delta}} &\sum_{g=1}^{G}\sum_{v=1}^{V}\sum_{j=1}^{n_v}\left[\frac{w_j^v}{\sigma_{y_g}^2}\left|p_g(q_j^v)-y_{g_j}^v\right|^2\right] \\ &+G\sum_{v=1}^{V}\sum_{j=1}^{n_v}\left[(A_v-B_v)w_j^v(\Delta_j^v)^2\right]+G\sum_{v=1}^{V}\left[B_v\left(\sum_{j=1}^{n_v}\sqrt{w_j^v}\Delta_j^v\right)^2\right].\end{aligned} \tag{4.67}$$

If the optimization was done independently for each of the G signals it would require solving for $GN+\sum_{g=1}^{G}(2M_gL+1)$ variables in total (N location error variables and $(2M_gL+1)$ frequency coefficient variables for each of the G signals). However, since the N location error variables are identical for each of the signals, with the joint optimization we reduce the variable space to $N+\sum_{g=1}^{G}(2M_gL+1)$ dimensions. The set of variables to solve for in the LMA algorithm is the same as in Equation 4.66.

The remainder of the derivation of the cost functions and Jacobian matrices to minimize the cost function in Equation 4.67 to determine the signal reconstructions is given in Appendix A.2.

4.1.3.3 Fixed Source Error

Joint optimizations are similarly extended to the Fixed Source Error model to improve the signal reconstructions. The cost function of Equation 4.44 for the individual signals is extended to G signals as

$$\min_{\mathbf{a}_1,\ldots,\mathbf{a}_G,\boldsymbol{\Delta}} \sum_{g=1}^{G}\sum_{v=1}^{V}\sum_{j=1}^{n_v}\left[\frac{\sigma_v^2}{\sigma_{y_g}^2}w_j^v\left|p_g(q_j^v)-y_{g_j}^v\right|^2\right]+G\sum_{v=1}^{V}\left[\left(\sum_{j=1}^{n_v}w_j^v\right)(\Delta^V)^2\right]. \tag{4.68}$$

This optimization setup reduces the number of variables to be solved for from $GV+\sum_{g=1}^{G}(2M_gL+1)$ for G separate signals to $V+\sum_{g=1}^{G}(2M_gL+1)$ variables for the joint optimization. The vector of variables to solve for is

$$\boldsymbol{\beta}=\begin{bmatrix}\Delta^1 & \ldots & \Delta^V & \mathbf{a}_1^T & \ldots & \mathbf{a}_G^T\end{bmatrix}^T. \tag{4.69}$$

The remainder of the derivation of the cost functions and Jacobian matrices to minimize the cost function in Equation 4.68 to determine the signal reconstructions is given in Appendix A.3.

4.1.3.4 Asynchronous Data Collection

The previous sections assumed that the data used for each of the different signals was sampled at the same locations, i.e., the sampling set (x_j^v locations) are identical for each of the signals. However, due to the asynchronous and varying sampling rate nature of embedded vehicle sensors, this assumption is not necessarily valid. However, as long as the samples for the different signals are obtained from an intersecting set of vehicles, the signal reconstructions could still benefit from the joint framework since the errors between samples from the same vehicle would still be correlated. This section details the joint signal reconstruction framework for the fixed source error assumption under asynchronous data collection environments.

Let there be n_{vg} data samples collected from the v^{th} vehicle for the g^{th} signal. Let N_g denote the total number of samples aggregated to reconstruct signal g, where

$$N_g = \sum_{v=1}^{V} n_{vg}. \tag{4.70}$$

Let $x_{g_j}^v$ denote the j^{th} sample from the v^{th} vehicle for the g^{th} signal, and let $w_{g_j}^v$ denote that sample's respective weight for the signal reconstruction. The cost function for the synchronous joint reconstructions in Equation 4.68 is therefore modified to

$$\min_{\mathbf{a}_1,\ldots,\mathbf{a}_G,\boldsymbol{\Delta}} \sum_{g=1}^{G}\sum_{v=1}^{V}\sum_{j=1}^{n_{vg}} \frac{\sigma_v^2}{\sigma_{y_g}^2} w_{g_j}^v \left| p_g(q_{g_j}^v) - y_{g_j}^v \right|^2 + \sum_{v=1}^{V} \left[\left(\sum_{g=1}^{G}\sum_{j=1}^{n_v} w_{g_j}^v \right) (\Delta^V)^2 \right] \tag{4.71}$$

for the asynchronous case.

The joint minimization follows the LMA framework and is very similar to what was outlined in Section 4.1.3.3. The main differences are the sizes of the vectors and matrices, and replacing the multiplicative factor G in the final two terms of the cost function with

summations over all the signals. The vector of variables $\boldsymbol{\beta}$ is the same as in the fixed source error model in Section 4.1.3.3 since the location error for all samples from a vehicle are assumed to be equal, so changing which samples from a vehicle are collected for each signal does not alter this assumption.

The remainder of the derivation of the cost functions and Jacobian matrices to minimize the cost function in Equation 4.71 to determine the signal reconstructions is given in Appendix A.4.

4.1.3.5 Joint Reconstruction Analysis

The joint reconstructions are compared against individual reconstructions for four example signals. The signals are sums of sinusoids of varying amplitudes, frequencies, and phase shifts, to create the following four signals with varying degrees of smoothness:

$$\begin{aligned} y_1 = &\sin(2\pi(0.132695)x + 2.1) + 0.3\sin(2\pi(0.21)x + 0.5) + \\ &1.1\sin(2\pi(0.325869)x + 0.3) + 0.4\sin(2\pi(1.612)x + 0.1), \end{aligned} \tag{4.72}$$

$$\begin{aligned} y_2 = &1.1\sin(2\pi(0.232695)x - 1.187) + 0.4\sin(2\pi(0.31)x - 0.5) + \\ &1.3\sin(2\pi(0.825869)x + 0.83) + 0.35\sin(2\pi(1.312)x + 0.7547), \end{aligned} \tag{4.73}$$

$$\begin{aligned} y_3 = &1.4\sin(2\pi(0.17856)x - 4.874) + 0.9\sin(2\pi(0.61)x + 0.9547) - \\ &0.8\sin(2\pi(1.03658)x + 2.5478) + 0.25\sin(2\pi(1.723)x - 1.8547), \end{aligned} \tag{4.74}$$

$$\begin{aligned} y_4 = &\sin(2\pi(0.152695)x + 1.1) + 0.25\sin(2\pi(0.19)x + 0.5) + \\ &1.1\sin(2\pi(0.295869)x - 0.31) + 0.02(x - 7.1)^2. \end{aligned} \tag{4.75}$$

The signals were considered over the domain $x \in [0, 20]$ and a Tukey window was applied to the first and last 10% of each of the four signals.

Reconstructions were performed for both the individual and joint versions of reconstruction Algorithms 2 and 3, along with the reconstruction from Algorithm 4 that does

not account for the errors in variables. The reconstructions are performed for two example noise conditions, Joint Example A and B, as given by the sampling parameters in Table 4.6.

The resulting reconstructed signals for Joint Example A are shown in Figure 4.6 with the associated RMSE values given in Table 4.7. For all of the signals the joint reconstruction for Algorithm 3 improved on the individual Algorithm 3 reconstructions and produced the lowest error. The improvements on Algorithm 2 when going to the joint reconstruction were not as substantial, however this is generally due to the sampling noise conditions, where for this example the correlated vehicle error component dominated the independent term. The location errors are larger than the periods of some of these signals, which is why the individual reconstructions sometimes have difficulty reconstructing the signal. However the joint reconstructions are able to use information from other signals to help with the reconstructions for all signals. Note that the reconstruction algorithm still works even with different bandlimit parameters for each of the signals.

An example of the joint reconstruction performing considerably better for Algorithm 2 occurs for the sampling conditions in Example B in Table 4.6 where the independent noise term becomes more dominant. Reconstructions of the example signals for these parameters is shown in Figure 4.7 with RMSE results presented in Table 4.8. Both the independent and joint versions of Algorithms 2 and 3 outperform Algorithm 4 due to the errors in variables corrections. The joint version of Algorithm 2 performs best as it best accounts for the error model in the optimization.

The joint reconstructions are not guaranteed to improve the results over the independent reconstructions for all scenarios. If the sampling noise, particularly with respect to the location error, is significantly larger than even the dynamics of the lowest frequency signal, then both the independent and joint reconstructions can overfit to the data for each of the signals. Additionally, if due to the sampling conditions, one of the considered signals has a significantly better reconstruction from the independent scenarios than the

Table 4.6: Reconstruction and sampling parameters for the joint reconstruction example comparisons

Parameter	**Value**	
	Joint Example A	Joint Example B
M_1, M_2, M_3, M_4	$\frac{36}{L}, \frac{30}{L}, \frac{38}{L}, \frac{9}{L}$	
σ_v	0.7	0.2
σ_x	0.2	0.4
$\sigma_{y_1}, \sigma_{y_2}, \sigma_{y_3}, \sigma_{y_4}$	0.005, 0.01, 0.015, 0.02	0.05, 0.1, 0.15, 0.2
V	395	
Vehicle sample spacing	Rayleigh dist., mean 2	

Table 4.7: Root mean-square-error (RMSE) results for joint reconstructions of four example signals for Joint Example A

	RMSE				
	Alg 2	Alg 2-J	Alg 3	Alg 3-J	Alg 4
Signal 1	1.00	0.60	0.33	0.24	0.62
Signal 2	0.95	0.89	0.66	0.41	0.99
Signal 3	0.98	0.86	0.70	0.43	0.93
Signal 4	0.23	0.41	0.08	0.12	0.48

other signals in the joint reconstruction, then it is possible that the reconstruction for that signal becomes slightly worse for the joint reconstruction as the sampling errors from the other reconstructions dictate the direction of improvement of the joint cost function. Although the final joint reconstructions are likely to be improved in the aggregate across all the signals, it is possible that a signal with an already good fit does not exhibit any improvement.

4.1.4 Parameter Optimization

Section 4.1.4.1 describes an algorithm to select an appropriate bandlimit given the sampling conditions and Section 4.1.4.2 details how the signal reconstruction variance can be predicted as a function of the number of aggregated samples. This algorithm is used to estimate the number of samples required to meet a specified variance.

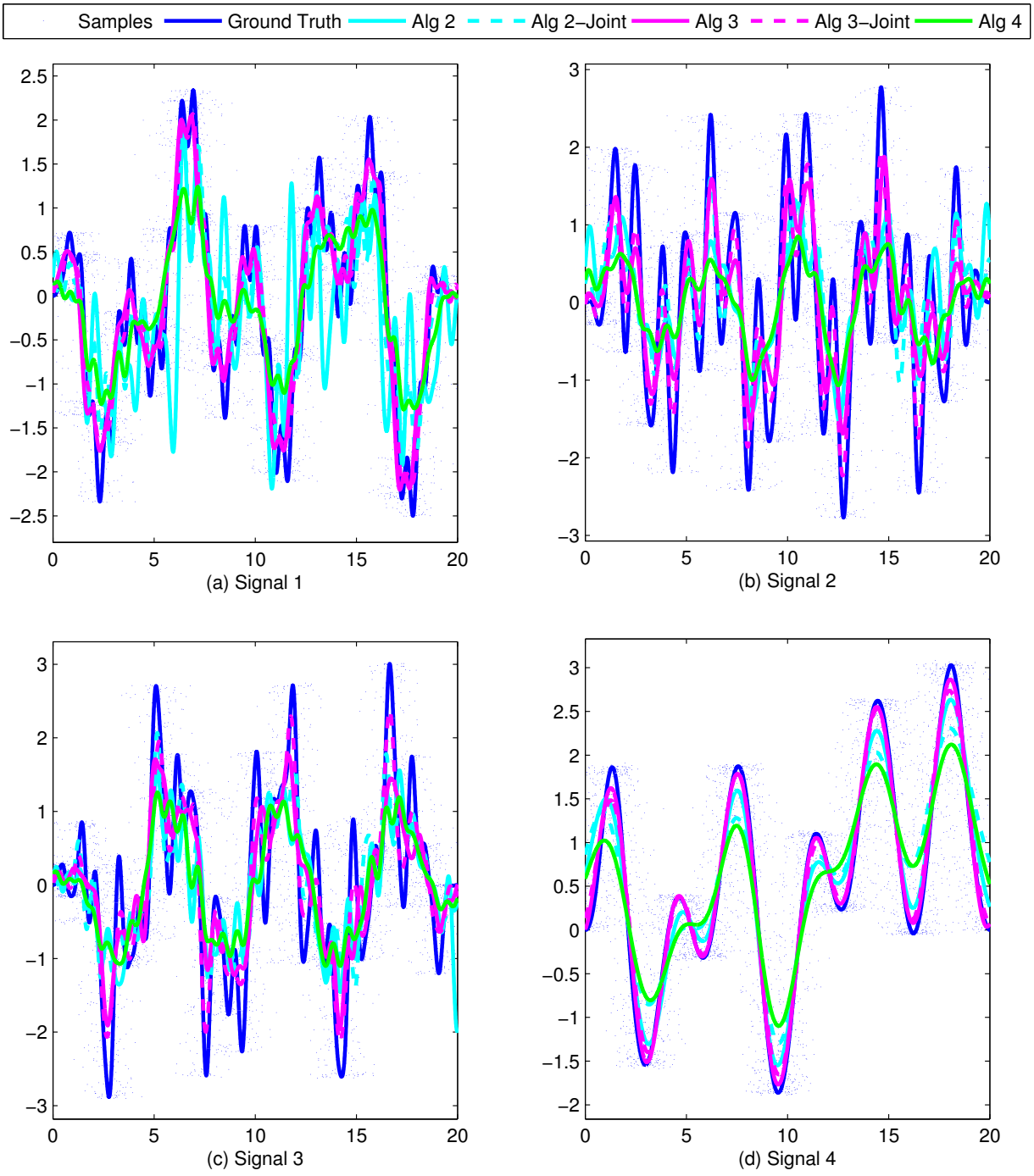


Figure 4.6: Comparison of joint and individual reconstructions for four example signals with sampling and reconstruction conditions from Joint Example A in Table 4.6

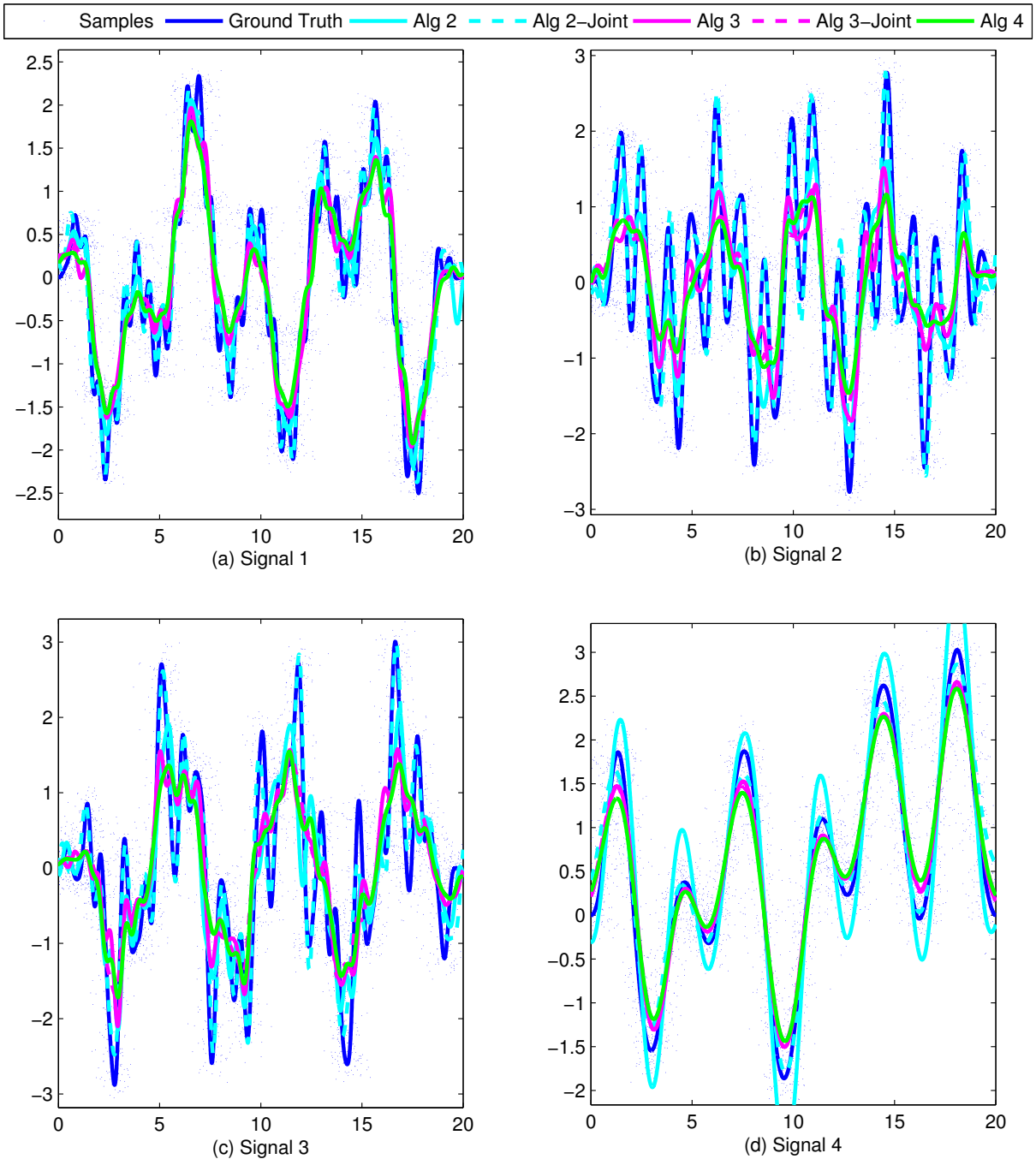


Figure 4.7: Comparison of joint and individual reconstructions for four example signals with sampling and reconstruction conditions from Joint Example B in Table 4.6

Table 4.8: Root mean-square-error (RMSE) results for joint reconstructions of four example signals for Joint Example B

	RMSE				
	Alg 2	Alg 2-J	Alg 3	Alg 3-J	Alg 4
Signal 1	0.34	0.20	0.38	0.38	0.41
Signal 2	0.83	0.33	0.78	0.85	0.87
Signal 3	0.80	0.34	0.75	0.76	0.81
Signal 4	0.40	0.14	0.22	0.24	0.26

4.1.4.1 Bandlimit Selection

For the presented signal reconstruction algorithms, the choice of bandlimit, M, and number of sampling sources, V, or total aggregated samples, N, affect the accuracy and viability of the reconstruction. The bandlimit is inversely proportional to the finest signal components that can be reconstructed. If the original signal has spatial frequency components higher than the chosen bandlimit, there will be components of the signal that cannot be reconstructed regardless of the amount of sample noise or number of samples. This results in a baseline bound for how accurately the signal can be reconstructed for a given bandlimit. However using an extremely high bandlimit to try and reconstruct the signal perfectly has its own limitations. One limitation is that a bandlimit higher than the maximum frequency of the signal could overfit the reconstruction to noise in the data. Secondly, for higher bandlimits, higher sampling density requirements necessitate that more samples are required before the reconstruction becomes viable.

An example of such behavior is shown in Figure 4.8a, which displays the RMSE of the reconstructions of an incline angle signal from the simulated data as a function of the number of data samples. The reconstructions are shown for bandlimits $M = \left\{\frac{10}{L}, \frac{40}{L}\right\}$ and sampling value noise $\sigma_y^2 = \{0, 0.2, 0.8\}$. As shown in the figure, reconstructions have a baseline error determined by the bandlimit. The baseline RMSE is lower for the higher bandlimit since it captures more frequencies of the original signal. However, before the

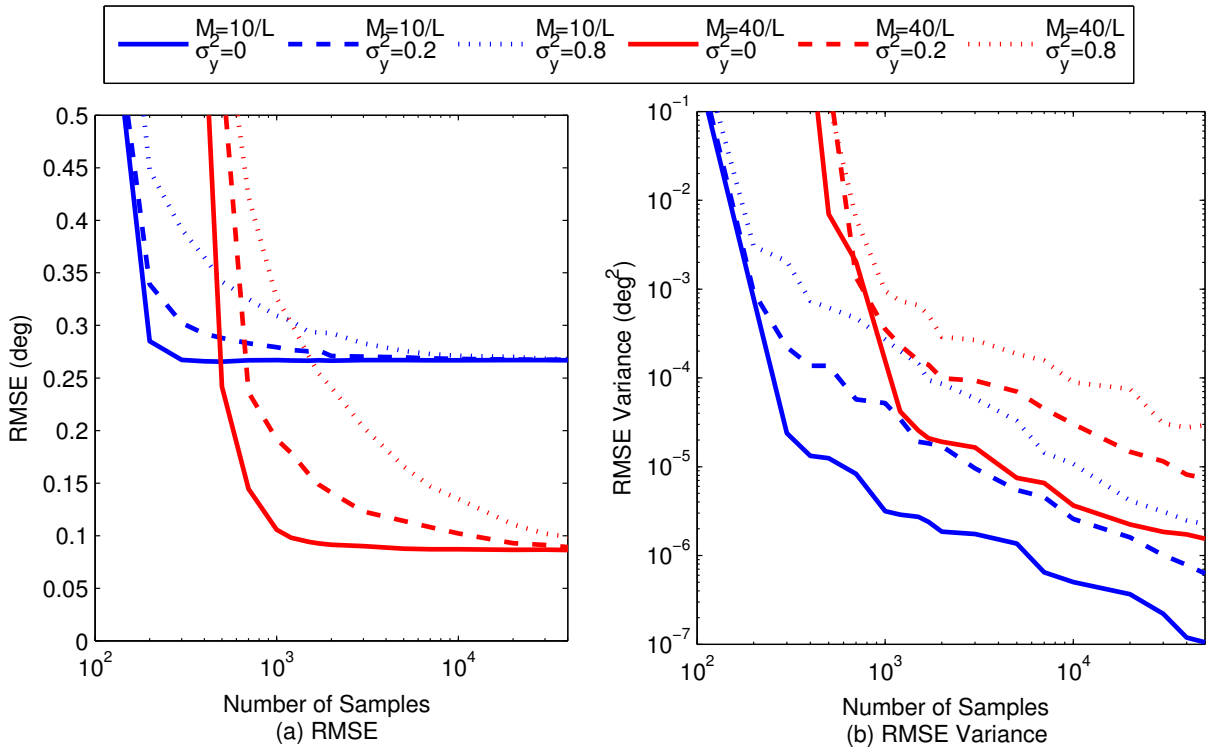


Figure 4.8: (a) RMSE, and (b) Variance of the RMSE, of signal reconstruction as a function of the number of samples

sampling density is sufficient for the higher bandlimit reconstruction to be viable and begin to approach the baseline, the higher bandlimit reconstruction is worse than for the lower bandlimit reconstruction.

Reconstructions using noisy data approach the baseline as the number of samples increases. However, there is a risk of overfitting noisy data with a higher bandlimit. The higher frequency components in the reconstruction allow the signal to fluctuate between noisy values, and more samples are required to accurately reconstruct the signal. This is why the RMSE is greater for the higher bandlimit reconstruction for low N.

Due to these indicated behaviors, it is important to determine an appropriate bandlimit M for the signal reconstruction. Using prior domain knowledge to determine the bandlimit is ideal, however such information is not always available and an appropriate bandlimit instead must be estimated. For a general nonuniform sampling problem with only sample

value noise, the work in [64] increases the bandlimit until a stopping criterion based on a function of the error is met. We adapt this heuristic to the MSVR sampled data with the sample location error models described in Chapter 3.

Let $C(M)$ represent the result of the cost function used for deriving the respective signal reconstruction (Equations 4.16, 4.36, 4.44, and 4.53) as a function of the bandlimit, excluding the Voronoi weights for a segregated set of validation data. The bandlimit is increased until the value of $C(M)$ increases greater than a threshold number of times in a given window of tested bandlimits. This heuristic tests for when increasing the bandlimit no longer provides significant benefits and potentially begins to overfit the data. Given a set of noisy data, as M continues to increase, the reconstruction error will eventually begin to increase due to the overfitting. This method does not necessarily find the reconstruction with the minimum reconstruction error, however it finds a bandlimit where further increases are not necessarily substantially beneficial. This acts as a regularization mechanism for the bandlimit given the noise in the sampled data.

Consider three example reconstructions using the fixed source error model of Algorithm 3 of the bank angle signal in Figure 4.3 from 2500 samples using the following three noise conditions:

- $\sigma_v = 5$ m, $\sigma_x = 0.5$ m, $\sigma_y = 0.01^\circ$
- $\sigma_v = 50$ m, $\sigma_x = 10$ m, $\sigma_y = 0.3^\circ$
- $\sigma_v = 40$ m, $\sigma_x = 20$ m, $\sigma_y = 0.8^\circ$

The validation data cost function and resulting reconstructed signal RMSE for the three different conditions as a function of the bandlimit are shown in Figure 4.9. As shown in this figure, both the cost function and the RMSE decrease initially as the increasing bandlimit better reconstructs the original signal, then begins to increase as the higher bandlimit overfits the sampled data. The bandlimit where the error or validation cost function begins to increase depends on the noise parameters. For the given reconstructions, the selected

Table 4.9: Selected bandlimits and resulting RMSE values for reconstructions as compared to the optimal

Sampling Noise Conditions	Selected Bandlimit $\left(\frac{1}{L}\right)$	Recon. RMSE (°)	Min. RMSE Bandlimit $\left(\frac{1}{L}\right)$	Min. RMSE (°)
$\sigma_v = 5$ m, $\sigma_x = 0.5$ m, $\sigma_y = 0.01°$	72	0.021	82	0.017
$\sigma_v = 50$ m, $\sigma_x = 10$ m, $\sigma_y = 0.3°$	31	0.217	45	0.202
$\sigma_v = 40$ m, $\sigma_x = 20$ m, $\sigma_y = 0.8°$	37	0.469	40	0.428

bandlimits and resulting reconstruction RMSE compared to the minimum from testing all bandlimits are given in Table 4.9. For each of the reconstructions, the selected bandlimit was slightly less than the optimal, due to the regularization, however the reconstruction RMSE was very close to the optimal value. This performs relatively well given that the optimal bandlimit, RMSE values, and ground-truth signal are unknown in practice.

For the first listed sampling conditions, the reconstructions are shown in Figure 4.10 for the ideal bandlimit compared to reconstructions from scenarios with either too high or too low bandlimits. As expected, for the too low bandlimit there are components of the signal at a higher frequency than the bandlimit that cannot be reconstructed, resulting in a poor reconstruction. Also the too high bandlimit overfits to the sampling noise with too high magnitudes for the high frequency signal components. How much these reconstructions deviate from the ideal depends on the ground-truth signal and sampling conditions, however the presented heuristic for estimating an appropriate bandlimit is generally applicable.

4.1.4.2 Number of Samples

Given a selection of N random samples, the variance of the reconstruction values, and therefore also of the RMSE, decreases as N increases, as shown in Figure 4.8b. Each incrementally added sample stabilizes the reconstruction, however there are diminishing benefits for each additional sample.

Quantifying the variance of the reconstructed signal is critical to examining the validity

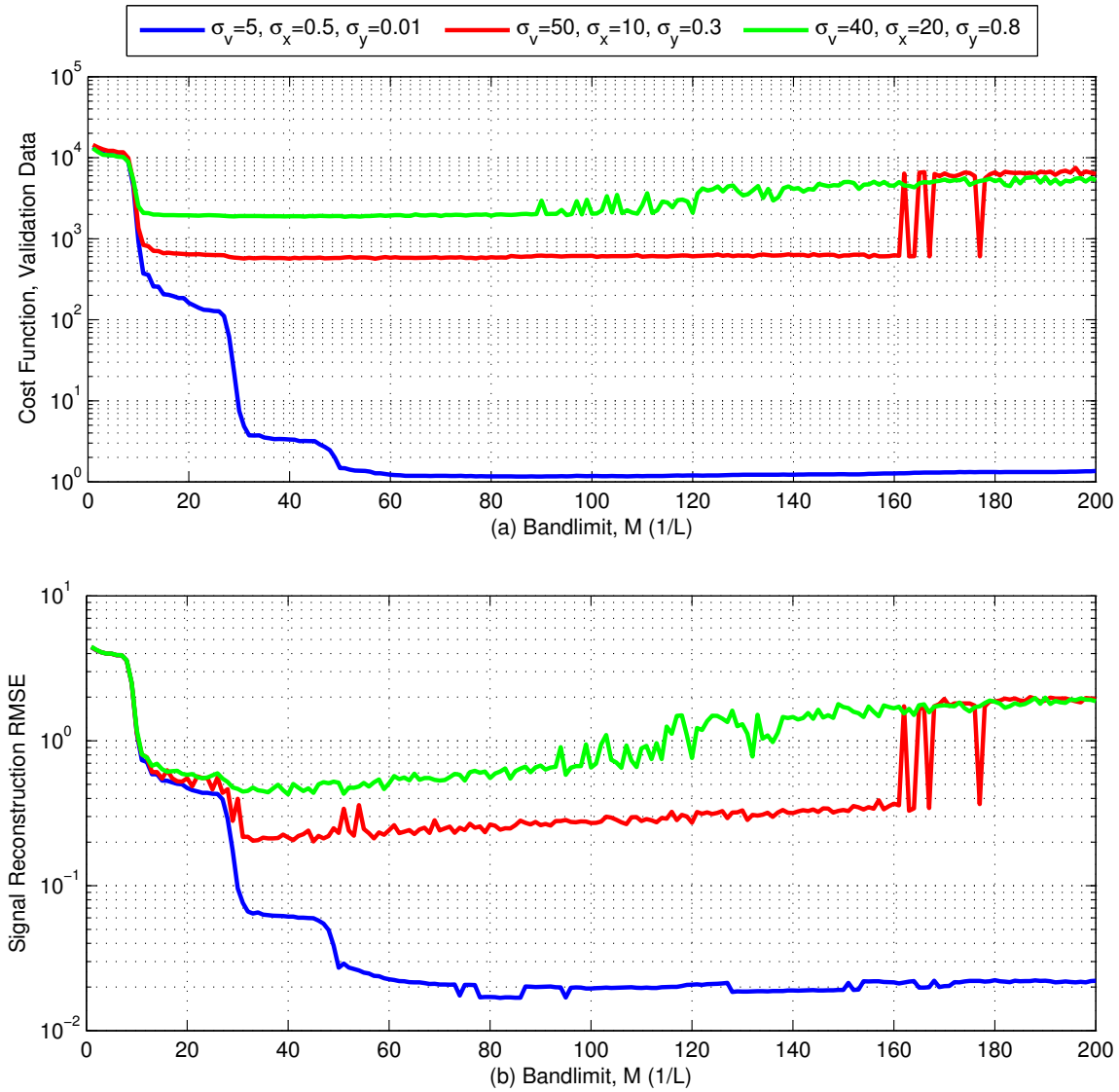


Figure 4.9: (a) Cost function of validation data $C(M)$, and (b) Resulting signal reconstruction RMSE for reconstructing the bank angle signal with different noise conditions for varied bandlimits

and stability of the reconstruction. By estimating the variance of the reconstruction as a function of the variance of the input sample error, we can quantify the stability of the reconstruction and predict the number of samples required to reach a desired variance.

When analyzing how the error propagates from the input samples to the reconstruction, it is important to consider both the real and imaginary components of the reconstruction

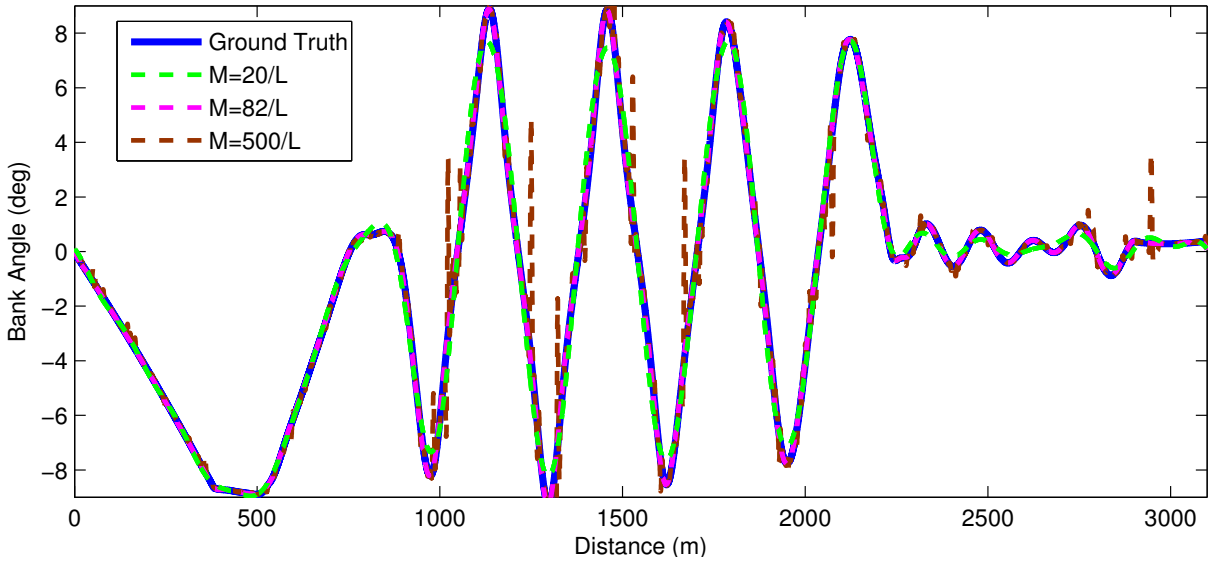


Figure 4.10: Reconstruction comparison of ideal bandlimit to those from too high and low bandlimits for sampling conditions, $\sigma_v = 5$ m, $\sigma_x = 0.5$ m, $\sigma_y = 0.01°$, for 2500 bank angle samples

algorithm. The noise in the input data affects not just the magnitude, but also the phase of the frequency coefficients, $\mathbf{a}$. The following details a method to analyze error propagation through a system of complex variables.

Denote an example complex number, $\mathbf{x}_1$, as the vector $\mathbf{x}_1 = \begin{bmatrix} x_{11} & x_{21} \end{bmatrix}^T$, where $\mathbf{x}_1 = x_{11} + ix_{21}$ [65]. This formulation allows us to represent N complex variables using $2N$ real variables. Therefore, given a function $f : \mathbb{C}^m \to \mathbb{C}^n$, the function in this notation is represented as,

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_n \end{bmatrix} = \begin{bmatrix} y_{11} \\ y_{21} \\ y_{12} \\ y_{22} \\ \vdots \\ y_{1n} \\ y_{2n} \end{bmatrix} = f(\mathbf{X}) = f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m) = f(x_{11}, x_{21}, x_{12}, x_{22}, \ldots, x_{1m}, x_{2m}). \quad (4.76)$$

The function is composed of real and imaginary components such that,

$$\mathbf{Y} = \begin{bmatrix} f_{11}(\mathbf{X}) \\ f_{12}(\mathbf{X}) \\ \vdots \\ f_{1n}(\mathbf{X}) \end{bmatrix} + i \begin{bmatrix} f_{21}(\mathbf{X}) \\ f_{22}(\mathbf{X}) \\ \vdots \\ f_{2n}(\mathbf{X}) \end{bmatrix}. \tag{4.77}$$

We construct the complex $(2n \times 2m)$ Jacobian matrix, $\mathbf{J}(\mathbf{Y})$ of the output as follows,

$$\mathbf{J}(\mathbf{Y}) = \begin{bmatrix} \frac{\partial f_{11}}{\partial x_{11}} & \frac{\partial f_{11}}{\partial x_{21}} & \frac{\partial f_{11}}{\partial x_{12}} & \frac{\partial f_{11}}{\partial x_{22}} & \cdots & \frac{\partial f_{11}}{\partial x_{1m}} & \frac{\partial f_{11}}{\partial x_{2m}} \\ \frac{\partial f_{21}}{\partial x_{12}} & \frac{\partial f_{21}}{\partial x_{21}} & \frac{\partial f_{21}}{\partial x_{12}} & \frac{\partial f_{21}}{\partial x_{22}} & \cdots & \frac{\partial f_{21}}{\partial x_{1m}} & \frac{\partial f_{21}}{\partial x_{2m}} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial f_{1n}}{\partial x_{11}} & \frac{\partial f_{1n}}{\partial x_{21}} & \frac{\partial f_{1n}}{\partial x_{12}} & \frac{\partial f_{1n}}{\partial x_{22}} & \cdots & \frac{\partial f_{1n}}{\partial x_{1m}} & \frac{\partial f_{1n}}{\partial x_{2m}} \\ \frac{\partial f_{2n}}{\partial x_{12}} & \frac{\partial f_{2n}}{\partial x_{21}} & \frac{\partial f_{2n}}{\partial x_{12}} & \frac{\partial f_{2n}}{\partial x_{22}} & \cdots & \frac{\partial f_{2n}}{\partial x_{1m}} & \frac{\partial f_{2n}}{\partial x_{2m}} \end{bmatrix}. \tag{4.78}$$

The second-order statistics of $\mathbf{X}$ are represented by the $2m \times 2m$ covariance matrix, $\mathbf{V}(\mathbf{X})$,

$$\mathbf{V}(\mathbf{X}) = \begin{bmatrix} v^2(x_{11}) & v(x_{11}, x_{21}) & \ldots & v(x_{11}, x_{1m}) & v(x_{11}, x_{2m}) \\ v(x_{21}, x_{11}) & v^2(x_{21}) & \ldots & v(x_{21}, x_{1m}) & v(x_{21}, x_{2m}) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ v(x_{1m}, x_{11}) & v(x_{1m}, x_{21}) & \ldots & v^2(x_{1m}) & v(x_{1m}, x_{2m}) \\ v(x_{2m}, x_{11}) & v(x_{2m}, x_{21}) & \ldots & v(x_{2m}, x_{1m}) & v^2(x_{2m}) \end{bmatrix}, \tag{4.79}$$

where $v(x_{ij}, x_{kl})$ is the covariance between x_{ij} and x_{kl}.

By assuming a first-order Taylor series expansion, we can propagate the error to determine the covariance matrix for $\mathbf{Y}$, $\mathbf{V}(\mathbf{Y})$, by,

$$\mathbf{V}(\mathbf{Y}) = \mathbf{J}(\mathbf{Y})\mathbf{V}(\mathbf{X})\mathbf{J}^T(\mathbf{Y}). \tag{4.80}$$

We can now apply this formulation to determine the variance of the reconstructed signal $p(x)$. This variance is calculated in terms of the variance of the average square of the signal

value, $\frac{1}{L}\int_0^L |p(x)|^2\,dx$. This is equivalent to the average energy of the signal, E_p, which by Parseval's identity,

$$E_p = \frac{1}{L}\int_0^L |p(x)|^2\,dx = \mathbf{a}^*\mathbf{a}. \tag{4.81}$$

Therefore we can determine the signal variance by calculating the variance of the reconstruction parameters $\mathbf{a}$ and therefore also the average signal energy. We first require the covariance matrix $\mathbf{\Sigma}$ for the input sensor measurement errors. The covariance matrix $\mathbf{\Sigma}$ is determined as,

$$\mathbf{\Sigma} = \mathbf{M}\mathbf{C}_{yy}\mathbf{M}^T, \tag{4.82}$$

where $\mathbf{M}$ is a $2N \times N$ matrix to rearrange the covariance matrix $\mathbf{C}_{yy}$ into that required for the complex form, where

$$M_{lk} = \begin{cases} 1, & \text{if } l = 2k-1 \\ \\ 0, & \text{otherwise} \end{cases}. \tag{4.83}$$

We assume the input locations and sample values are purely real and therefore only have error in their real components. The covariance for the determined frequency coefficients and the average energy of the signal are given respectively by,

$$\mathbf{V}(\mathbf{a}) = \mathbf{J}_a\mathbf{\Sigma}\mathbf{J}_a^T, \tag{4.84}$$

$$\mathbf{V}(E_p) = \mathbf{J}_E\mathbf{\Sigma}\mathbf{J}_E^T. \tag{4.85}$$

The derivations of the Jacobian matrices $\mathbf{J}_a$ and $\mathbf{J}_E$ for the nonuniform sampling algorithm are provided in Appendix B.

For reconstructions by Algorithm 4, the covariance matrix $\mathbf{C}_{yy}$ should be slightly modified to adapt to the sample location error. The sample value variance σ_y^2 should be reapproximated as the variance of the sample error to the reconstructed signal, assuming that all error manifests itself as sample value error. We demonstrate the validity of this

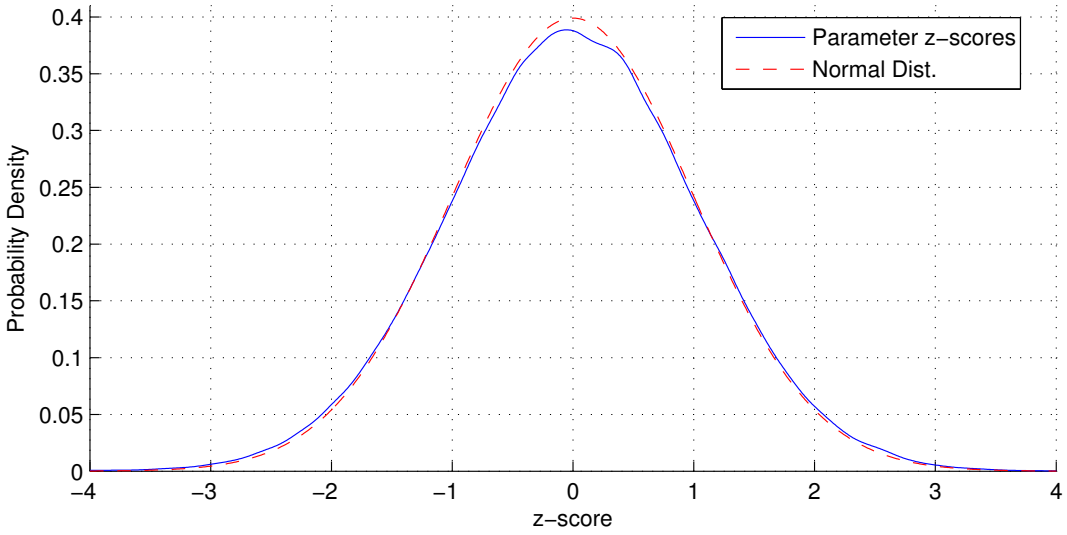


Figure 4.11: Comparison of the distribution of z-scores of the estimated parameter variances to the normal distribution

approach by collecting the calculated variances of the reconstruction parameter $\mathbf{a}$ from reconstructing the following signal,

$$\begin{aligned} y = &\sin(2\pi(0.132695)x + 2.1) + 0.3\sin(2\pi(0.21)x + 0.5) \\ &+ 1.1\sin(2\pi(0.325869)x + 0.3) + 0.4\sin(2\pi(2.012)x + 0.1), \end{aligned} \tag{4.86}$$

under 13 different error conditions from combinations of sampling variances in $\sigma_x^2 \in \{0, 0.01, 0.04, 0.1\}$ and $\sigma_y^2 \in \{0, 0.01, 0.04, 0.09\}$ from 1500 samples, each repeated for 100 iterations. If the variances are correct, we would expect the z-scores of the values of $\mathbf{a}$ given the calculated variances to be normally distributed. The probability density function (PDF) of the z-scores from these trials as compared to the normal distribution is shown in Figure 4.11. The distribution well approximates a normal PDF, with a slight underestimate of the variances. This is a result of estimating the sample variances after the reconstruction, where there may be some overfitting, instead of using the original derived values.

The variance of the average signal energy is expected to decay at a rate $\frac{1}{N}$. To demonstrate this, we iterate through 100 reconstructions each of the signal in Equation 4.86

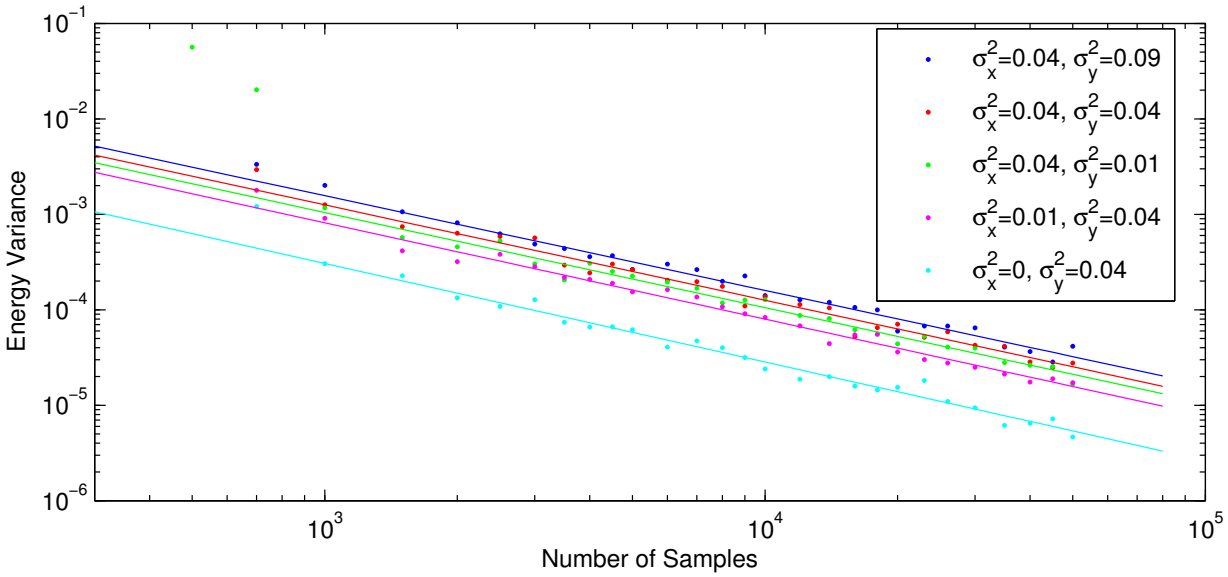


Figure 4.12: Signal reconstruction energy variance as a function of the number of samples

for noise parameters $(\sigma_x, \sigma_y) \in \{(0, 0.2), (0.1, 0.2), (0.2, 0.1), (0.2, 0.2), (0.2, 0.3)\}$. The real variances of the energy are displayed in Figure 4.12 for the different reconstructions, including a best fit line on the log-log scale for each example of the noise parameters once the reconstruction becomes viable. The line on the log-log scale corresponds to the equation $\sigma_E^2 = \frac{e^b}{N^{-m}}$ where m and b are the slope and intercept respectively of the best fit line on the log-log scale. For decay rate $\frac{1}{N}$ we would expect $m = -1$. The values of m for the given examples are $\{-1.033, -1.01, -0.9974, -0.9983, -0.9922\}$, which is in good agreement with the $\frac{1}{N}$ energy variance decay rate.

We can therefore use the energy variance calculations to predict the required number of samples to bound the average energy variance below a specified value. For a calculated energy variance, σ_{E0}^2, for a given number of samples, N_0, if the variance is desired to be below some threshold ε, then the number of required samples, N_ε, to reach that threshold is given by,

$$\hat{N}_\varepsilon > \frac{\sigma_{E0}^2 N_0}{\varepsilon}. \tag{4.87}$$

It would also be beneficial to estimate the intercept value $\hat{b}$ from multiple variance calcula-

tions using subsets of the data instead of relying on a single estimate. In those situations,

$$\hat{N}_\varepsilon > \frac{e^{\hat{b}}}{\varepsilon}. \tag{4.88}$$

4.2 Binary Event Detection

Some examples of road information are best represented as isolated binary events, i.e., localized events described by a binary variable indicating whether or not they exist at a given location. Example events include potholes, speedbumps, slippery road regions, or other events indicated in Table 2.1. Drivers would benefit from the knowledge of these events for planning their routes, and automated vehicle control systems could be adjusted to prepare for such events instead of reacting after encountering them.

The MSVR sampling conditions, as given in Chapter 3, affect binary event detection as they do continuous signal reconstructions. In the following we present a framework for detecting binary events given these conditions, with mechanisms to filter data, reduce network bandwidth, and adapt to multi-lane environments.

4.2.1 Detection Framework

There are two concurrent goals in the binary event detection framework. The first goal is to accurately and reliably detect the appropriate event either within the vehicles or after data has been aggregated from the vehicles to the Cloud. The second goal involves localizing the exact position of the event on the road, both in terms of which lane the event resides in, and its exact longitudinal position on the road.

For binary events we construct a detection framework, outlined in the block diagram in Figure 4.13. In general, we first select data from both real-world and simulated driving events. The simulated data is first used to train the model. The training data is grouped, either within the individual vehicles or after being transmitted to the Cloud for aggregation.

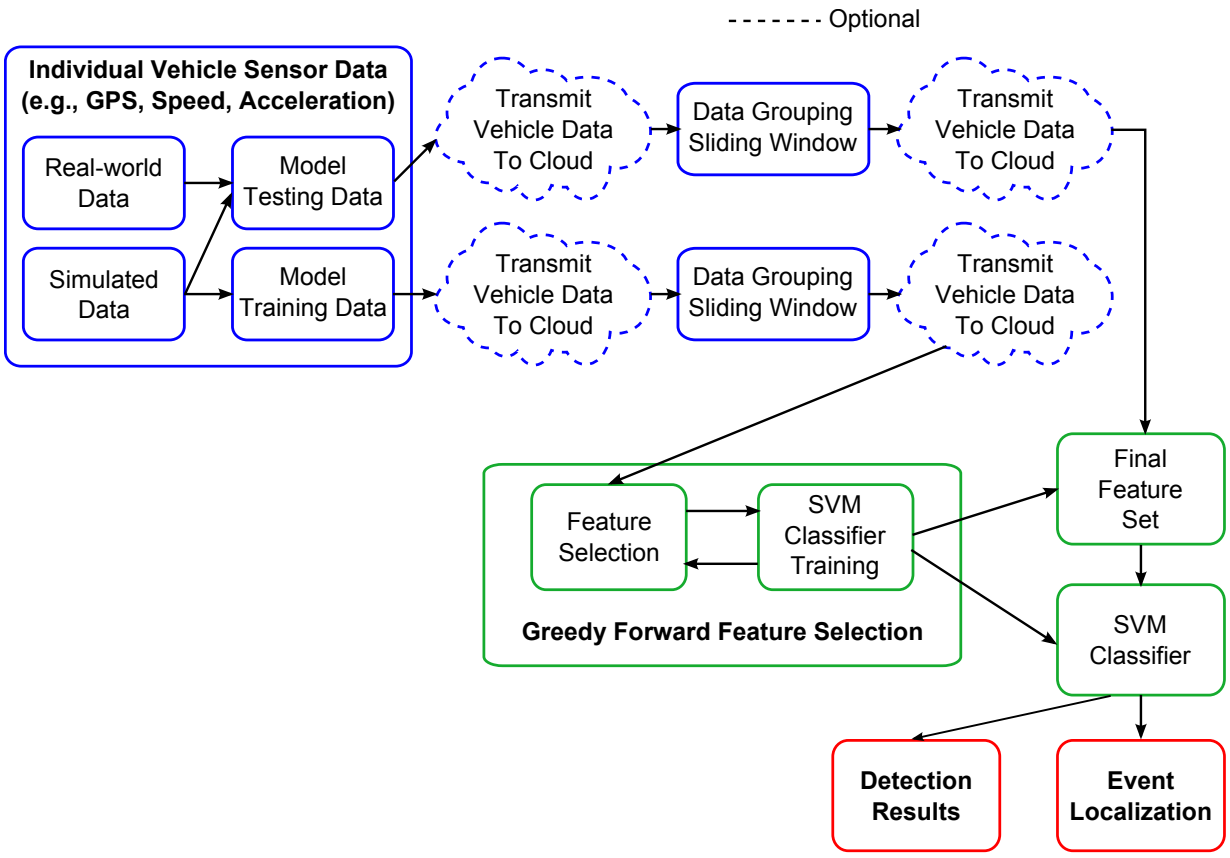


Figure 4.13: Detection framework for the training and testing of binary road features.

From the aggregated data, features are selected and a model is trained to detect the specific event. Simulated and real-world data is then put through the same process to test the detection model and produce final results.

The following sections delve into further detail on the detection framework and the processes outlined in Figure 4.13. Section 4.2.2 describes how the data is grouped for the calculation of candidate features. Section 4.2.3 explains how those features may be adapted for multi-lane road environments. Section 4.2.4 details how features can be selected from the candidate list for training the classifier. Section 4.2.5 explains how data may be filtered to reduce the required transmission bandwidth, and Section 4.2.6 discusses how the detected binary events may be finely localized to their exact location. Refer to Table 4.10 for a guide to new notation introduced in this section.

Table 4.10: Binary event detection framework notation

s	Sliding window length
$w_i^{\mathbf{x}}$	Multi-lane weighting for sample $\mathbf{x}$ in lane i
w_i^v	Multi-lane weighting for samples from vehicle v in lane i
β	True detection rate
α	False alarm rate

4.2.2 Candidate Features

It is important to identify mathematical features that are able to suitably discriminate between event and non-event regions in the data set. Whereas some previous works have manually generated feature lists from observations of sensor data, we believe that manual selection does not account well for all potential environmental variations and could lead to misidentifying events in certain scenarios. We instead rely on machine learning approaches to select appropriate features.

For aggregating data, we group the data in a window of length s. The length s should be determined to encompass the length of the event, the distance for any measurable residual effects in the vehicle, and a tolerance for reasonable GPS error. The window should slide an amount less than s to count data in multiple windows. Since the true event location is originally unknown, this redundancy ensures that there exists some window that overlaps with the event region.

Features are calculated over the data in each window to form a list of candidate features for use in classifying whether or not each window contains the specified road event. These features are functions of any sensor data that has been identified as having discriminative capabilities for that event.

4.2.3 Multi-lane Adaptations

Many road events are localized to individual lanes. Therefore features should be calculated differently when applying the sliding window to each of the lanes. Each window should encompass all the data points measured across the width of the road, since due to GPS error, it is not known a priori from which lanes the data originated. However to differentiate the lanes, a weighting scheme is applied to the data for calculating the candidate features. The weighting of each aggregated data point when calculating the features is the posterior probability of the measurement belonging to the specified lane, as determined by Bayes' rule. Given n lanes, the weighting for lane i of a data point with measured GPS coordinates $\mathbf{x}$ is,

$$w_i^{\mathbf{x}} = P(\text{lane} = i | \mathbf{X} = \mathbf{x}), \tag{4.89}$$

$$= \frac{P(\mathbf{X} = \mathbf{x}|\text{lane} = i)P(\text{lane} = i)}{\sum_j^n P(\mathbf{X} = \mathbf{x}|\text{lane} = j)P(\text{lane} = j)}. \tag{4.90}$$

As an initial assumption, we may assume that the location error is i.i.d. normally distributed around the true lane centerlines. Therefore the prior probability of a GPS measurement being made at $\mathbf{x}$ while driving in lane j is

$$P(\mathbf{X} = \mathbf{x}|\text{lane} = j) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\|\mathbf{x}-\hat{\mathbf{q}}_j\|^2}{2\sigma^2}}, \tag{4.91}$$

where $\hat{\mathbf{q}}_j$ is the estimated true location of the GPS location, and is taken as the point on the centerline of lane j closest to $\mathbf{x}$.

However this method assumes the sample location error is i.i.d., which we demonstrated in Chapter 3 was not the case in the vehicle sampling environment. The weighting algorithm should more appropriately account for the correlated MSVR sampling conditions.

We assume that a vehicle drives in only a single lane over the measured domain. Therefore we can calculate the joint probability of all the samples from a single vehicle belonging

to a lane. The weights (or probabilities) should be equal for all the samples from a specific vehicle since it is assumed that the vehicle does not change lanes. This assumption is not always valid, however by restricting the domain to a limited distance it could apply to most vehicles. Also, side information from a lane change detection algorithm could be used to indicate cutoff points for where a new vehicle lane weight should be calculated.

Let w_i^v represent the sample weights for the data points from vehicle v in lane i. For the data samples $\mathbf{x}_j^v$ from vehicle v,

$$w_i^v = \frac{P\left((\mathbf{X}_1^v, \ldots, \mathbf{X}_{n_v}^v) = (\mathbf{x}_1^v, \ldots, \mathbf{x}_{n_v}^v) | \text{lane} = i\right) P(\text{lane} = i)}{\sum_j^n P\left((\mathbf{X}_1^v, \ldots, \mathbf{X}_{n_v}^v) = (\mathbf{x}_1^v, \ldots, \mathbf{x}_{n_v}^v) | \text{lane} = j\right) P(\text{lane} = j)}. \tag{4.92}$$

The conditional prior probability is determined using the covariance matrix from the autocorrelated error model in Equation 3.15, as

$$P\left((\mathbf{X}_1^v, \ldots, \mathbf{X}_{n_v}^v) = (\mathbf{x}_1^v, \ldots, \mathbf{x}_{n_v}^v) | \text{lane} = j\right) = \frac{1}{\sqrt{(2\pi)^{n_v} |\mathbf{C}_{xx}^v|}} \exp\left(-\frac{1}{2} \begin{bmatrix} d_{1j}^v \\ \vdots \\ d_{n_v j}^v \end{bmatrix}^T (\mathbf{C}_{xx}^v)^{-1} \begin{bmatrix} d_{1j}^v \\ \vdots \\ d_{n_v j}^v \end{bmatrix} \right), \tag{4.93}$$

where $d_{i\,j}^v$ is the distance from x_i^v to the closest centerline point in lane j, $\hat{q}_{i\,j}^v$, such that

$$d_{i\,j}^v = \|x_i^v - \hat{q}_{i\,j}^v\| r_{i\,j}^v, \tag{4.94}$$

where $r_{i\,j}^v$ is a binary variable, $r_{i\,j}^v \in \{-1, 1\}$, indicating if x_i^v is to the right or left of the lane's centerline.

4.2.4 Classifier Training

It is important to determine an appropriate combination of the candidate features to use for detection, since not every feature has an equal ability to discriminate the specified event in all situations. For example, a pothole induces a large vertical acceleration as a vehicle drives in and out of the pothole. A large magnitude for the vertical acceleration

could therefore be an appropriate feature. However, that feature would not necessarily differentiate a pothole from a speedbump, which could also induce a vehicle to have a similar large vertical bounce. A lateral acceleration component could therefore help to differentiate between these two events since a speedbump is generally uniform across the road, while potholes are often smaller and isolated to only one side of the vehicle. Therefore reducing the candidate feature list to a final selected feature list is important as it not only reduces the dimension of the problem space by eliminating candidate features that may not assist in detection, but also determines an appropriate combination of features to well detect the event.

One example binary event classifier we use is the Support Vector Machine (SVM) [66]. SVMs are discriminative classifiers that identify the boundary between the features of two classes of data by maximizing the margin between the two classes. These are appropriate for binary event classification as they can discriminate between the two classes (namely, event regions and non-event regions). Radial basis kernel functions [66] are used to create nonlinear boundaries between the classes, which will be shown in Section 5.2.4.2 to be important as many vehicle sensor signals have nonlinear relationships between event and non-event classes.

One feature selection method we use is a greedy forward feature selection algorithm [67] to determine which of the candidate features are best suited for classification. Pseudocode for this feature selection algorithm is shown in Figure 4.14. Using this process, a separate SVM is first created for every individual candidate feature. A score function is computed for each SVM based on the true detection rate, β, and the false alarm rate, α, on the validation data. If P is the number of windows that actually contain the event, N is the number of windows that do not actually contain the event, TP is the number of windows in which the event was correctly detected, and FP is the number of windows in which the event was mistakenly detected, then $\beta = \frac{TP}{P}$ and $\alpha = \frac{FP}{N}$. For each round, whichever

```
1: i ← 0
2: score(0) ← −∞
3: selectedFeatures ← {}
4: candidateFeatures ← {allFeatures}
5: do
6:     i ← i + 1
7:     bestScore ← −∞
8:     bestScoreIndex ← 0
9:     for j : candidateFeatures do
10:        [β, α] ←trainAndTestSVM(selectedFeatures, candidateFeatures(j))
11:        γ(j) ← β − α
12:        if γ(j) > bestScore then
13:            bestScore ← γ(j)
14:            bestScoreIndex ← j
15:        end if
16:    end for
17:    selectedFeatures.add(candidateFeatures(bestScoreIndex))
18:    candidateFeatures.remove(bestScoreIndex)
19:    score(i) ← γ(bestScoreIndex)
20: while (score(i) − score(i − 1) > scoreThreshold)
21: selectedFeatures.remove(i)
22: finalSVM ←trainSVM(selectedFeatures)
```

Figure 4.14: Pseudocode for greedy forward feature selection algorithm for training the binary road event SVM detector

candidate feature gives the highest score value, γ, where

$$\gamma = \beta - \alpha, \tag{4.95}$$

is added as a selected feature. This process is then repeated by training new SVMs with the selected feature list in addition to each of the remaining candidate features individually. We stop adding candidate features to the selected feature list when the difference in score values on the validation data in consecutive rounds falls below a score threshold.

4.2.5 Data Filtering

Filtering data is an important mechanism in the vehicle-Cloud information extraction architecture. Filtering data within the vehicle is a mechanism to prevent transmitting excessive data to the Cloud, thus conserving the limited bandwidth available in the ITS network. When data filtering is applied to a specific detection algorithm, it can prevent

the inclusion of specifically identified data in the algorithm, with the intent of improving the detection rate.

The filtering algorithm is dependent on the type of data used in the detection process. Generally the input data is compared against a threshold based on the remainder of the data set, or on some expected output from processing the data. If $h(\cdot)$ is a function representing a feature calculation for a detection algorithm, and $g(\cdot)$ is a function representing the expected output for the feature calculated in $h(\cdot)$, then for a given data sample x_j from a potentially larger data set $\mathbf{x}$, the following distance value d_j could be evaluated,

$$d_j = d\left(h(x_j) - g(\mathbf{x})\right), \tag{4.96}$$

where $d(\cdot)$ is a distance metric function determined for that specific application. Comparing d_j to a threshold ε could be used to determine whether or not the value x_j should be transmitted or included in the next step of the detection algorithm. Specific versions of this function are shown for a vehicle data filter in Section 5.2.4.1 and a Cloud data filter in Section 5.2.4.3.

4.2.6 Localization

Correctly detecting a binary event in a window is not necessarily sufficient information for describing the event. Due to GPS error or outliers, an event may be detected in multiple windows around its true location, and the window size may be significantly larger than the actual event size. It is important to attempt to finely locate the event within these windows.

For the SVM classifiers, event localization is performed using the decision values from the SVM training. The decision values for each true event from training are averaged to create a base pattern. After detecting a pothole in testing, the resulting test SVM decision values are cross-correlated against the base pattern. The maximum of the cross-correlation defines a location shift to where the event is estimated to begin. The difference between

the shifted location and the true pothole location is defined as the localization error.

4.3 Temporal Weightings

Many road features have a temporal element associated with them. For example, icy roads return to normal as the weather changes or as they are salted, and municipalities eventually repair potholes. Depending on the nature of the event or its associated location, these events will all have different time constants. A slippery road may last for only hours, while a pothole may persist for months before it is repaired.

Therefore, the data aggregated from multiple vehicles should not be equally weighted when trying to detect road features. Older data should be weighted less heavily in the detection process as it is less likely to represent the current state of the environment. A time weighting function is therefore included with the weights associated with each data sample, similar to the time-decay sequential hypothesis testing method used in [68]. We use a time exponentially weighted model to decrease the weight for any sample, using a time constant τ_T, which varies depending on the nature of the specific event and should be approximated as a factor of $\frac{1}{\ln 2}$ multiplied by the expected half-life of the feature. For an initial weighting for a sample, w_0, for example derived from the multi-lane adaptation as given by Equation 4.93 or from the Voronoi weighting scheme for the signal reconstructions presented in Section 4.1.2.5, then the weighting for the respective sample at time t is given by

$$w(t) = w_0 \exp\left(-\frac{t}{\tau_T}\right). \tag{4.97}$$

Since the exponential function only approaches, but never equals, zero, any aggregated data sample in the system would have some non-zero weighting for all times. This can lead to significant storage and computational complexities as the run time for many algorithms is dependent on the number of data samples used, irrespective of their weightings. We therefore employ a lower threshold, λ_T, on the weightings to eliminate data samples from

the system when the weighting falls below the threshold. This eliminates the need to further consider the data samples with negligible weights in the system and reduces the runtime of any algorithm. The weighting scheme in Equation 4.97 is therefore modified to

$$w(t) = \begin{cases} w_0 \exp\left(-\frac{t}{\tau_T}\right), & \text{if } \exp\left(-\frac{t}{\tau_T}\right) \geq \lambda_T \\ 0, & \text{if } \exp\left(-\frac{t}{\tau_T}\right) < \lambda_T \end{cases}. \tag{4.98}$$

In the absence of temporal knowledge, the weightings from these schemes could be ignored or interpreted as multiplying the original weights by a factor of one. An application of the weights to the binary detection framework are discussed in Section 5.2.5.5. The following section demonstrates adaptations so that the temporal weights can be applied to continuous signal reconstructions.

4.3.1 Continuous Signal Reconstruction

The temporal weighting algorithm detailed in Section 4.3 is applicable to the signal reconstruction problems, with the initial weights, w_0, given by the Voronoi weights in Section 4.1.2.5. However tuning the time constant τ_T for the signal reconstruction problems is more critical than for the binary event detection algorithm. This is because the signal reconstruction algorithms are extremely sensitive to the weighting scheme and any deviations can degrade the reconstruction.

Consider a situation where the road environment is static, however the decay constant was poorly approximated to describe the system as fast changing. For the binary events, the data being input into the features would still have similar statistical properties to the existing data. The weighting scheme therefore would not change the result of the features, it would only result in there being less total weighted data than necessary. However in the signal reconstruction case, the Voronoi weights would be significantly changed, even though they should remain close to their original values. This could result in a poor reconstruction.

4.3.1.1 Parameter Optimization

We therefore explore an adaptive method to try and optimize the time constant parameter for signal reconstruction. Due to the MSVR sampling conditions, the amount of new data received and the data's respective locations cannot be easily predicted. Since this is not a traditional input-output type system, we require a different metric by which to model potential ground-truth signal changes. If the ground-truth signal were static, we would expect the statistical properties of the new input data at a given time to resemble the properties of the existing data. Any change in these properties is an indication that the ground-truth signal may have changed. We frame the optimization in regard to the signal reconstruction error.

Let $\epsilon^2(t)$ represent the weighted sample reconstruction error for all samples up to time t such that,

$$\epsilon^2(t) = \sum_{v=1}^{V(t)} \sum_{j=1}^{n_v(t)} \left| p(x_j^v) - y_j^v \right|^2 w(t - t_j^v), \tag{4.99}$$

where $w(\cdot)$ is the weight function in Equation 4.98, and $V(t)$ and $n_v(t)$ are the number of vehicles and samples per vehicle to have been aggregated at time t. Similarly, let $N(t)$ represent the weighted total number of aggregated samples at time t, given by the sum of the weights of the samples and therefore not necessarily an integer, where

$$N(t) = \sum_{v=1}^{V(t)} \sum_{j=1}^{n_v(t)} w(t - t_j^v). \tag{4.100}$$

Let δ represent a given time interval after which the time constant τ_T should be recalculated. We construct the following sample error equivalence equation, using a weight α to denote how much to weight new data relative to the existing data,

$$\epsilon^2(t+\delta) = \epsilon^2(t) \left(1 + \alpha \frac{N(t+\delta) - N(t)}{N(t)} \right). \tag{4.101}$$

Equation 4.101 is a projection of the future error given the current error and number of newly input samples. By rearranging Equation 4.101,

$$\alpha = \frac{N(t) \left(\epsilon^2(t+\delta) - \epsilon^2(t) \right)}{\left(N(t+\delta) - N(t) \right) \epsilon^2(t)}. \tag{4.102}$$

To avoid biasing above or below the existing error values, let β represent a new weight based on α such that new data is always weighted more heavily than older data. Let

$$\beta = 1 + |\alpha - 1|. \tag{4.103}$$

This β value is exponentially smoothed based on the number of new and old sample points and mapped to a β value for time $t + \delta$ denoted $\beta_{t+\delta}$, such that

$$\beta_{t+\delta} = \beta_t \left(\frac{N(t)}{N(t+\delta)} \right) + \beta \left(\frac{N(t+\delta) - N(t)}{N(t+\delta)} \right). \tag{4.104}$$

This value is converted into the exponential time decay constant for time t by

$$\tau_{Tt+\delta} = \frac{\delta}{\ln \beta_{t+\delta}}. \tag{4.105}$$

4.3.1.2 Results

To demonstrate the effects of the temporal weighting scheme on the signal reconstruction behavior, consider a situation involving the following two signals,

$$\begin{aligned} y_1(x) = &\sin(2\pi(0.132695)x + 2.1) + 0.3\sin(2\pi(0.21)x + 0.5) + \\ &1.1\sin(2\pi(0.325869)x + 0.3) + 0.4\sin(2\pi(1.612)x + 0.1), \end{aligned} \tag{4.106}$$

$$\begin{aligned} y_2(x) = &1.9\sin(2\pi(0.232)x + 1.2584) + 0.8\sin(2\pi(0.465)x - 0.8) + \\ &0.3\sin(2\pi(1.312)x + 0.258). \end{aligned} \tag{4.107}$$

The ground truth-signal is given by a linear transformation from $y_1(x)$ to $y_2(x)$ such that,

$$y(x) = \begin{cases} y_1(x), & \text{if } t \leq t_1 \\ \frac{t-t_1}{t_2-t_1} y_1(x) + \frac{t_2-t}{t_2-t_1} y_2(x), & \text{if } t_1 < t \leq t_2 \\ y_2(x), & \text{if } t > t_2 \end{cases}. \tag{4.108}$$

This function is a static function on $y_1(x)$ for $t \leq t_1$, then linearly changes to $y_2(x)$ over the interval $(t_1, t_2]$, then remains constant on $y_2(x)$ after t_2.

Consider three types of reconstructions, with 10 new sources sampling over the signal per time unit, each with sample spacing drawn from a Rayleigh distribution with mean one. In the first reconstruction, the parameter τ_T is set adaptively according to the algorithm given in Section 4.3.1.1. In the final two reconstructions, the time constant is kept static for the entire time period with one reconstruction performed with a relatively low time constant and one reconstruction performed with a relatively high time constant. The resulting values for the time constants for the given time period are shown in Figure 4.15. The values for t_1 and t_2 are indicated by the black dashed lines in the figure. RMSE values for the three different reconstructions are shown in Figure 4.16, with example reconstructions at various times compared to the ground-truth signal shown for the adaptive time constant algorithm in Figure 4.17 and for the relatively low static time constant in Figure 4.18.

The reconstruction using the low time constant reconstructs the signal poorly during the initial and final periods when the ground-truth signal is static since the Voronoi weights are being drastically changed from their original values. Conversely, the reconstruction using the high constant time constant performs poorly in the middle time interval when the signal is changing since it is unable to adapt quickly enough to the new data. The adaptive time constant performs relatively well in all regions.

4.4 Chapter Summary

In this thesis chapter, we introduced algorithms to detect both continuous and binary types of road environmental information, given the noisy, asynchronous, and undersampled MSVR sampling conditions of vehicle sensor data. Adaptations were provided so that the detection algorithms could operate in multi-lane environments and with temporally changing information. Methods of data filtering were introduced to try and reduce the required network bandwidth when aggregating data to the Cloud for detection. The algorithms were analyzed against each other and previous work to assess the viability of the

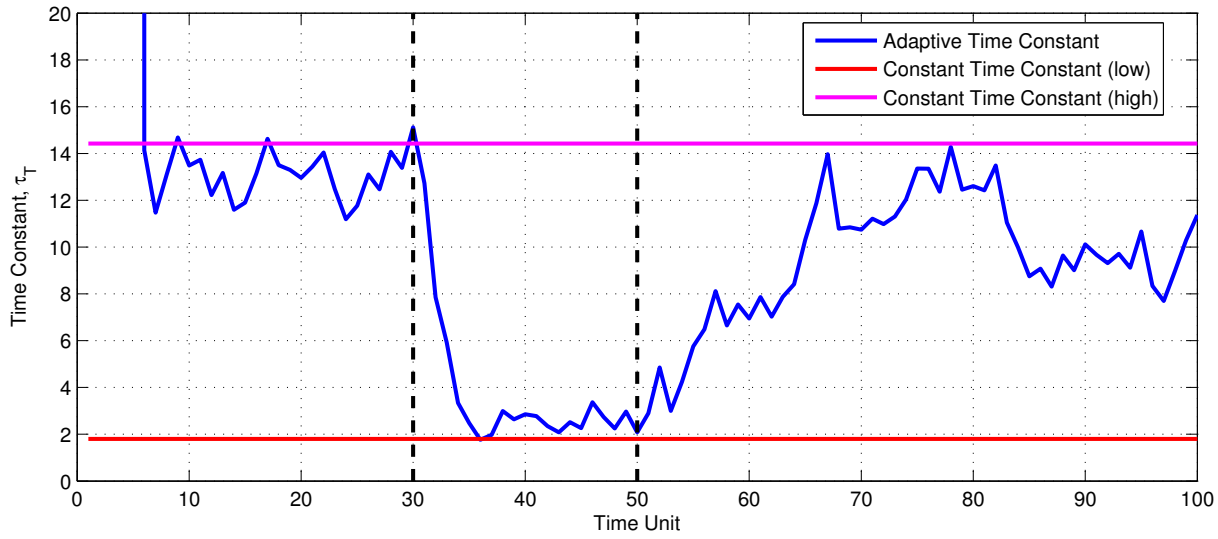


Figure 4.15: Evolution of time constant values for the adaptive algorithm compared against constant cases

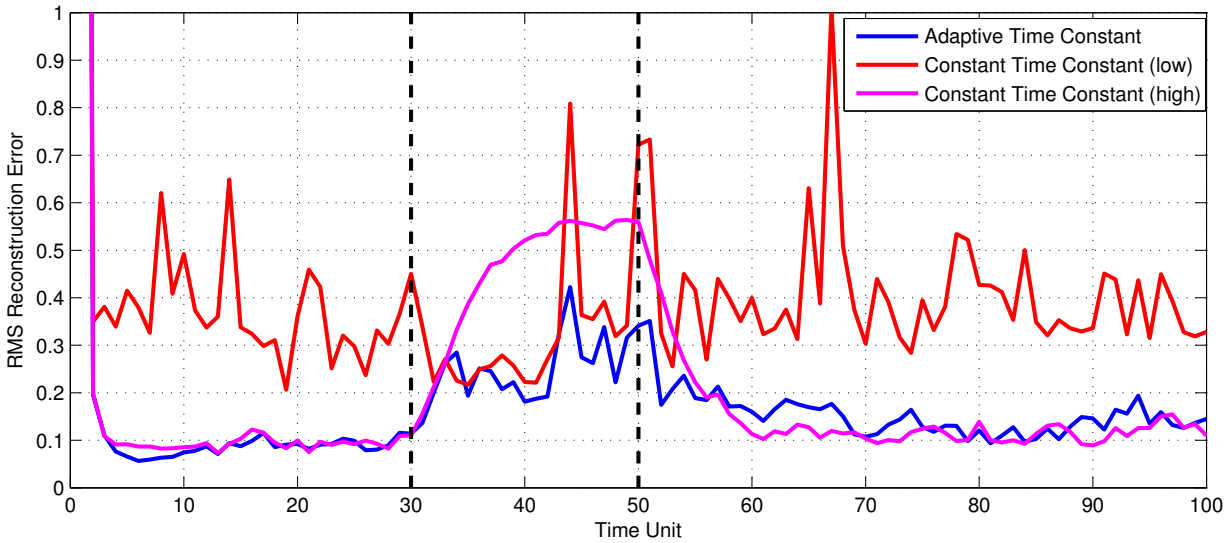


Figure 4.16: RMSE values for the adaptive and static time constant reconstructions

event detection processes under a variety of circumstances.

In the following chapter, these event detection algorithms are used to detect specific types of environmental road information.

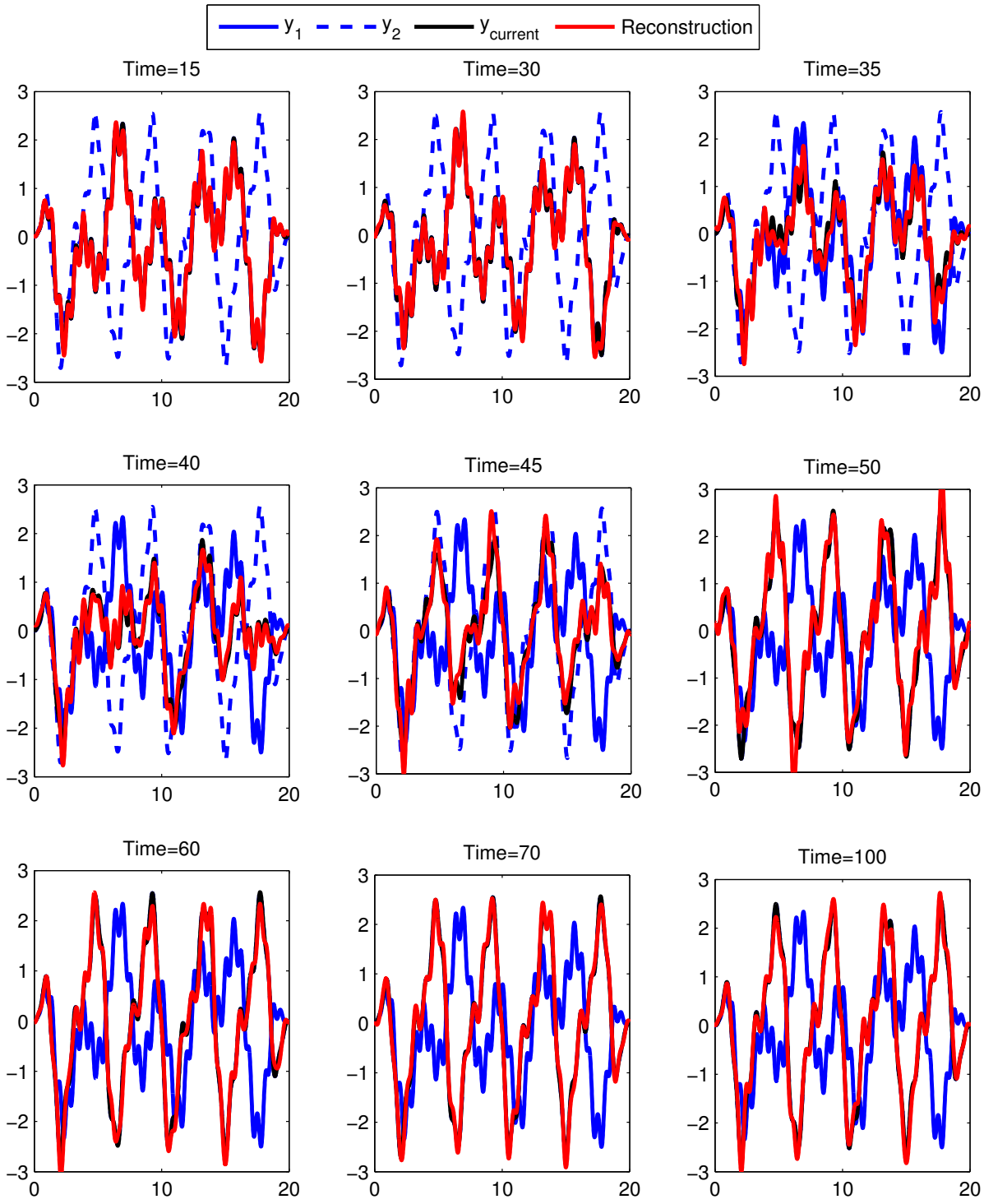


Figure 4.17: Reconstructions at various time intervals using the adaptive time constant

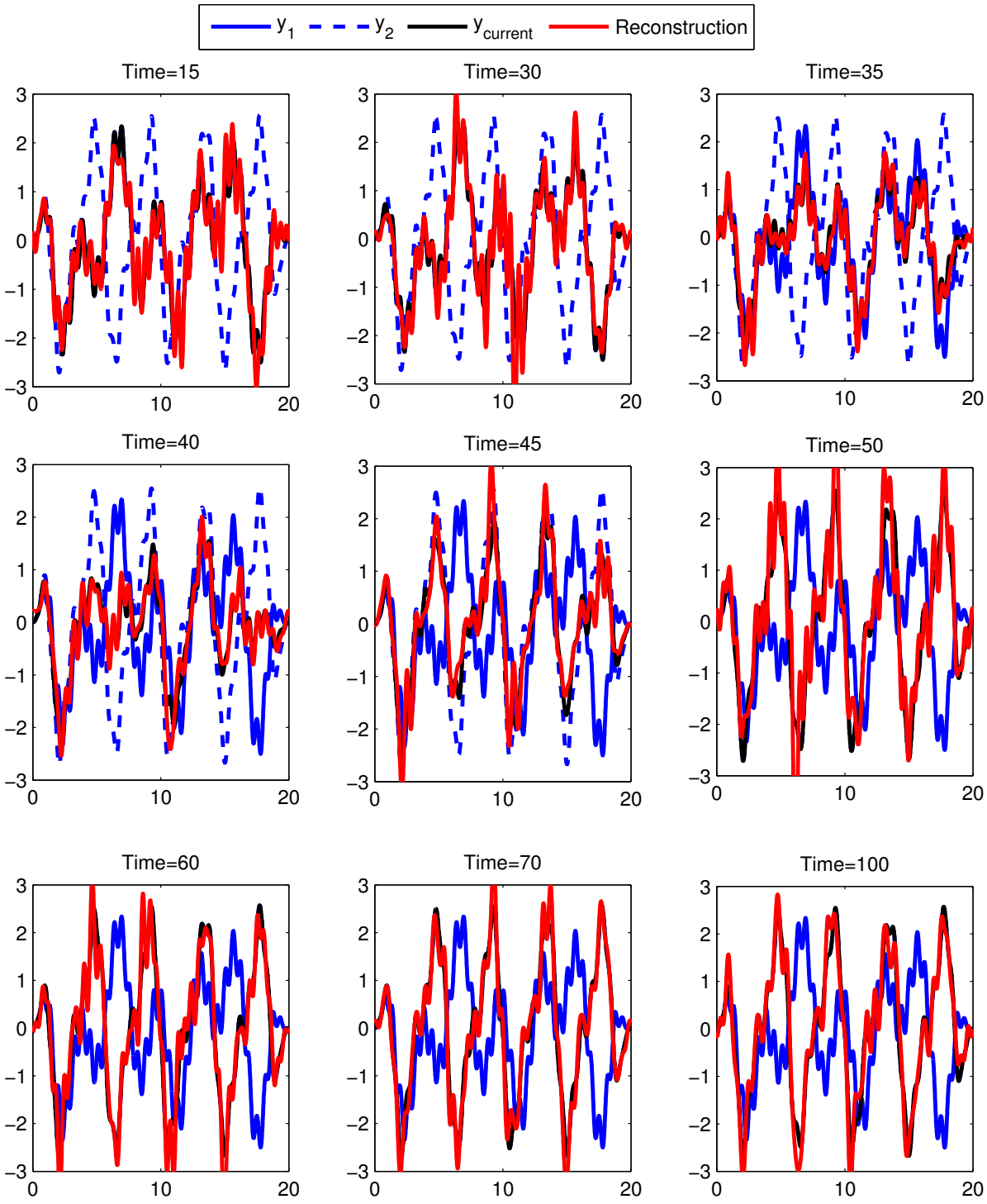


Figure 4.18: Reconstructions at various time intervals using the relatively low time constant

Chapter 5

Environmental Road Information

In this chapter, we apply the detection algorithms developed in Chapter 4, within the overall vehicle-Cloud detection framework described in Chapter 2, to extract information on specific types of environmental events. We extract road incline and bank angle information in Section 5.1 and pothole information in Section 5.2 from aggregated MSVR sampled vehicle sensor data.

5.1 Road Incline and Bank Angles

5.1.1 Introduction

Many vehicle control systems require an accurate model of the vehicle dynamics to determine any actuation function. In determining the vehicle dynamics, it is important to try and decouple the components of the vehicle state that should be expected from environmental conditions to those components that are indicative of anomalous or dangerous vehicle conditions. The road incline and bank angles are important factors that influence the vehicle state, and the knowledge of their conditions is useful for many vehicle applications.

Road incline and bank angle information is of particular importance to vehicle stability control systems [69], especially in larger vehicles that may issue roll-over warnings or that attempt to actuate the vehicle to restore stability [70]. However, since many of these stability systems require a large actuation power, it is important to dissociate the roll angle of the vehicle from the bank angle of the road to determine if initiating the control actions is actually mandated.

Speed and steering control algorithms also make use of the vehicle dynamics and therefore also the road angle measurements. Both adaptive cruise control [71] and brake control [72] algorithms are aided by knowledge of road topography. These control algorithms help the vehicle to determine optimal speed levels given that the available engine output power has to be properly directed in regard to the road incline angles and the increasing aerodynamic drag associated with increasing speeds. The speed control algorithms can further help regulate inter-vehicle fleet spacing, for example in highway conditions [73]. Similarly, steering [74] control algorithms are helped by the road incline and bank knowledge to know how to compensate for the environmental conditions.

For each of these control processes, prior knowledge could help to design predictive control algorithms [75] that more efficiently plan for optimal speed or engine output control points so that the vehicle control systems can conserve energy and feasibly operate around their desired control points.

However, it is difficult to decouple the influence of the road incline and bank angles from the remainder of the vehicle dynamics when only examining sensor measurements related to the vehicle's kinematic motion. For example, a lateral acceleration component could be caused by centripetal acceleration from the vehicle driving on a curve, or a gravitational acceleration component from the vehicle driving on a bank. These components need to be separated and indicated as such, particularly when many of these conditions can occur simultaneously (e.g., when driving on a banked curve).

In this thesis work, we demonstrate how to estimate the road bank and incline angles for entire stretches of road using only acceleration, speed, and GPS measurements from embedded vehicle sensors. Since the sampling frequency of the sensors is very low and the measurement data is so noisy, we are unable to determine continuous functions representing the road angle signals using data only from individual vehicles. It is only by aggregating data samples from multiple vehicles and reconstructing the signals using the framework detailed in Section 4.1 that a final signal estimate for entire roads can be determined.

In the following, Section 5.1.2 discusses related work to our road angle determination objective. Section 5.1.3 details our road angle estimation algorithm for individual sample measurements. Section 5.1.4 describes the simulated and real-world data we used to test the algorithm, and Section 5.1.5 presents the results of those experiments.

5.1.2 Related Work

There have been other approaches to determine road bank and incline angle information, however they generally rely on data from more sensors and vehicle-specific parameters. These parameters are not necessarily available in every vehicle, and each additional parameter or sensor introduces a new source of error. High frequency sensors are used to estimate vehicle roll and pitch angles using inertial sensor data in [76]. In [77], known four-wheel speeds, steering angles, acceleration, and gyroscopic data are used to construct a non-linear road-tire friction model to estimate road inclination and bank angles. Vehicle mass, gear and engine torque, wheel radius, vehicle frontal area, and air constant parameters are used with inertial sensor data to design a Kalman filter to estimate the road grade in [75]. However even with this detailed information they still advocate multiple measurement runs to estimate the road grade.

Some works use multiple GPS devices to identify road angles [78]. The carrier phase difference between two roof-mounted GPS antennae as well as the ratio of the calculated

Table 5.1: Road bank and incline angle notation

$\mathbf{a}$	Acceleration vector, vehicle frame
$\mathbf{w}$	Expected acceleration, inertial frame
$\mathbf{f}$	Forward acceleration, vehicle frame, change in speed
a_c	Centripetal acceleration
g	Acceleration due to gravity
v	Vehicle speed
$\mathbf{R}$	Rotation matrix from inertial to vehicle frame
$\mathbf{p}$	Acceleration components related to perturbations
$\mathbf{n}$	Noise in measured acceleration
θ	Road incline angle
ϕ	Road bank angle
D	Normed difference between observed and expected acceleration
p	Minimum expected number of measurements in a window
η	Minimum fraction of data points to expect in a window
V	Total number of vehicles
τ	Threshold on D to transmit data to Cloud

horizontal and vertical velocities are used to determine the road grade and vehicle mass in [79]. Using other sensors, a barometer in addition to acceleration and gyroscope data is used to estimate the road inclination in [80]. Our system does not rely on such extensive measurements; only GPS and accelerometer data are needed.

5.1.3 Road Angle Determination

Refer to Table 5.1 for relevant notation and Figure 5.1 for an illustration of the relevant terms.

The information extraction algorithm presented in this section uniquely determines the road bank, ϕ, and incline, θ, road angles from only acceleration and GPS measurements by determining the rotation matrix, $\mathbf{R}$, that rotates the expected acceleration vector in a local inertial reference frame, $\mathbf{w}$, to the measured acceleration vector from the vehicle

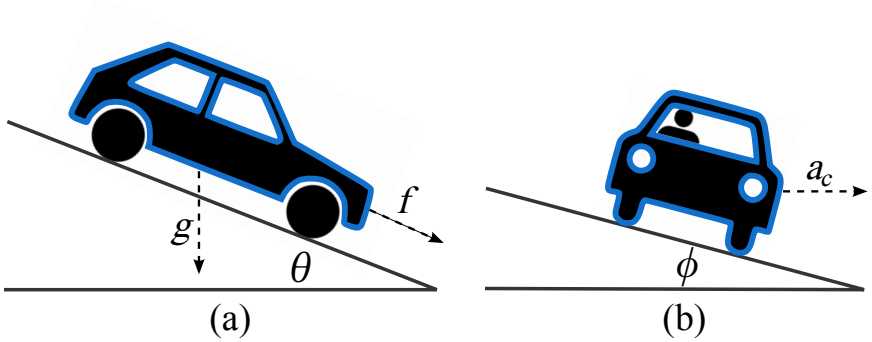


Figure 5.1: Diagram of Road (a) Incline (θ) angle viewed from the right side of the vehicle, and (b) Bank (ϕ) angle viewed from the rear of the vehicle

reference frame, $\mathbf{a}$, where

$$\mathbf{a} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T, \tag{5.1}$$

at that location. The rotation of the acceleration vectors between reference frames is defined by

$$\mathbf{Rw} = \mathbf{a} - \mathbf{f} - \mathbf{p} + \mathbf{n}, \tag{5.2}$$

where $\mathbf{f} = \begin{bmatrix} f & 0 & 0 \end{bmatrix}^T$ denotes the forward acceleration representing the change of speed in the vehicle reference frame, $\mathbf{p}$ denotes the acceleration resulting from perturbations unrelated to the road orientation (e.g., a pothole), and $\mathbf{n}$ denotes the sensor noise.

We determine $\mathbf{R}$ by minimizing the following cost function, representing the difference between the measured vehicle acceleration vector and the expected acceleration inertial frame vector rotated into the vehicle frame,

$$J(\mathbf{R}) = \frac{1}{2} \left\| (\mathbf{a} - \mathbf{f}) - \mathbf{Rw} \right\|^2. \tag{5.3}$$

The cost function in Equation 5.3 is adapted from Wahba's problem [81], which was originally designed to determine satellite attitude. Wahba's problem is commonly solved using the Singular Value Decomposition (SVD) [82]. However, the SVD solution (as well as other solution formulations [83]) requires multiple sets of paired vectors from the vehicle and inertial frames (e.g., accelerometer and magnetometer measurements) in order to determine a unique solution for $\mathbf{R}$. We however, are restricted to only a single accelerometer

measurement per location. We therefore require a method to reduce the infinite set of solutions to a unique one.

Fortunately, unlike satellites, automobiles are constrained to driving on set paths, those being the roads. Therefore, instead of using the same globally defined inertial reference frame for all locations, we can use the GPS coordinates to predetermine a local coordinate system aligned with the road at the measured location. This eliminates the need to determine the vehicle direction or yaw angle from the acceleration vectors and rotation matrix, since the vehicle direction is tangent to the road at that location. We can therefore reduce the inertial reference frame to only two dimensions, with one axis aligned perpendicular to the road on a flat plane, and a second orthogonal axis pointing towards the zenith. An illustration of the difference between these coordinate systems for two example locations is shown in Figure 5.2. In two dimensions, in a locally defined coordinate system, the expected inertial frame acceleration vector is reduced to

$$\mathbf{w} = \begin{bmatrix} a_c \\ \\ g \end{bmatrix}, \tag{5.4}$$

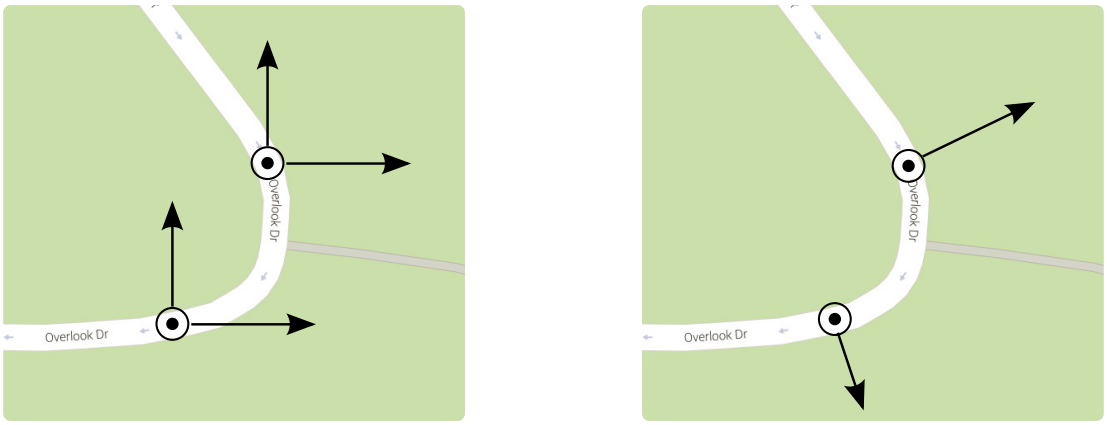
where a_c is the centripetal acceleration and g is the acceleration due to gravity. The expected centripetal acceleration is determined as,

$$a_c = \kappa v^2, \tag{5.5}$$

where κ is the curvature of the road at the given location and v is the vehicle's speed. In our experiments the curvature is determined from parametric equations of the road, $(X(t), Y(t))$, by

$$\kappa(X(t), Y(t)) = \frac{X'Y'' - Y'X''}{\left(X'^2 + Y'^2\right)^{\frac{3}{2}}}, \tag{5.6}$$

where the prime notation indicates a derivative with respect to the parametric variable t. Under this formulation, the rotation matrix is now a transformation between coordinate systems such that $\mathbf{R} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ (the vehicle provides a three-axis acceleration measurement



(a) Global Inertial Reference Frame

(b) Local Inertial Reference Frame

Figure 5.2: Example of the difference in coordinate system axes for the (a) 3D Global, and (b) 2D Local inertial reference frames for two example locations

and the inertial frame is in two dimensions), composed of rotations by the incline angle then bank angle as

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} 0 & -\sin\theta \\ 1 & 0 \\ 0 & \cos\theta \end{bmatrix} \tag{5.7}$$

$$= \begin{bmatrix} 0 & -\sin\theta \\ \cos\phi & \sin\phi\cos\theta \\ -\sin\phi & \cos\phi\cos\theta \end{bmatrix}. \tag{5.8}$$

To maintain the intuition of the rotational operation of $\mathbf{R}$, the matrix will still be referred to as a rotation matrix.

Define the matrix $\mathbf{B}$ by the inertial and vehicle frame vectors,

$$\mathbf{B} = (\mathbf{a} - \mathbf{f})\mathbf{w}^T. \tag{5.9}$$

Given that the cost function represented by Equation 5.3 can be rearranged as follows,

$$J(\mathbf{R}) = \frac{1}{2}\left(\|\mathbf{a} - \mathbf{f}\|^2 + \|w\|^2\right) - \mathrm{tr}(\mathbf{R}\mathbf{B}^T), \tag{5.10}$$

maximizing $\text{tr}(\mathbf{R}\mathbf{B}^T)$ is equivalent to minimizing the cost function. By the Singular Value Decomposition,

$$\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{5.11}$$

Define

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & d \\ 0 & 0 \end{bmatrix}, \tag{5.12}$$

where

$$d = \det(\mathbf{U})\det(\mathbf{V}). \tag{5.13}$$

Since $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices,

$$d = \pm 1. \tag{5.14}$$

By Wahba's SVD solution, $\hat{\mathbf{R}}$ is determined by

$$\hat{\mathbf{R}} = \mathbf{U}\mathbf{M}\mathbf{V}^T. \tag{5.15}$$

However reducing the dimension and using the steps from Equation 5.9 - Equation 5.15 still does not provide a unique solution for $\hat{\mathbf{R}}$. Since there is only a single measurement per sample, $\mathbf{B}\mathbf{B}^T$ has a two-dimensional null space. Therefore the last two columns of $\mathbf{U}$ (organized by decreasing singular values), denoted as $\mathbf{U}_{2-3}$, are not unique and $\hat{\mathbf{R}}$ is therefore not uniquely determined. However from knowledge of the construction of $\mathbf{R}$, we can determine which vectors in $\text{null}(\mathbf{B}\mathbf{B}^T)$ to use in constructing $\mathbf{U}$ to arrive at the unique and correct solution. Let $\boldsymbol{\Upsilon}$ be a matrix composed of orthonormal basis vectors for $\text{null}(\mathbf{B}\mathbf{B}^T)$, where $\Upsilon_{11} \neq 0$. We need to determine variables α and β to determine the last two columns of $\mathbf{U}$ such that

$$\mathbf{U}_{2-3} = \boldsymbol{\Upsilon} \begin{bmatrix} \alpha & \beta \\ \beta & -\alpha \end{bmatrix}. \tag{5.16}$$

Equation 5.16 ensures that $\mathbf{U}$ is orthogonal. There are three additional constraints:

1. To maintain $\mathbf{U}$ as a unitary matrix,

$$\alpha^2 + \beta^2 = 1. \tag{5.17}$$

2. From Equation 5.8, $R_{11} = 0$. The product of terms from Equation 5.15 that form R_{11} must therefore also equal zero. This constraint is stated as,

$$U_{11}V_{11} + dU_{12}V_{12} = 0. \tag{5.18}$$

3. For any realistic road conditions, $\cos\theta \geq 0$. Therefore

$$\text{sgn}(R_{21}) = \text{sgn}(R_{32}). \tag{5.19}$$

From these constraints, the resulting solutions for α and β are

$$\alpha = -\frac{\zeta\Upsilon_{12} + dU_{11}V_{11}}{\Upsilon_{11}V_{12}}, \tag{5.20}$$

$$\beta = \frac{\zeta}{V_{12}}, \tag{5.21}$$

where,

$$\zeta = \frac{-dU_{11}V_{11}\Upsilon_{12} + \Upsilon_{11}\sqrt{V_{12}^2(\Upsilon_{11}^2 + \Upsilon_{12}^2) - (U_{11}V_{11})^2}}{\Upsilon_{11}^2 + \Upsilon_{12}^2}. \tag{5.22}$$

Substituting Equation 5.20 and Equation 5.21 into Equation 5.16 uniquely determines $\hat{\mathbf{R}}$ as given by Equation 5.15. The resulting incline and bank angles, θ and ϕ respectively, are therefore estimated as,

$$\hat{\theta} = -\arcsin(\hat{R}_{12}), \tag{5.23}$$

$$\hat{\phi} = \arctan 2(\hat{R}_{22}, \hat{R}_{32}). \tag{5.24}$$

Extracting the angle and bank information in each vehicle results in an MSVR sampled data set once aggregated on the Cloud. The aggregated MSVR samples are then reconstructed into signals for the entire road using the reconstruction algorithms in Section 4.1.

The iterative reconstructions also offer the opportunity to improve the angle estimates of each individual sample, and therefore also improve the reconstructed signal. As given by the linear LMA update equation in Equation 4.18, the true location of each sample is re-estimated as a variable on each iteration of the LMA algorithm. The angle estimates could therefore be updated after each iteration by recalculating the inertial frame centripetal acceleration based on the updated sample location estimate $\hat{q}_j^v$ instead of the measured sample location x_j^v.

5.1.3.1 Vehicle Data Filter

An in-vehicle data filter is designed to reduce the required bandwidth of the information extraction system. This filter is created by examining the difference between the measured acceleration vector in the vehicle frame and the rotated expected acceleration vector from the local inertial reference frame.

By comparing the terms from the general rotation equation in Equation 5.2 to the cost function used to estimate $\mathbf{R}$ in Equation 5.3, it is clear that the solution for $\hat{\mathbf{R}}$ does not fully account for any perturbations or noise. Any difference between the rotated inertial frame vector and the vehicle measurement is therefore a potential indication that the measurement is too noisy or that there are unaccounted for dynamics measured in the vehicle. In either case, the resulting angle estimate is unlikely to be accurate and it would be beneficial to exclude it from any signal reconstruction.

The in-vehicle filter is therefore constructed by comparing the norm of the difference between the acceleration vectors in the cost function, D, where

$$D = \left\| \hat{\mathbf{R}}\mathbf{w} - (\mathbf{a} - \mathbf{f}) \right\|_2, \tag{5.25}$$

against a threshold, τ, and transmitting to the Cloud only the data that satisfies the inequality,

$$D \leq \tau. \tag{5.26}$$

Table 5.2: CarSim® road course components for incline and bank reconstruction experiments

Section	**Loc. (m)**	**Road Details**
Loop elevating 34 m	0-385	Euler spiral transitions straight road to circular. Incline and bank angles increase.
	385-527	Circular path. Incline and bank angles steady.
	527-750	Euler spiral, transitions circular road to straight. Incline and bank decrease to 0°.
Straight, flat	750-810	Straight, flat
Sinusoidal curves	810-1461	Two periods. 0° incline, bank angle varies to peak at curve crests in appropriate direction.
	1461-2119	Two periods. Incline decreases to 4° downhill and returns to 0°, bank continues variation with curves.
Back to flat	2119-2291	Straight, bank returns to 0°
Hills	2291-2891	Bank varies slightly, unrelated to incline
Straight, flat	2891-3100	Straight, flat

This filter eliminates poor estimates from the system, which both reduces the required bandwidth for transmission to the Cloud, and also improves the accuracy of the signal reconstruction.

5.1.4 Road and Vehicle Data

To test our model in a variety of circumstances, we require data from different vehicles driving in diverse scenarios. Obtaining such data through real-world driving is labor-intensive and expensive. Also, ground-truth road angle information is currently difficult to obtain for public roads. To address these problems, we propose using a simulator to synthesize data, in addition to testing on real-world data. Simulated data provides a large-scale, flexible, and controllable testbed to validate the algorithms, while the real-world data complements this by providing a realistic empirical test environment.

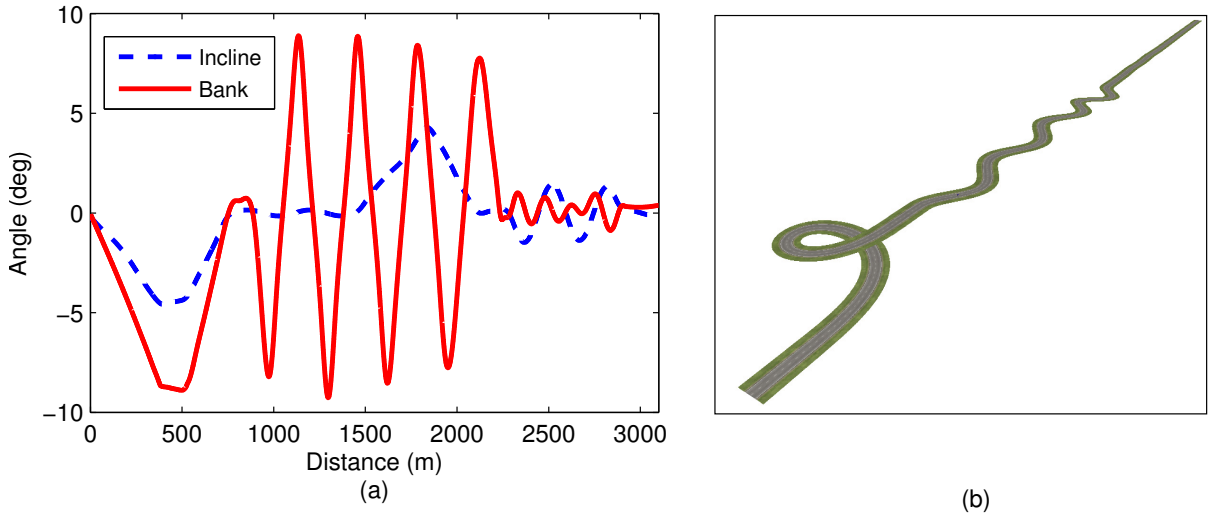


Figure 5.3: Ground-truth CarSim® (a) road incline and bank angles, (b) road course image

5.1.4.1 Simulated Data

As introduced in Section 2.8, we use the CarSim® [24] program to simulate vehicles driving over customized road courses. We designed a road course according to standard practices [84] with slight modifications to reflect real-world deviations from the ideal standard. The incline, bank, and curvature values were varied to create an elevating loop, banked curves, inclined banked curves, hills, and straight roads, as described in Table 5.2. Ground truth incline and bank angle signals for this road are shown in Figure 5.3a, with an image of the road from CarSim® in Figure 5.3b. The course was simulated driven over by standard Sedan, Minivan, and SUV models at varying speeds and accelerations.

The resulting vehicle acceleration, speed, and location data were output at 200 Hz and subsequently downsampled to 1 Hz to reflect real-world embedded vehicle sensors for use in the analysis.

5.1.4.2 Real-world Data

The real-world data was collected using a vehicle-mounted smartphone inertial sensor deployed in Pittsburgh, PA, USA [85]. The sensors asynchronously measured the GPS loca-

Table 5.3: Data sample information for the roads used in the real-world data set

Road	V	N	L (m)	Map	Curvature
Overlook Dr. (one-way)	90	26520	800	Figure 3.3	Figure 5.4
Overlook Dr. (two-way)	66	20494	550	Figure 5.5	Figure 5.6
Greenfield Rd.	31	5823	375	Figure 5.7	Figure 5.8
Beechwood Blvd.	25	5269	540	Figure 5.9	Figure 5.10

tion at 1 Hz and the three-axis acceleration at 40 Hz. These stated frequencies represent the average, since the shared data bus within the phone caused delays resulting in an irregular data output rate. Finite difference equations were used to estimate the speed and forward acceleration from the GPS data.

Four roads were selected from the data set to test with road angle reconstruction algorithms. Summary information regarding the number of data samples per road, and references to maps and curvature figures are provided in Table 5.3. These roads were selected from the data set due to their frequently changing curvature, bank, and elevation. The location measurements are from multiple sources, nonuniformly spaced, and have the types of correlated MSVR error described in Chapter 3.

It is difficult to obtain ground-truth road incline and bank angle data for real-world roads. However there are resources such as GoogleMaps which provide estimates for spot elevation data. Given locations x_1 and x_2 on a road, we can map the incline angle data to an elevation change for comparison by integrating over the reconstructed incline angle as,

$$\text{Elevation Change} = \int_{x_1}^{x_2} \tan\left(-\theta(x)\right)\, \mathrm{d}x. \tag{5.27}$$

The GoogleMaps data is not necessarily ground-truth, however it is the best available resource for comparison. Topographic survey maps were also examined, however they did not have the resolution required to extract the exact road locations from their surrounding areas for comparison purposes.

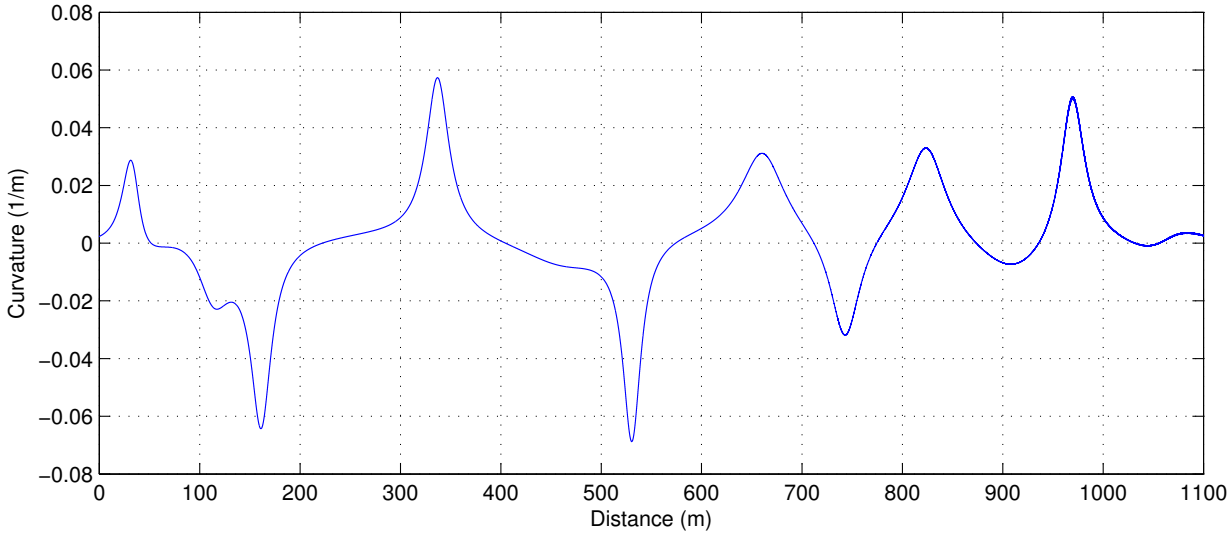


Figure 5.4: Curvature for Overlook Dr. (one-way) as determined from fitting parametric equations to the road on GoogleMaps

Figure 5.5: GoogleMaps image of Overlook Dr. (two-way) used in data set with selected GPS locations from vehicle traces

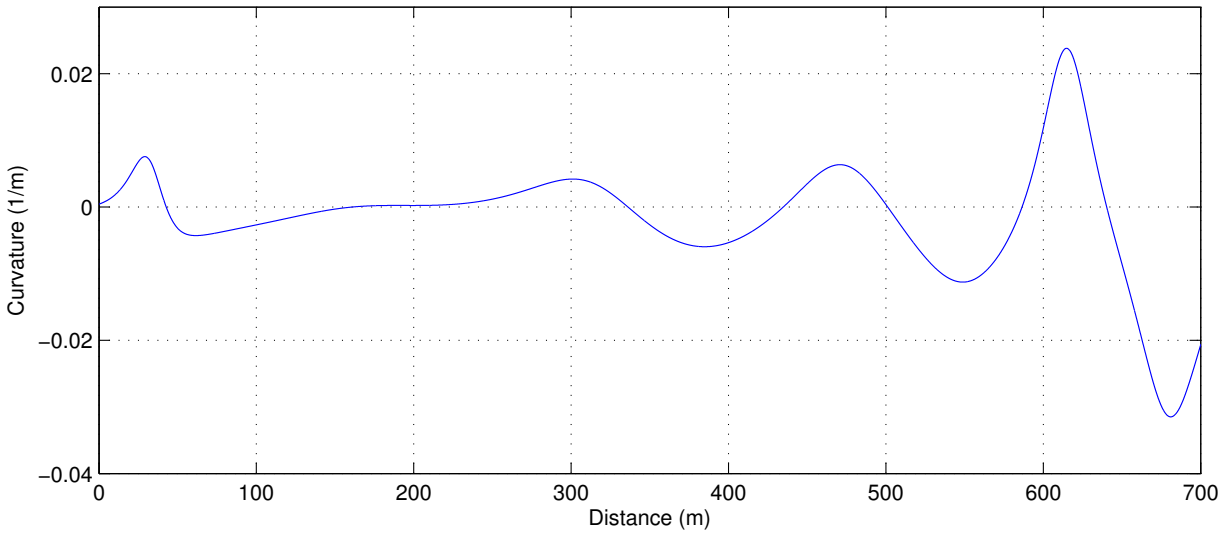


Figure 5.6: Curvature for Overlook Dr. (two-way) as determined from fitting parametric equations to the road on GoogleMaps

Figure 5.7: GoogleMaps image of Greenfield Rd. used in data set with selected GPS locations from vehicle traces

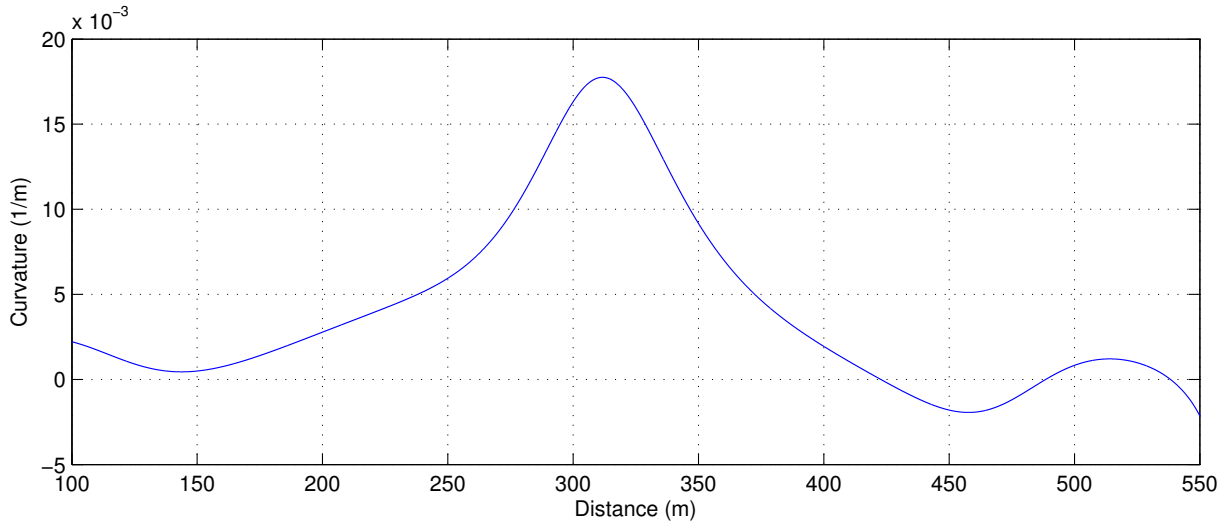


Figure 5.8: Curvature for Greenfield Rd. as determined from fitting parametric equations to the road on GoogleMaps

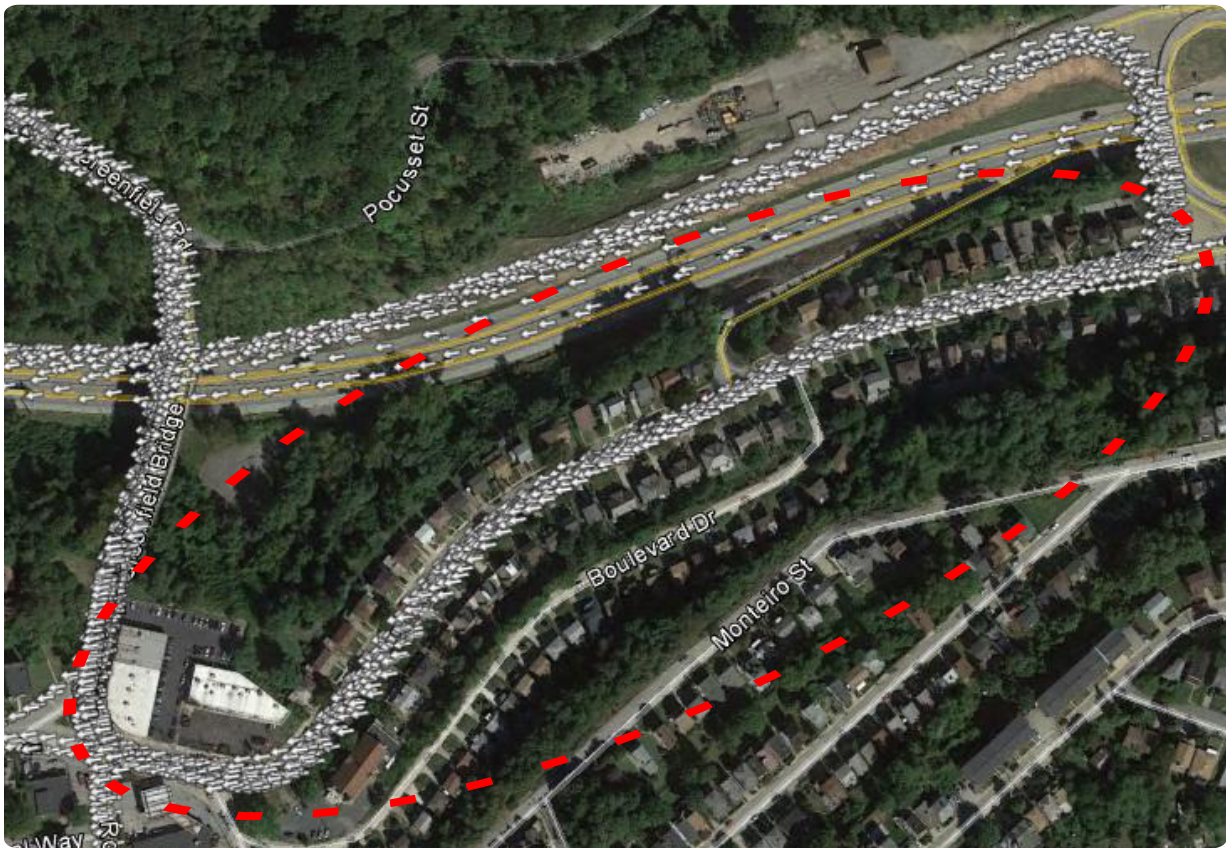


Figure 5.9: GoogleMaps image of Beechwood Blvd. used in data set with selected GPS locations from vehicle traces. The arrows in this trace do not correspond to the direction of the vehicles.

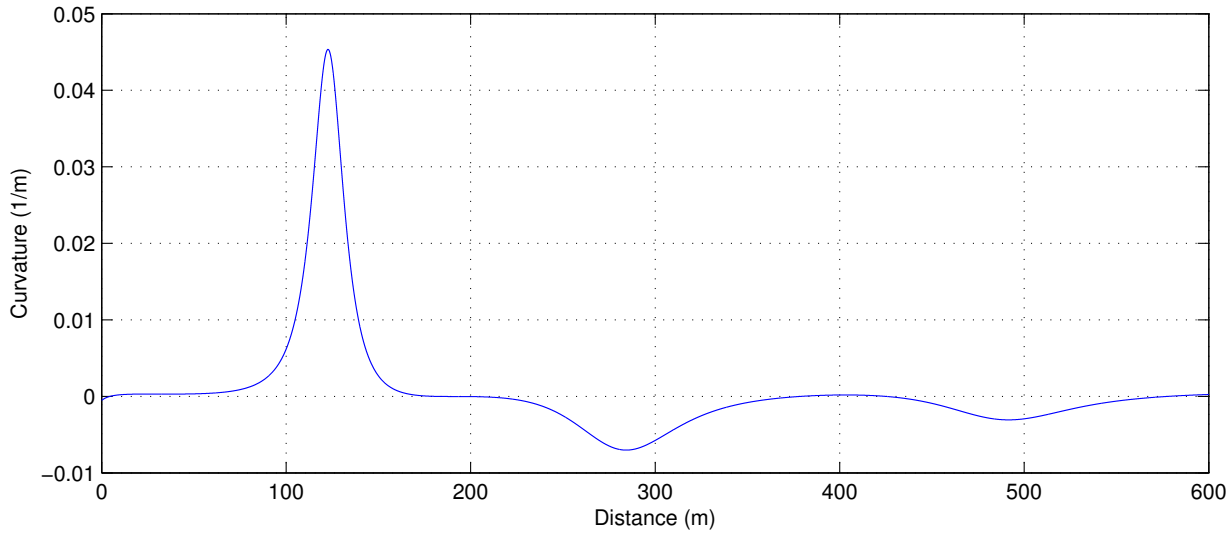


Figure 5.10: Curvature for Beechwood Blvd. as determined from fitting parametric equations to the road on GoogleMaps

5.1.5 Road Angle Results

We analyze the results of applying our road angle estimation algorithm to both simulated and real-world data. In Section 5.1.5.1, we demonstrate that the angle estimation algorithm is effective for individual noiseless samples in simulated data, and reconstruct the full continuous road angle signals from those estimated samples. In Section 5.1.5.2, we demonstrate how the iterative signal reconstruction algorithms from Section 4.1.2 can be used to re-estimate the angle values for each sample between iterations given the newly estimated locations to further improve the final signal reconstructions. Finally, Section 5.1.5.3 presents the road angle signal reconstructions for the real-world data set.

5.1.5.1 Road Angle Estimation Algorithm

We randomly selected 50 traces from the simulated data set, where the speed of the runs varied between 5 km/h - 90 km/h. No noise was added to the location or acceleration measurements for each data sample. Since $\Delta_j^v = 0 \ \forall v, j$, the final signal was reconstructed from the MSVR samples using Algorithm 4 in Section 4.1.2.4. The ground-truth, estimated

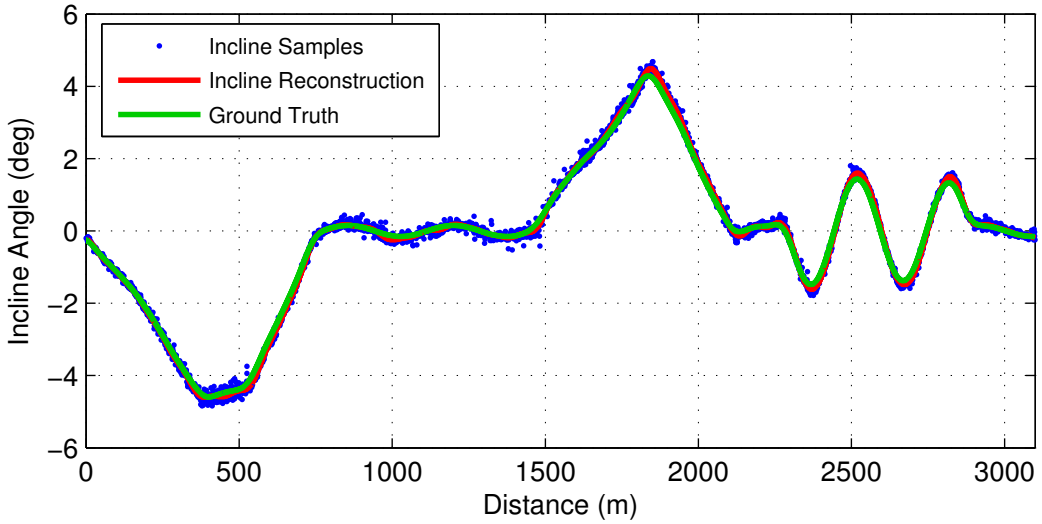


Figure 5.11: Incline angle signal reconstruction for noiseless CarSim® data.

angle samples, and reconstructed signal are shown for the incline and bank angles in Figure 5.11 and Figure 5.12 respectively for the simulated data.

The reconstructed signal aligns well with the ground truth. The RMS error for the incline and bank angle signals are 0.09° and 0.21° respectively. Some of the samples deviate from the ground truth and reconstructed signal. This occurs when the pitch and roll angles of the vehicle from the local inertial reference frame do not match the incline and bank angles of the road. This can result from driving at a speed above that intended for the road design. Added knowledge of vehicle-specific parameters could help correct for this effect [86], however the reconstruction still reliably matches the ground-truth signal, indicating that the road angle estimation algorithm is working as intended.

5.1.5.2 Correcting Locations in Reconstruction

Estimating the true sample location values, $\hat{q}_j^v$, for each sample is important for reconstructing the signal, however as described at the end of Section 5.1.3, for algorithms such as the road angle reconstruction, estimating $\hat{q}_j^v$ offers a second benefit.

The angle estimation algorithm in Section 5.1.3 reduces the global inertial reference

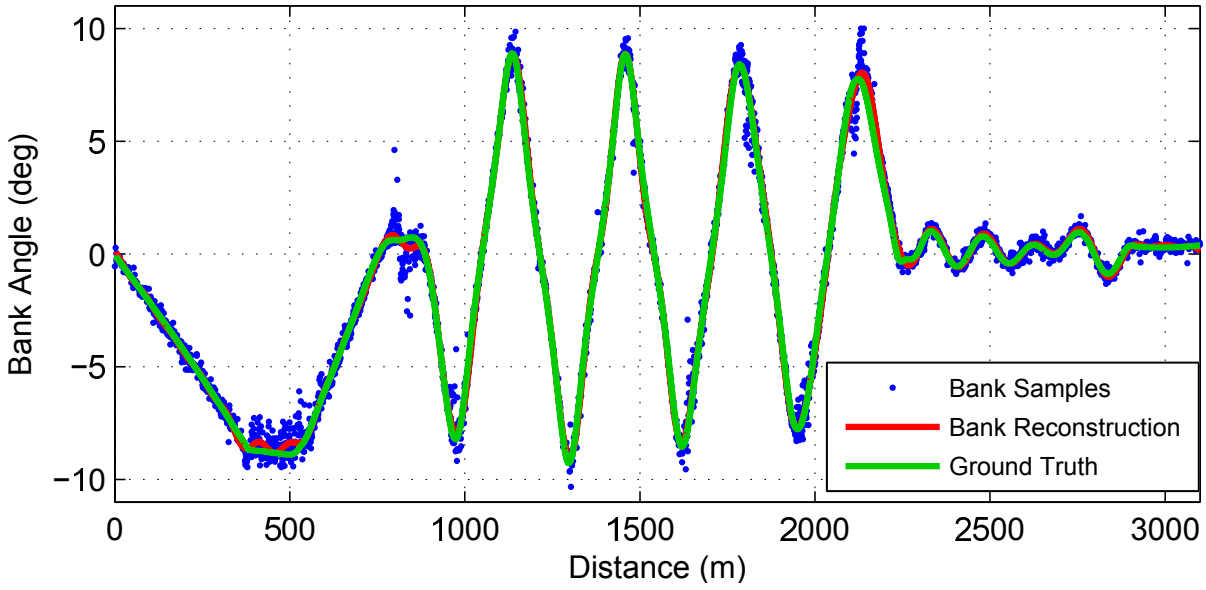


Figure 5.12: Bank angle signal reconstruction for noiseless CarSim® data.

frame from three dimensions to two dimensions by creating a local inertial reference frame aligned with the road. The alignment is determined by the GPS location with one axis aligned vertically and the second orthogonal axis aligned on a flat plane perpendicular to the direction of the road. The origin and direction of these axes determine the curvature of the road for the specific data sample, as given by Equation 5.6. This curvature value is ultimately used in Equation 5.5 to calculate the expected vehicle centripetal acceleration. If the measured GPS location differs from the true location, then the road curvature calculated in Equation 5.6 may be incorrect, resulting in a poor estimate for the inertial frame acceleration vector in Equation 5.4. These errors could propagate to produce poor estimates for the incline and bank angles.

However since $\hat{q}_j^v$ is re-estimated on each iteration of the LMA minimization algorithm, the road and incline angles can be re-estimated using $\hat{q}_j^v$ to redetermine the origin of the local inertial reference frame instead of using the origin originally determined by x_j^v. This improves the reconstruction on each iteration as not only do estimated locations of each angle sample approach their true values, but the angle estimates are also improved to

Table 5.4: Root-mean-square error angle values (degrees) for both re-estimating and not re-estimating angle values between LMA iterations. Results correspond to data shown in Figure 5.13.

	RMSE (deg)	
	No Angle Re-estimation	Angle Re-estimation
Bank	0.25	0.16
Incline	0.95	0.66

better resemble their true values as well.

An example of this process in action is shown in Figure 5.13 for reconstructing the incline and bank angles from the simulated data from 50 vehicles with location error given by $\sigma_v = 70$ m, and $\sigma_x = 15$ m. Due to the sample location bias error being relatively larger than the independent component, reconstructions are shown for only the Fixed Error joint reconstruction model. However the results extend similarly to the other reconstructions. The RMSE values for the different reconstruction types are provided in Table 5.4. There is a definite advantage to re-estimating the angle values between iterations, marked by the improvement in the RMSE. The advantages would be much larger for situations where the curvature (or any variable dependent on location) has much larger variations with respect to the magnitude of the location errors.

5.1.5.3 Real-world Data

The MSVR signal reconstruction algorithms were applied to the real-world data set to determine the incline and bank angle signals for the city roads detailed in Section 5.1.4.2. The in-vehicle data filter from Section 5.1.3.1 was applied with threshold $\tau = 0.9$ m/s^2. This threshold eliminated 20%-30% of the data samples for each of the real-world examples. The elevation change equation, given by Equation 5.27, requires a reference point for comparison. The elevation obtained from GoogleMaps at the start of the road was used as this initial reference value.

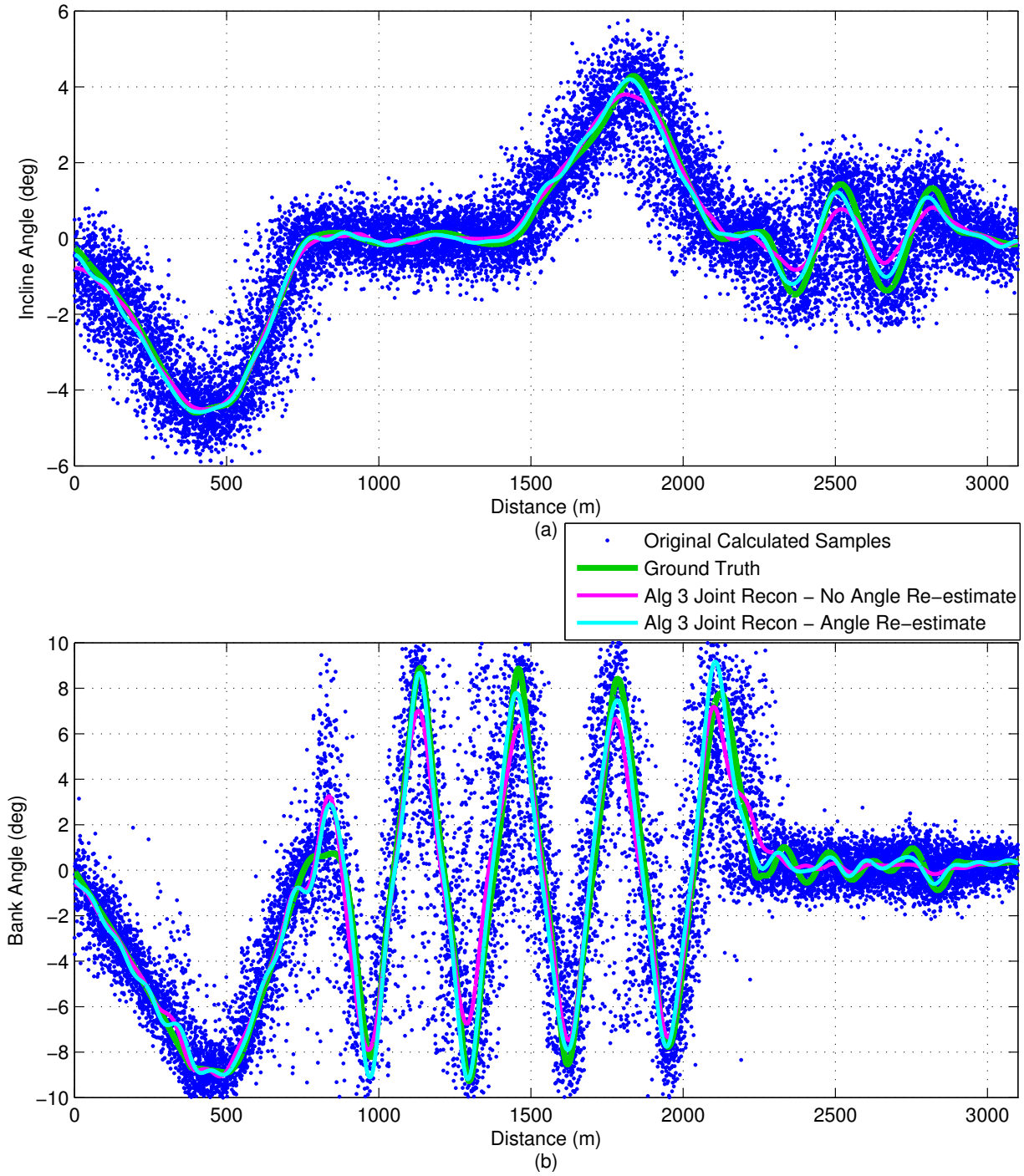


Figure 5.13: Reconstructed bank and incline angles from 50 vehicles with white noise with standard deviation 0.08 m/s^2 added to acceleration signals and $\sigma_v = 70$, $\sigma_x = 15$ error, using the Fixed Error joint reconstruction algorithm, when both re-estimating and not re-estimating the incline and bank angles between LMA iterations

The incline and bank angles are shown in Figure 5.14a and Figure 5.14b respectively for Overlook Dr. (one-way). Although there is a fair amount of noise in calculating the angles for each individual data sample, the reconstructed signal has a reasonable range. To validate the results, the incline angle was converted to an elevation signal by Equation 5.27. The results are shown in Figure 5.15 as compared to the cubic spline interpolation of the spot elevation data obtained from GoogleMaps. The RMS error between the two signals is 0.89 m, and the two signals exhibit similar properties over the length of the domain.

We further compare the results from the other three roads in the data set. The incline and bank angles for the two-way component of Overlook Dr. are provided in Figure 5.16, with the mapped elevation data compared to GoogleMaps shown in Figure 5.17. The elevation change of the reconstruction was -9.0 m compared with -7.25 m from GoogleMaps. The two signals follow approximately the same pattern, with the largest difference coming from the larger decline towards the end of the road in our reconstructed signal.

The incline and bank angles for the Greenfield Rd. reconstruction are provided in Figure 5.18, with the respective elevation comparison to GoogleMaps in Figure 5.19. The reconstruction was done for both the autocorrelated error model and for the fixed error model. Both results compare reasonably well to the GoogleMaps data, with elevation changes of -12.5 m and -13.3 m for the autocorrelated and fixed error model reconstructions respectively, and -11.4 m from GoogleMaps. The RMS errors are 1.37 m for the autocorrelated error model and 1.76 m for the fixed error model. Both the reconstructions and the GoogleMaps data share the property of the road flattening out near the end before again declining.

The incline and bank angles for the Beechwood Blvd. reconstruction are provided in Figure 5.20, with the respective elevation comparison to GoogleMaps in Figure 5.21. The reconstructions were again performed for both the autocorrelated and fixed error models. These elevation signal reconstructions significantly differ from the GoogleMaps data. The

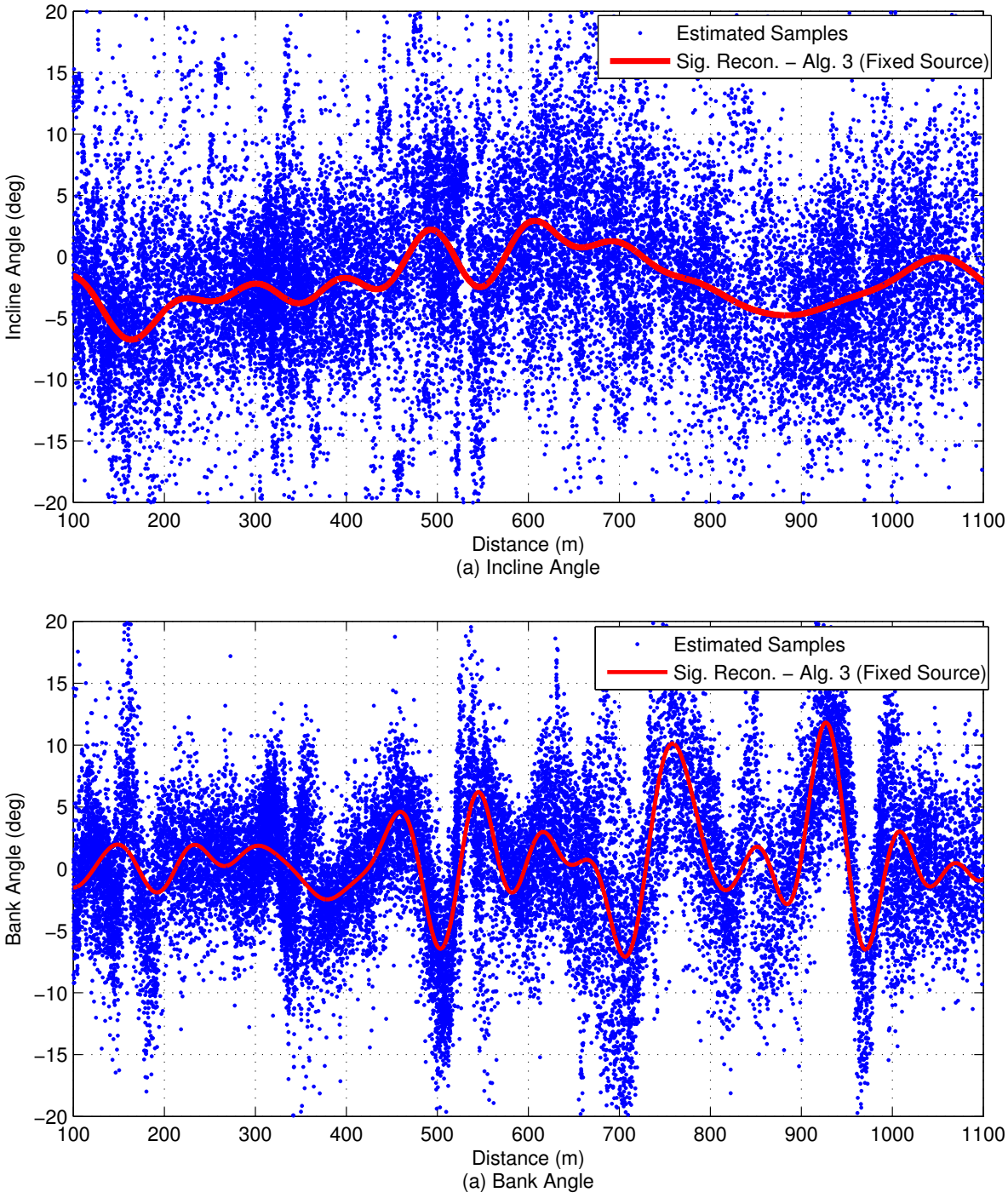


Figure 5.14: Reconstructed (a) incline and (b) bank angles for real-world data from Overlook Dr. (one-way)

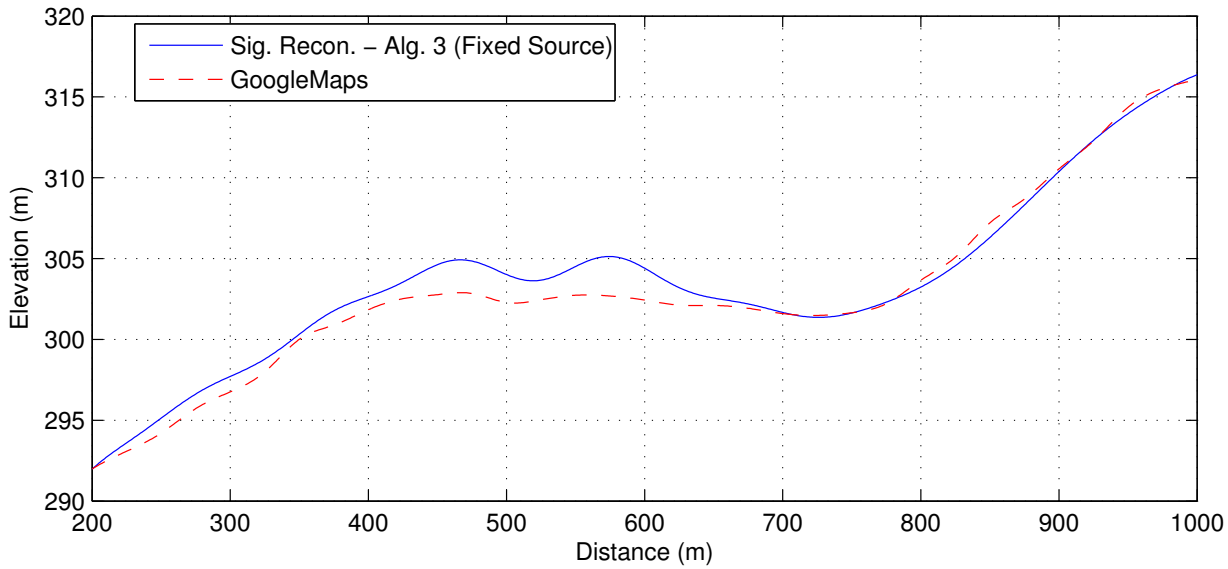
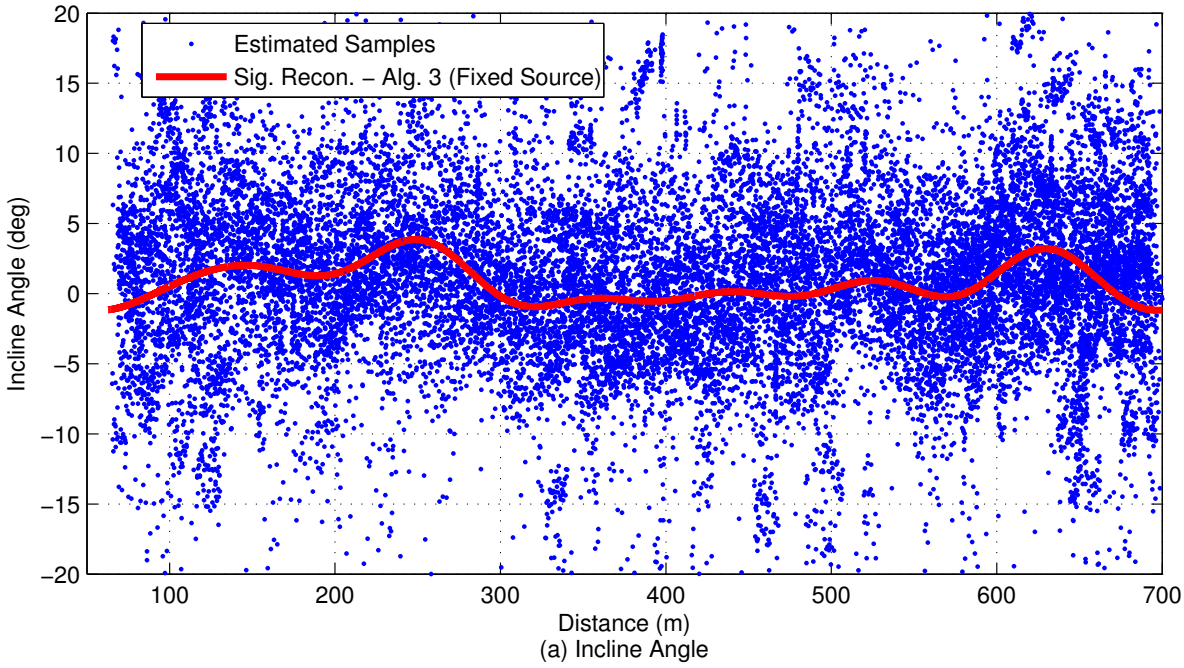


Figure 5.15: Reconstructed real-world Overlook Dr. (one-way) road elevation signal compared to GoogleMaps data

autocorrelated and fixed error models estimate elevation changes of -11.5 m and -12.2 m respectively, while GoogleMaps data provides for a relatively flat road with and overall elevation change of just -0.6 m. However from video inspection, displayed by the example road image in Figure 5.22, there does appear to be a steady decline to the road that is more consistent with our signal reconstructions than the GoogleMaps data.

Since we integrate our reconstructed incline signal to compare to GoogleMaps, any slight error bias in the reconstructed signal will manifest itself as diverging signals in the elevation comparison. Therefore for some figures, a slight error even in one part of the incline signal, could make it appear as if the two signals are significantly different. It is therefore important to also look at the general pattern of the two signals. It would be beneficial as well to differentiate the elevation signal to compare it as an incline signal to our reconstruction. However since it is not known exactly how the GoogleMaps elevation data is determined, and the fact that the signal is created by interpolating splines on the spot elevation data, it could create questionable dynamics in the incline signal if differentiated.

Overall, the road angle reconstruction estimates and signal reconstructions perform very



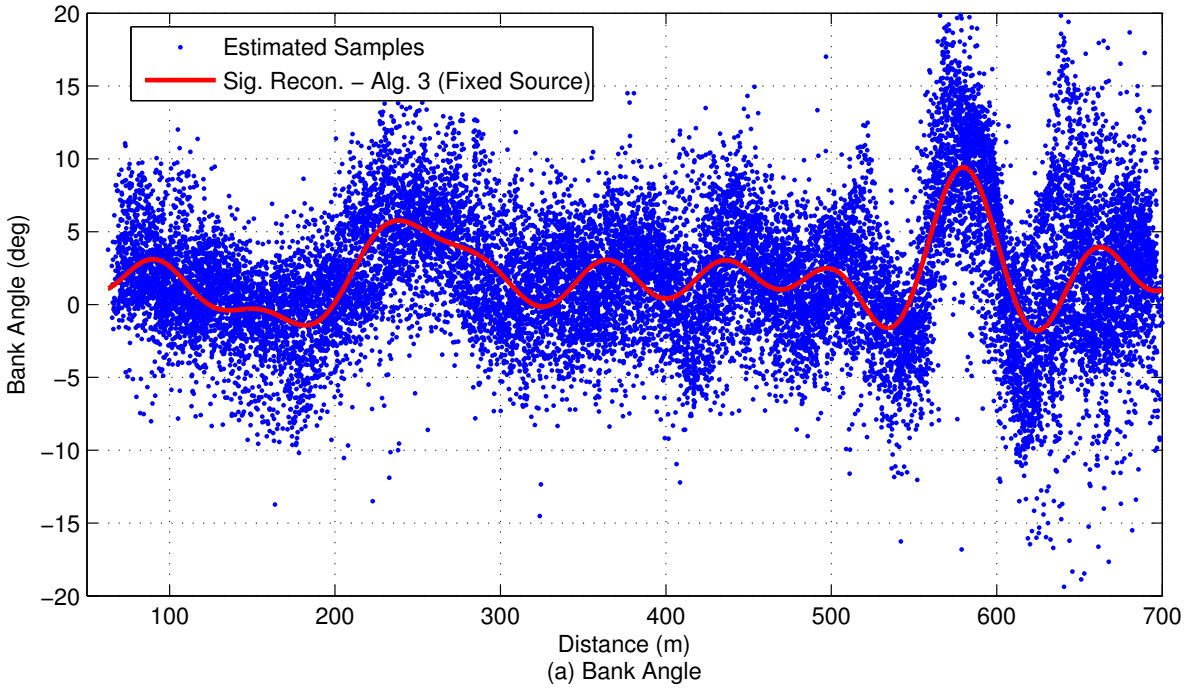


Figure 5.16: Reconstructed (a) incline and (b) bank angles for real-world data from Overlook Dr. (two-way)

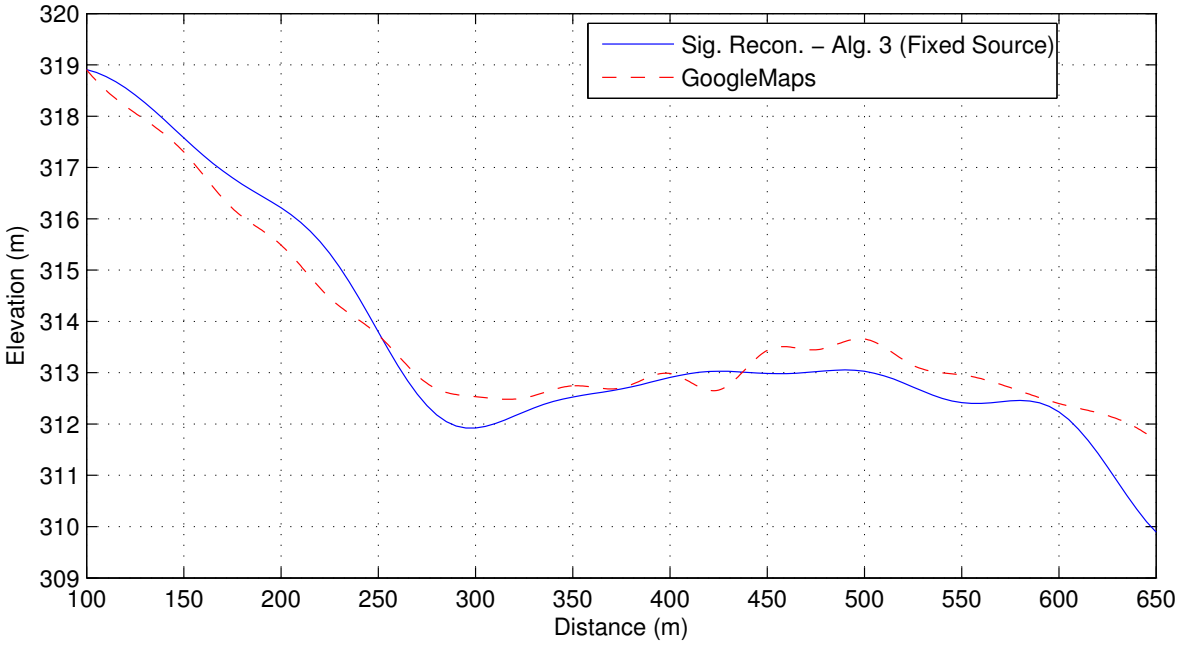
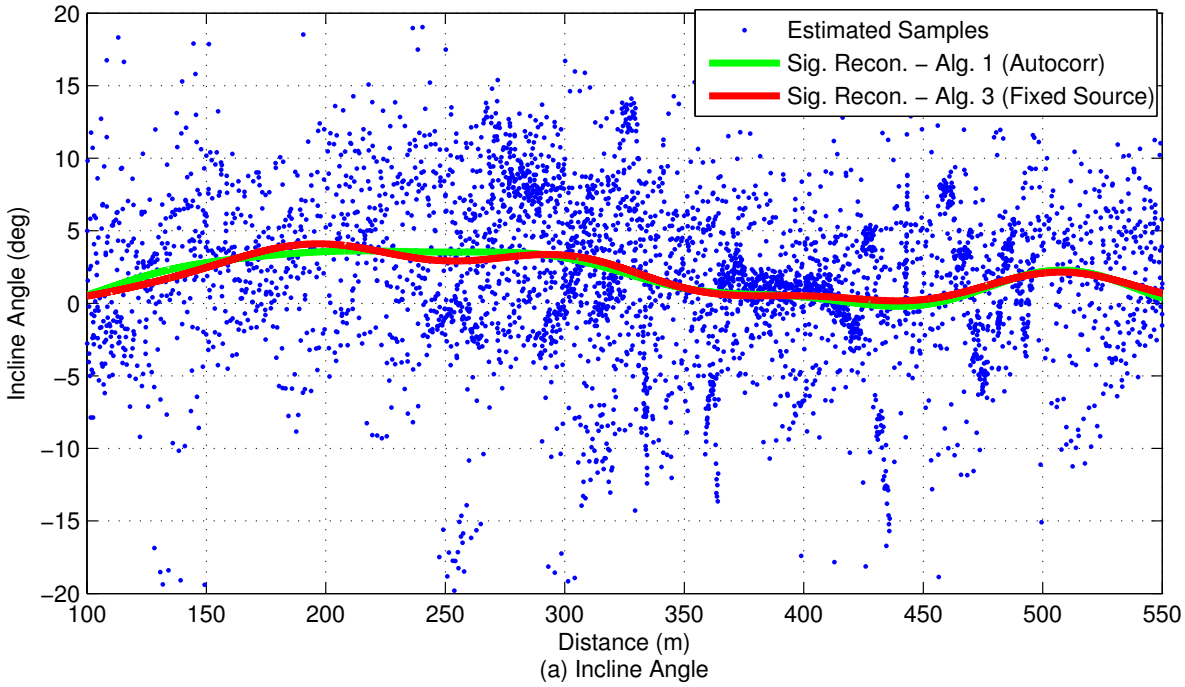


Figure 5.17: Reconstructed real-world Overlook Dr. (two-way) road elevation signal compared to GoogleMaps data

well for the simulated data set. Different reconstructions were compared depending on the location error model, and the benefits of re-estimating angle values jointly while estimating the reconstructed signal were demonstrated. The angle reconstructions were also performed for real-world vehicle sensor data. The reconstructed incline and bank angle signals were presented with comparisons to elevation signals obtained from GoogleMaps data. The elevation signals compared favorably, with any differences explained with references to images from the specific roads. This evidence, in addition to the mathematical derivation of the angle estimation algorithm, supports the application of the algorithm to well estimate the true road bank and incline angle signals.



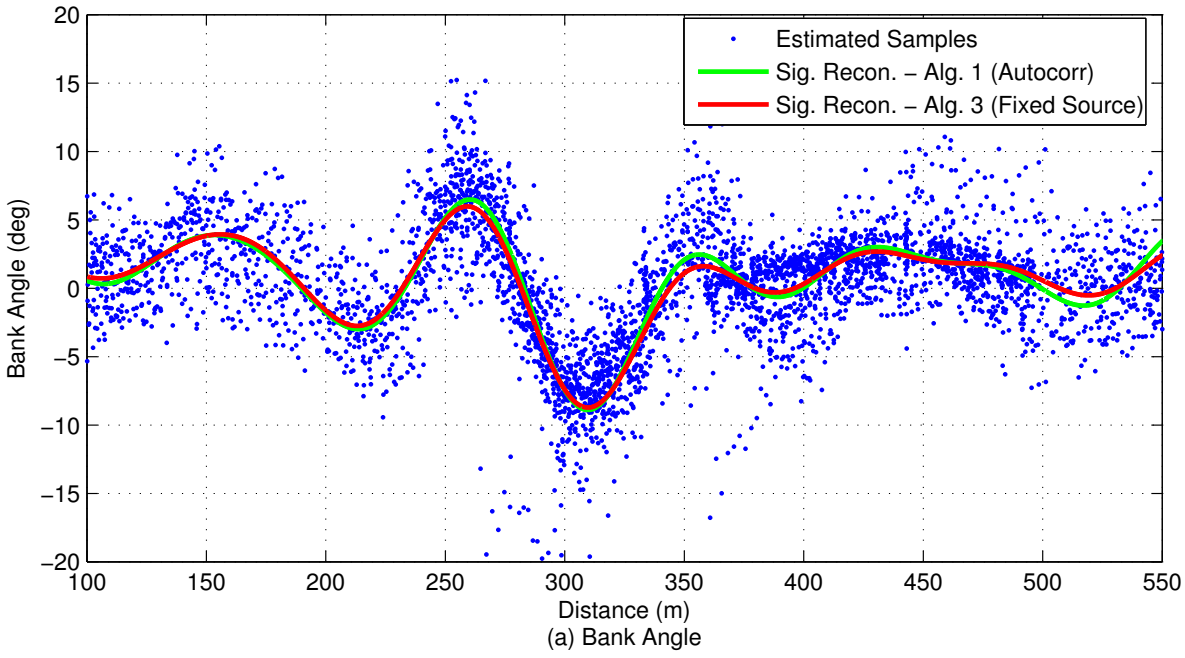


Figure 5.18: Reconstructed (a) incline and (b) bank angles for real-world data from Greenfield Rd.

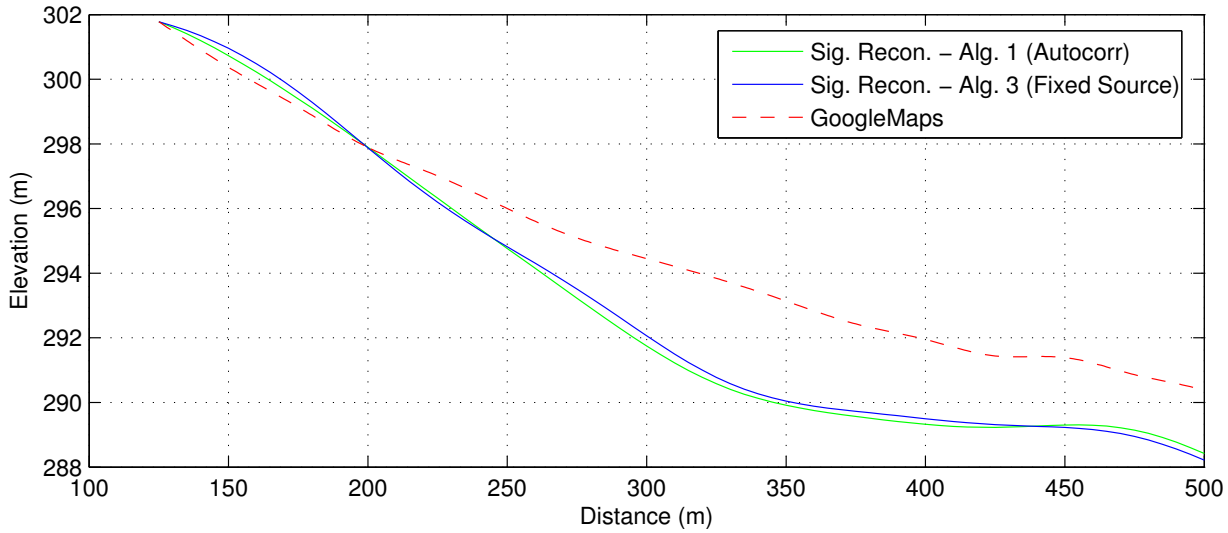


Figure 5.19: Reconstructed real-world Greenfield Rd. road elevation signal compared to GoogleMaps data

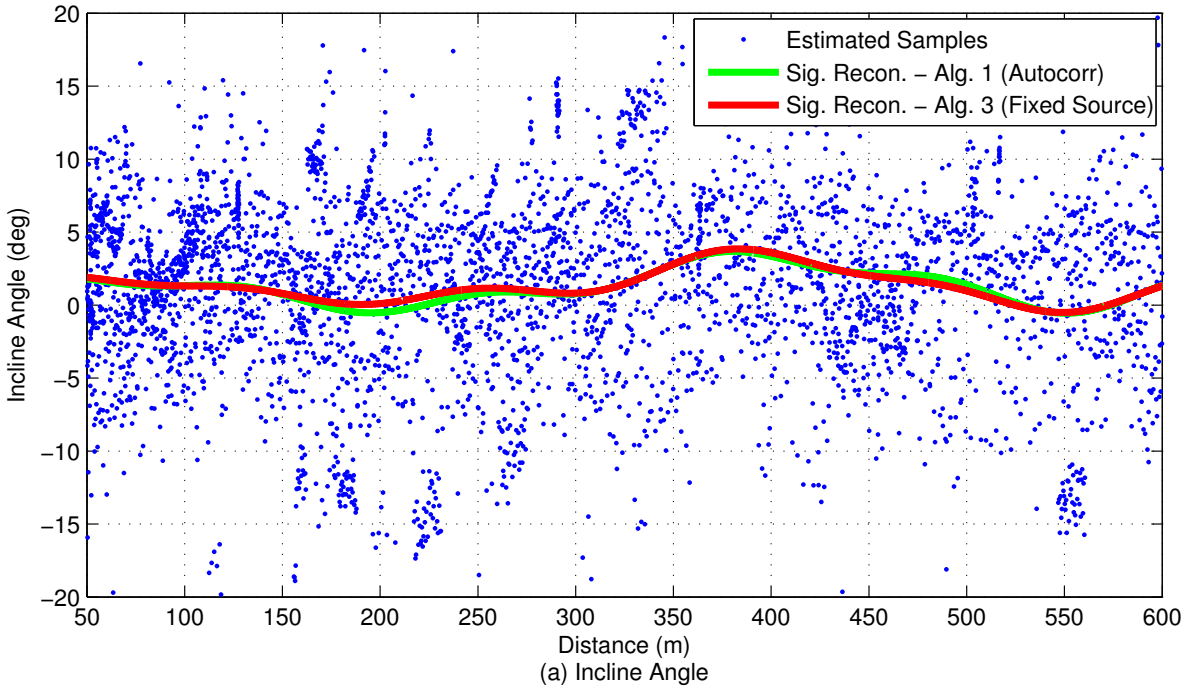
5.2 Pothole Detection

5.2.1 Introduction

Detecting anomalous road features is critical for driver and vehicle safety. One such example of an anomalous road feature is a pothole. Potholes are hazardous road features that damage wheels, suspensions systems, vehicle frames, and potentially injure drivers and passengers. They are responsible for millions of dollars in insurance claims and roadway repairs each year [87]. Drivers would benefit from the knowledge of pothole locations in planning their routes and an automated detection system would assist municipalities in planning repairs. Automatic suspension control systems would also be able to adjust to anticipate potholes instead of reacting after hitting them.

In this thesis work, we follow the binary event detection framework described in Section 4.2 to create a system to detect and localize potholes in multi-lane environments. The detection process is conducted under the functional constraints and noisy MSVR sampling conditions of vehicle sensor data detailed in Chapter 3.

Embedded vehicle accelerometers serve as our data source for the pothole information



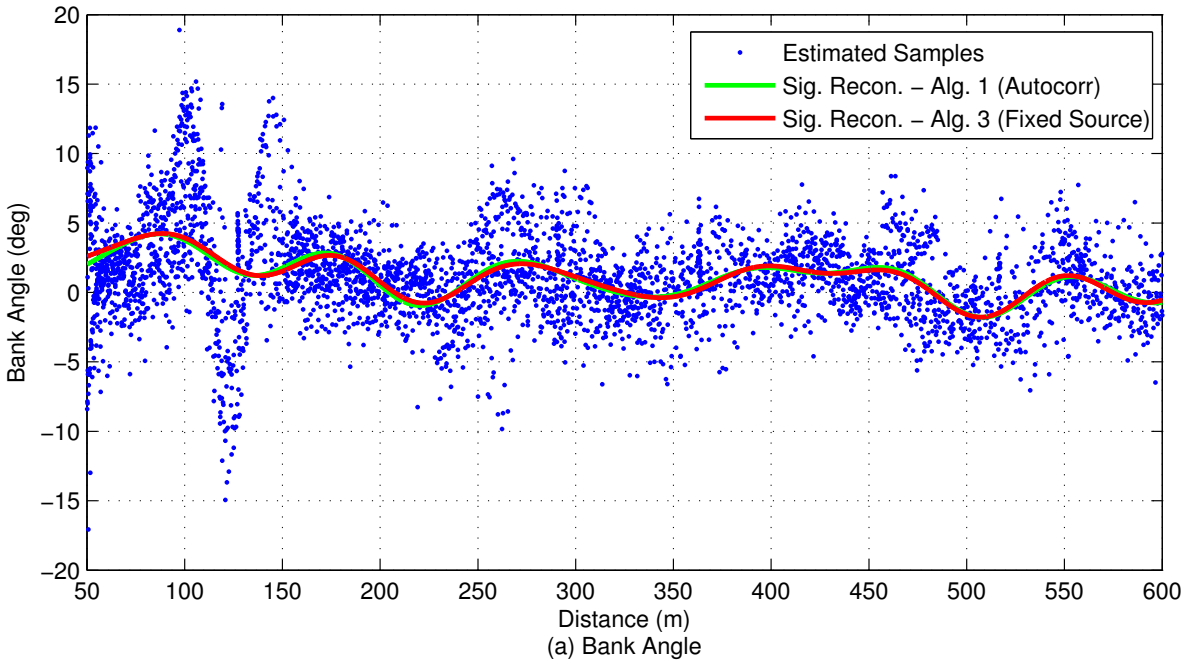


Figure 5.20: Reconstructed (a) incline and (b) bank angles for real-world data from Beechwood Blvd.

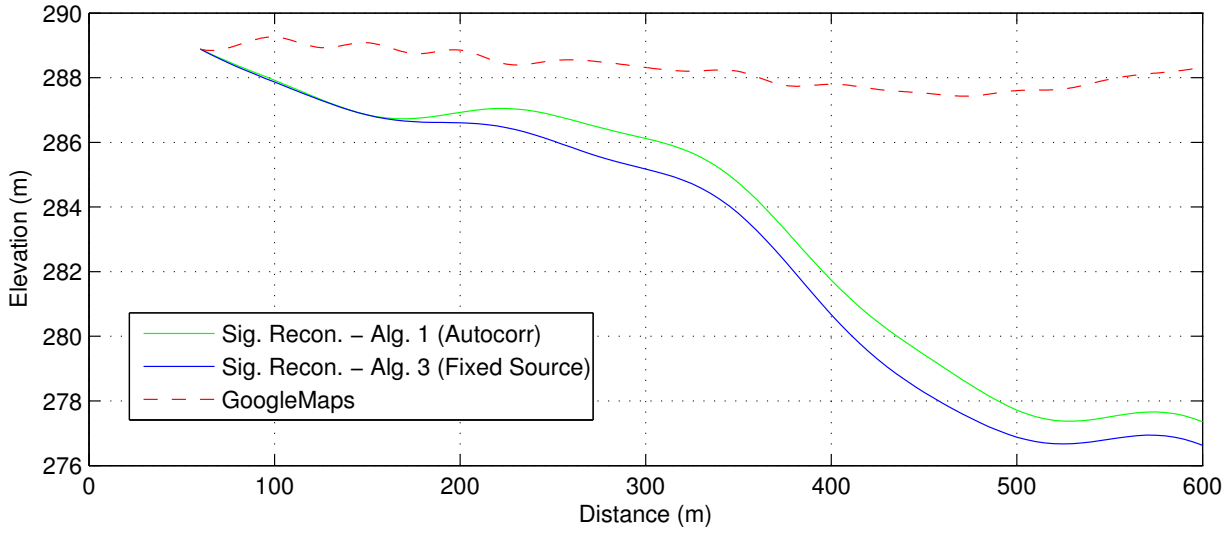


Figure 5.21: Reconstructed real-world Beechwood Blvd. road elevation signal compared to GoogleMaps data

Figure 5.22: Live capture image of Beechwood Blvd.

extraction objective. The low sampling frequency (e.g., 1 Hz) of such sensors result in unreliable detection from individual vehicles and necessitate the crowdsourcing framework to increase the detection rate. For example, a vehicle traveling at 50 km/h would only make a measurement every 13.9 m, an interval that significantly exceeds the length of a typical pothole. This spacing results in a low probability that the sensors in a single vehicle capture a discriminating measurement at the appropriate pothole location.

In addition to the typical noisy MSVR sampling conditions, aggregating accelerometer data from multiple vehicles is complicated by vehicle and sensor heterogeneity. The accelerometer data of a vehicle driving over a pothole depends on that specific vehicle's response to the pothole. Since vehicles differ in physical properties such as size, weight, length, and suspension systems, as well as driver behaviors such as steering and speed, different vehicles respond differently even when driving over the same pothole. Therefore, the expected signal from each vehicle is different.

Similarly, not all potholes are identical. The position of the pothole on the road, the curvature and incline of the road, and the length and depth of the pothole all affect the expected accelerometer signals. The derived detection framework needs to be generally applicable to all these varied road and driving conditions and care needs to be taken to not consider acceleration components related to physical phenomena other than potholes in the detection system. This is an important consideration when generating the candidate feature list from the signal data.

The remainder of this section is outlined as follows. Related work to the pothole detection problem is presented in Section 5.2.2. The simulated and real-world data used for the pothole detection experiments is described in Section 5.2.3. The architecture of the pothole detection system is described in Section 5.2.4. Results from the detection system are described in Section 5.2.5.

5.2.2 Related Work

Pothole detection has been the focus of some previous works, however the sensor device has generally been a smartphone instead of embedded vehicle sensors. We focus on the use of embedded sensors since they are standardized across vehicles and are integrated into the controls and communications systems of the vehicle. This makes them a much more logical resource as the detection system can be integrated into the existing framework of the vehicle and Cloud system. Smartphones generally have a much higher sensor operating frequency (300+ Hz) than embedded vehicle sensors. These higher frequencies allow for a detection system to measure the full dynamic motion of the vehicle caused by potholes or other road features. This is important since when we transition to using low frequency acceleration signals we lose some of the signal properties that distinguish pothole regions in a single vehicle since so little data is available. Signal undersampling has been shown to be problematic for other vehicular applications such as speed estimation [88].

Regarding existing pothole detection methods, the Pothole Patrol system [89] uses speed, high-pass, and vertical and lateral acceleration filters to identify potholes from test signals from Boston taxis. Road bumps are detected in [90] by examining the peak vertical acceleration and the duration for which the acceleration dips below a heuristically defined threshold. Gaussian Mixture Models are used in [91] on aggregated data to determine potholes from 100 taxis in Shenzhen, China by examining z-scores of listed Pothole Patrol features [89]. A linear model for speed using 38 Hz sensors was constructed to try to eliminate the speed dependence in [92], however the vertical acceleration, which is used to create most of the features, can deviate significantly from the linear model resulting in poor adaptations to varied speed.

Since these works use non-embedded high-frequency sensors, they rely on only one vehicle for detection. They therefore ignore problems resulting from GPS error since they only consider a single trace where each vehicle must hit a pothole and are not focused on

determining the exact pothole location. The GPS error increases classification difficulty, as some normal road regions resemble pothole regions, and vice-versa, due to errors in the reported locations. This is particularly problematic when aggregating data from multiple vehicles with different dynamics as the exact ordering of the measurements cannot be determined. These, and previous works which use crowdsourcing detection techniques for pothole detection [3], do not consider multi-lane scenarios where GPS position error obfuscates the pothole data with normal road data from adjacent lanes. We also address the issue of finely localizing the pothole longitudinally on the road, which has not been addressed in previous works.

Our proposed system uses accelerometer data and is therefore reliant on a few initial vehicles being unable to avoid the potholes. The information obtained by those vehicles that could not avoid driving over the potholes can be shared with subsequent vehicles for their benefit. In comparison, there are other systems which attempt to help drivers by pre-detecting potholes in real-time by using images from vehicle-mounted cameras instead of accelerometers. However, even with the advantage of being able to picture the pothole before the vehicle enters it, the required angle of the camera and subsequent processing time does not necessarily provide the driver with sufficient opportunity to avoid the pothole. With regard to the image-based methods, the work in [93] looks for pothole regions in images via segmentation methods employing shape-based thresholding, and examining geometric and texture based properties of suspect pothole regions. Similarly, large simulated potholes are found in images in [94] by looking for large circular objects with a predetermined brightness difference. Unlike accelerometer-based detection methods, image-based methods require specific lighting conditions to function properly and may be unusable at night or in poor weather conditions.

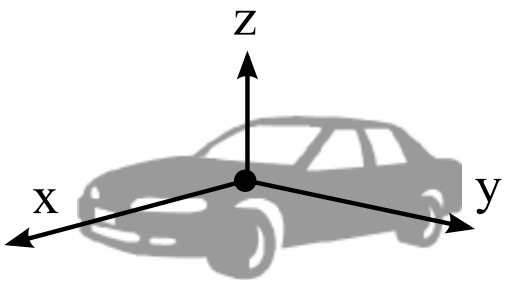


Figure 5.23: Coordinate system in relation to the vehicle for pothole detection

5.2.3 Pothole Data

Similar to the road angle estimation experiments as outlined in Section 5.1.4, we require data from different vehicles driving over potholes in diverse scenarios in order to create an extensive model capable of detecting potholes in a variety of circumstances. We also require ground-truth information for training the detection models. Obtaining such data through real-world driving is prohibitively labor-intensive and expensive. Therefore, similar to the road angle experiments and as outlined in Section 2.8, we use a simulator to synthesize training and testing data for our model and augment the system by testing the models created using the simulated data on real-world data. The following sections outline the simulated and real-world data used for the pothole detection problem. The data and variables are defined in reference to the coordinate system shown in Figure 5.23.

5.2.3.1 Simulated Data

As detailed in Section 2.8, we use the CarSim® program to simulate vehicles driving over potholes. With CarSim®, we simulated vehicles driving a total of 5220 km in diverse conditions over potholes 22950 times with known ground-truth locations. These totals vastly exceed what can be collected and labeled manually, and provides us with a basis for creating models to subsequently apply to real-world data.

We designed an 8.7 km long, two-lane road course featuring different degrees of curvature, incline, and bank that one may expect to find in real-world scenarios. The road

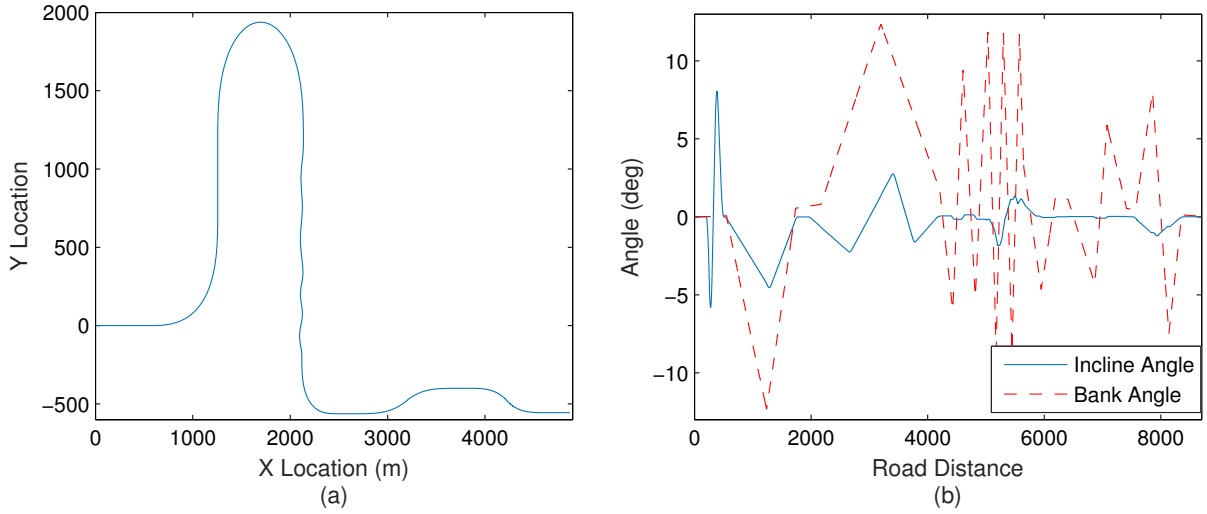


Figure 5.24: Ground truth simulated road details, (a) Overhead road view, (b) Incline and Bank angles

course was designed according to standard practices [84] with slight modifications to reflect real-world deviations from the ideal standard. The road topography including an overhead view, and incline and bank angles is shown in Figure 5.24. The course contains many more features than what would be typical for an 8.7 km road, however this allows us to test for potholes under difficult conditions. The non-pothole features of these courses were meant to induce vehicle acceleration responses unrelated to potholes. Using the additional features, the detection system could be trained to avoid false positives.

The terrain of the normal road is an important consideration for road course design, particularly when comparisons are made to potholes. For a flat distance l traveled on the road, where l is determined by the arc length of the parametric equations for the road $(X(t), Y(t))$ as

$$l\left(X(t), Y(t)\right) = \int_0^t \sqrt{\left(X'(T)\right)^2 + \left(Y'(T)\right)^2}\, dT, \tag{5.28}$$

where the prime notation indicates the derivative with respect to the parametric variable, the road roughness profile, $h(l)$, was constructed using the 2D Weierstrass-Mandelbrot

(W-M) function [95],

$$h(l) = L\left(\frac{G}{L}\right)^{D-1}\sum_{n=1}^{n_{\max}}\frac{\cos\left(\frac{2\pi\gamma^n l}{L}+\phi_n\right)}{\gamma^{(2-D)n}}, \tag{5.29}$$

to represent a paved asphalt road. In the W-M function, L is the length of the road, n is the frequency level index, $n_{\max}$ is the maximum frequency level, γ is the relative phase difference between fractal modes (set equal to 1.5), and ϕ_n is a random phase shift for frequency n uniformly distributed in the range $[0, 2\pi)$. The parameters D and G are the fractal dimension and fractal roughness parameters, respectively, chosen based on the power spectral density of the surface distribution,

$$\phi(\omega) = R\omega^k, \tag{5.30}$$

where

$$D = \frac{k+5}{2}, \tag{5.31}$$

$$G = (2R\ln\gamma)^{\frac{1}{2D-2}}. \tag{5.32}$$

The W-M function has been shown to well approximate road roughness and its effect on vehicle dynamics [96, 97]. No road is uniform across the width of the road so four different profiles were generated for the two-lane simulated road: one profile for each side of the vehicle in each of the two lanes. These profile locations represent the tire contact points on each side of the vehicle when driving in each of the respective lanes. The four different profiles were constructed by randomly shifting each of the phase shifts from the road profile in the adjacent lane. Potholes were simulated by adding a constant dip of the respective pothole size to the roughness profile. The details of the simulated potholes are shown in Table 5.5 and represent varying lengths and depths for shallow potholes with no structural failure below the top of the base layer beneath the road surface [98]. An example of the roughness profiles for a 0.8 m long, 6 cm deep pothole is shown in Figure 5.25.

Table 5.5: Simulated road pothole details

Quality	**Details**
Length	$\{0.2, 0.4, 0.6, 0.8, 1.0, 1.2\}$ m
Depth	$\{2, 4, 6, 8, 10\}$ cm
Location	Every 100 m, Randomly chosen on one side of one of the two lanes

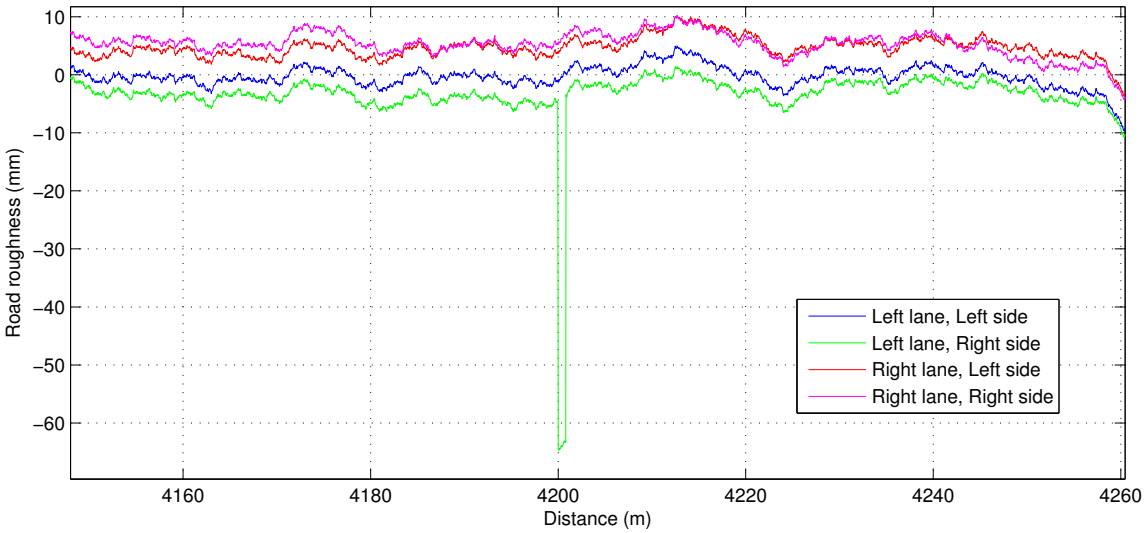


Figure 5.25: Road roughness data for the profiles on each side of each lane for the simulated data with a 0.8 m long, 6 cm deep pothole on the right side of the right lane

For testing purposes we recreated two road types by appropriately selecting runs from the simulated data set. The first road is a city road where the vehicles traveled at an average speed of 50 km/h with a 10 km/h standard deviation. The second road is a highway road where the vehicles traveled at an average speed of 100 km/h with a 15 km/h standard deviation. The vehicles were randomly selected to drive in one of the two lanes with equal probability. Experiments testing single-lane roads were done by simulating vehicles in only the left lane.

The data for each simulation was output at 200 Hz, then downsampled to 1 Hz with randomized starting points to emulate the low-frequency data acquisition of real-world vehicle embedded sensors. Data on the following variables was collected:

Table 5.6: Standard deviation on error added to measured sensor data from simulations

Sensor	**Standard Deviation**
GPS	4 m i.i.d. for each vehicle plus 0.3 m i.i.d. each individual sample Errors added in both coordinate directions (i.i.d.)
Speed	3.6 km/h
Acceleration	0.1 m/s^2

- GPS coordinates, (x, y)
- Vehicle speed, v
- Longitudinal, lateral, and vertical accelerations (a_x, a_y, a_z), respectively at the vehicle center of mass

Noise was added to the simulated data to better represent real-world deviations. Table 5.6 shows the standard deviations for each of the sensors. All noise is i.i.d. zero-mean, and normally distributed unless otherwise indicated. These values represent typical GPS errors for short time frames [26].

5.2.3.2 Real-world Data

Real-world data was collected by driving repeated laps on a square of multi-lane city roads in Warren, MI, USA. Each lap covered 6.6 km. The trip was repeated 8 times in each of the left and right lanes. Since the multi-lane scenario is the focus of this experiment, we exclude data from the beginning and end of each road segment where the vehicle had to merge into or from the right lane for turning.

Data was collected from the embedded vehicle sensors for use in the pothole detection system. Although the simulations assumed that each sensor operated at 1 Hz, due to manufacturer specifications, some of the embedded sensors deviated from this frequency. The GPS location was updated at 1 Hz. The magnitude of the vertical acceleration updated at 1 Hz while the lateral acceleration updated at 50 Hz. The speed was updated at 10 Hz.

Due to the shared CAN bus and the asynchronous measurements made from the different sensors, the output times for the sensor data were not perfectly periodic and were instead often bunched or had larger delays. This was demonstrated by the previously presented example speed data in Table 3.3.

A video feed of the trip from a dashboard-mounted smartphone was used to visually identify potholes for ground-truth reference. The timestamps on the video were mapped to the GPS locations from the phone to identify the pothole locations. Lane locations were determined by fitting parametric curves to Google Earth data. A total of 46 potholes were identified in the left lane and 44 were identified in the right lane. There were a number of regions where there were multiple consecutive potholes with small spacing (often less than a meter) between them, and these were labeled as a single pothole region. Due to the low quality of the roads, this resulted in pothole regions that were significantly longer than those used in the simulated data. Due to the number of required mappings between devices, the asynchronous clock nature between the devices, the 1 Hz sampling from the GPS device leading to poor spatial resolution, and the low quality of the road terrain, the ground-truth pothole locations were difficult to identify. The reported numbers are a best-effort determination.

5.2.4 Pothole Detection Architecture

The goals of our pothole detection system are two-fold. The first goal is to accurately and reliably detect potholes in multi-lane environments. This involves determining in which lane a pothole is in and determining its exact longitudinal location on the road. The second goal is to reduce the required bandwidth of the system by avoiding transmitting excessive sensor data from the vehicles to the Cloud, while still maintaining good pothole detection rates and localization accuracy.

To accomplish these goals, we design a *Filtered Multi-stage Detection System*, with the

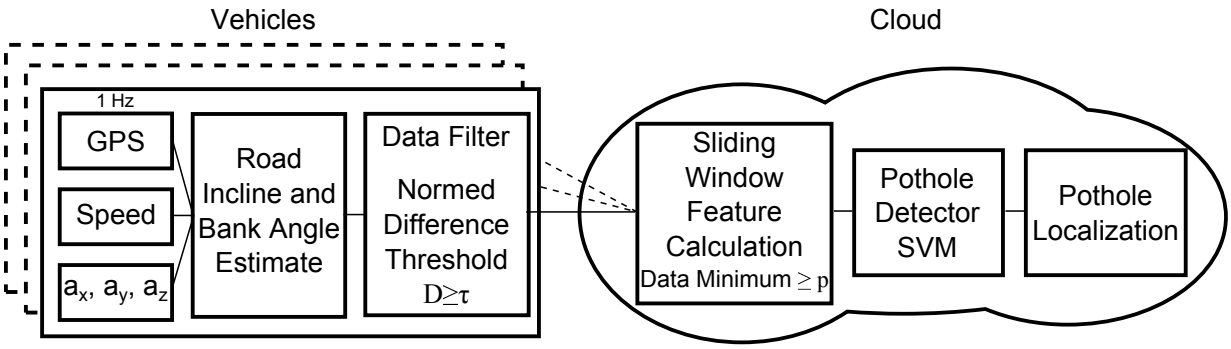


Figure 5.26: Architecture of the *Filtered Multi-stage Detection* Pothole system.

architecture shown in the block diagram in Figure 5.26. Following this architecture, the vehicles first locally acquire GPS, speed, and acceleration samples from their embedded sensors. If road bank and incline angle information is unavailable a priori, then the raw sensor data is used to estimate the road angles locally in each vehicle to determine which acceleration components correspond to normal road driving, as detailed in Section 5.1.3. Any acceleration components that differ from normal road conditions by a certain threshold, as explained in the filtering algorithm in Section 5.2.4.1, are then transmitted to the Cloud for aggregation.

Once the data is aggregated in the Cloud, a sliding window scheme groups the data to calculate features, as described in Section 5.2.4.2, with weightings for each data sample to account for multi-lane environments, as described in Section 4.2.3. After calculating the resulting features, the features are input into the trained classifier to determine if a pothole exists. For positive detections, the classifier output is used to localize the pothole longitudinally on the road.

5.2.4.1 Vehicle Data Filter

The in-vehicle data filter is designed to reduce the required bandwidth of the detection system by refraining from transmitting data to the Cloud that correspond to normal road conditions. The filter is constructed similarly to the in-vehicle road angle filter in Sec-

tion 5.1.3.1.

This filter is created by comparing the measured acceleration vector from the vehicle to an expected acceleration vector derived from gravity, centripetal acceleration, and the forward acceleration. If the observed acceleration vector differs from the expected, it is a possible indication that a pothole is present. This process also helps to dissociate large acceleration components that may mistakenly indicate a pothole by appropriately determining if those acceleration components should be expected at that location.

The design of this filter is an example of using side information, where environmental information obtained for other purposes can be used to augment another algorithm. If the road angles are not known a priori, for example as provided as side information by a continuous reconstruction of the road angle data as given by example results in Section 5.1.5, then the road angle estimation algorithm for individual samples, as detailed in Section 5.1.3, can be used instead to estimate the angles for each individual sample.

Once the angles are estimated or transmitted to the vehicle and the rotation matrix $\hat{\mathbf{R}}$ is constructed, a comparison can be made between the measured and expected acceleration vectors in the cost function, D, given by Equation 5.25. The normed difference between the vectors is compared against a threshold, τ, and only the data which satisfies the following inequality,

$$D \geq \tau, \tag{5.33}$$

is transmitted to the Cloud. Data in windows for which $D < \tau$ are not uploaded to the Cloud under the assumption that the data does not contain any discriminating information as it too closely resembles the expected normal road data. The data transmitted is only the acceleration components that hopefully correspond to the perturbation,

$$(\mathbf{a} - \mathbf{f}) - \hat{\mathbf{R}}\mathbf{w}. \tag{5.34}$$

The filter is essentially the opposite of the filter for road angle signal reconstruction in Section 5.1.3.1. For the road angles, we only want to transmit data where the rotated

acceleration vector and the measured one are in close agreement, whereas for potholes we look for acceleration vectors with a large difference as that could indicate a perturbation.

This filter is particularly useful for multi-lane detection. The most difficult part of multi-lane detection is having normal road data from adjacent lanes blend together with the data from the pothole region (and vice-versa) due to GPS error. This filter helps to systematically reduce the data blending problem since the normal road data from adjacent lanes is no longer uploaded to the Cloud and acceleration components related to expected normal driving conditions are removed.

5.2.4.2 Features

We first examine the effects that heterogeneous vehicles, potholes, and driving patterns have on the accelerometer signals. This is important for determining discriminating features for use as candidates in the detection models, and to design the models to be applicable under general conditions without tuning for specific situations.

We analyze the effect that vehicle speed has on the simulated vertical acceleration measurements (excluding gravity for these purposes). As an example, the 100 Hz vertical acceleration signals for the three different vehicle types driving over a 40 cm long, 4 cm deep pothole are shown in Figure 5.27a and Figure 5.27b for vehicles traveling at 25 km/h and 100 km/h, respectively. The vehicle's front and back tires each individually produce significant vertical acceleration peaks when driving over the pothole, with the distance between them dependent on the length of the vehicle. Since the sensor temporal output frequency is the same for the trials at each speed, there is less data available in a given distance at higher speeds since the traveling time is reduced. However, the response of the suspension system for the different speeds differs beyond just the spatial sampling rate since the impulses corresponding to the vehicle entering and exiting the pothole are closer in time at higher speeds. Also, the aftershocks are larger at a further distance from

the pothole at higher speeds since the vertical acceleration decays as a function of time, not distance. This emphasizes the importance of including speed as a component in the candidate features as it affects the acceleration signal properties.

The initial peaks from driving over the potholes are potential signature elements to design candidate features on for pothole detection. We therefore examine the upper and lower bounds of the vertical acceleration as a function of velocity. The results are shown in Figure 5.28 for normal road and pothole regions. When driving over a pothole, the upper bound on the acceleration increases until about 30 km/h and then decreases. The lower bound is approximately constant at all speeds. The bounds on the vertical acceleration on the normal road, however, symmetrically increase as the velocity increases. This opposing behavior between the normal road and pothole regions increases the difficulty of detecting potholes at high speeds, particularly when there is less available data. The bounds are also proportional to the size of the pothole and the vehicle center of mass height.

We perform a similar investigation with the lateral acceleration, as shown in Figure 5.29. A lateral acceleration response distinguishes potholes from features that span the width of the road, such as speedbumps or expansion joints. The upper and lower bounds on the lateral acceleration as a function of speed behave similarly to the vertical acceleration when driving over a pothole. There is an increase in the magnitude of the bound over the pothole until about 30 km/h, which is followed by a decrease. The lateral acceleration is much more stable than the vertical acceleration when driving on a normal road.

The bounds and range of the acceleration values, as well as their relation to velocity are therefore important discriminative features that could be used to detect the potholes. Candidate features are generated as functions of the acceleration and velocity measurements including weighted means, standard deviations, and maximum values of the raw aggregated inertial data. Absolute values as well as products and ratios of the accelerations with speed along the three coordinate axes are also calculated. The various combinations

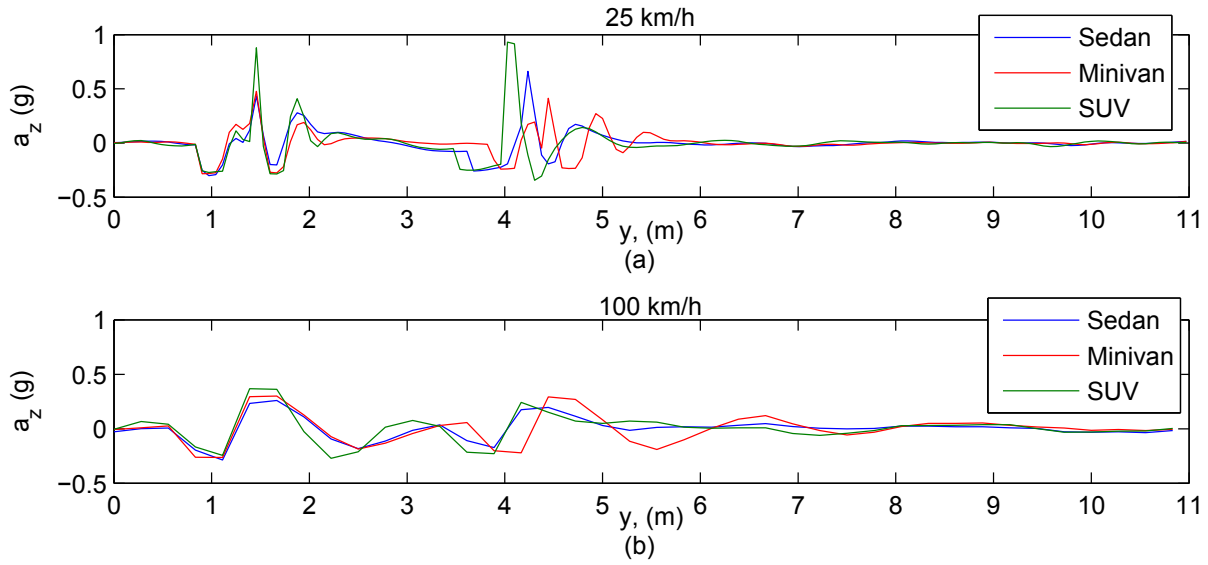


Figure 5.27: Vertical acceleration signals from vehicles driving over a 40 cm long, 4 cm deep pothole at (a) 25 km/h and (b) 100 km/h.

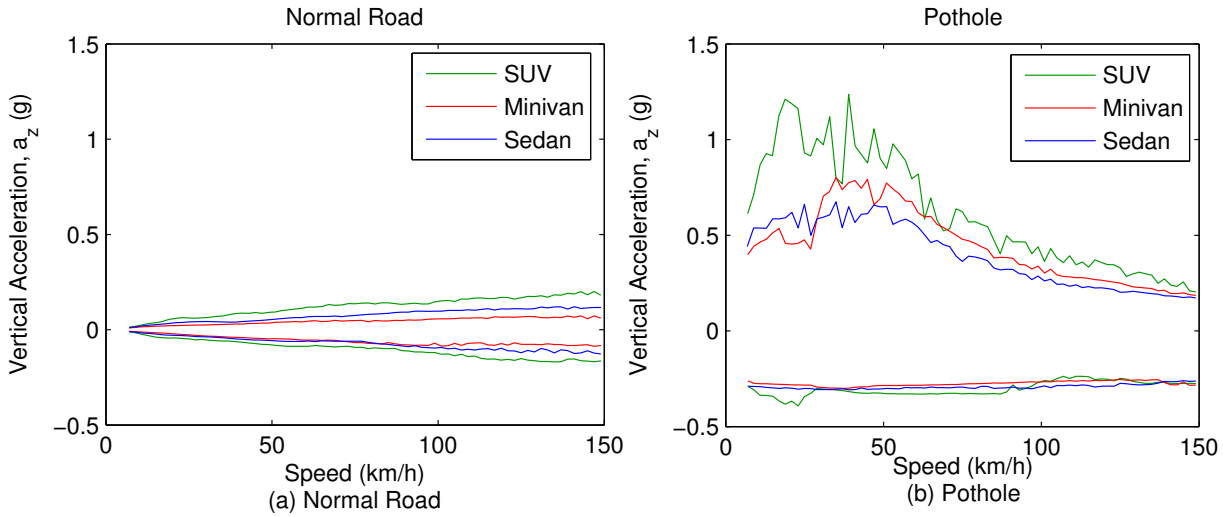


Figure 5.28: Upper and lower bounds of the vehicle vertical acceleration as a function of speed on (a) normal road, and (b) 40 cm long, 4 cm deep pothole.

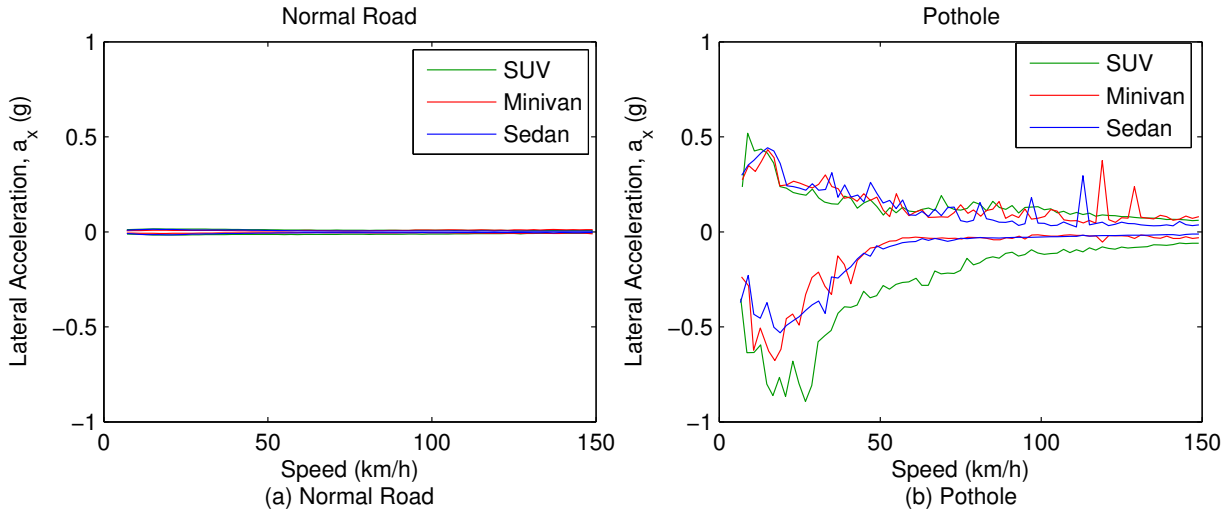


Figure 5.29: Upper and lower bounds of lateral acceleration on (a) normal road, and (b) 40 cm long, 4 cm deep pothole.

of these functions result in a list of 85 candidate features.

The low sampling frequency of the embedded sensors limits the nature of the candidate feature list. A vehicle traveling at 50 km/h would average only 0.72 samples in each 10 m window when using 1 Hz sampling. Therefore, many vehicle-specific features that would require multiple data samples per window, such as the number of zero-crossings, can not be defined. The window size would have to be significantly increased to consider these features, however that would result in the windows being orders of magnitude larger than the pothole region, which would encompass too much of the normal road to be able to identify potholes. Therefore the features are determined over the entire aggregated data set and are not defined specifically for data from any individual vehicles beyond using the weighting schemes for multi-lane or temporal issues, outlined in Sections 4.2.3 and 4.3 respectively.

The resulting features selected by the greedy forward feature selection algorithm are shown in Table 5.7. Note that there are features selected corresponding to each acceleration coordinate axis. As vehicles drive into potholes, they generally enter one tire at a time, resulting in a noticeable change in the accelerations along all three coordinate axes

Table 5.7: Selected SVM features from greedy forward selection algorithm

$\max(a_x)$	$\text{mean}\lvert a_y\rvert$	$\max(a_z)$	$\text{mean}(v)$
$\max\lvert a_x\rvert$	$\text{mean}\left\lvert\frac{a_y}{v}\right\rvert$	$\text{mean}\lvert a_z\rvert$	
$\text{std}(a_x v)$		$\text{std}(a_z v)$	
		$\max(a_z v)$	
		$\text{std}\lvert a_z\rvert$	
		$\text{mean}\lvert a_z v\rvert$	

as one corner of the vehicle dips lower than then rebounds compared to the rest of the vehicle. As expected from previous observations, speed has a major influence on the vehicle accelerations and speed-related features are therefore selected by the algorithm.

5.2.4.3 Cloud Data Filter

The features described in the previous section are not calculated for every window for data aggregated on the Cloud. When the in-vehicle data filter described in Section 5.2.4.1 is applied, only a fraction of the measured data is sent to the Cloud. In normal road regions, this results in only noisy or corrupt data being transmitted to the Cloud. This could lead to a significant number of false alarms since there would be no normal road data aggregated from other vehicles to balance out the extreme data. A second filter is therefore implemented in the Cloud before applying the pothole detector SVM to the aggregated data to mitigate this situation. We disallow positive pothole detections in any window that does contain a minimum number of data samples, p. This is based on a threshold fraction, η, of the expected number of data samples from the V vehicles if all the data was transmitted, where

$$p = \eta \frac{Vs}{v}. \tag{5.35}$$

The SVM is only applied in windows for which the sum of the weights of all the aggregated data samples in the window is greater than p.

5.2.5 Detection Results

The *Filtered Multi-stage Detection System*, as described in Section 5.2.4, is used to detect and localize potholes. In the following, we first analyze the efficacy of the in-vehicle data filter and its impact on the bandwidth requirements of the vehicle-Cloud network in Section 5.2.5.1. We then analyze the detection and localization accuracy for the corresponding tradeoffs in the number of required vehicles and the bandwidth requirements in the simulated data on multi-lane roads in Section 5.2.5.2 and single lane roads in Section 5.2.5.3. Results from testing real-world data are provided in Section 5.2.5.4. Temporal effects on the system are analyzed in Section 5.2.5.5. Finally the impact of the scope of the data set used for training the detection model is examined in Section 5.2.5.6.

5.2.5.1 In-vehicle Data Filtering

The filtering algorithm outlined in Section 5.2.4.1 uses the deviations from the expected acceleration measurements to both remove acceleration components unrelated to potholes and to prevent the transmission of data to the Cloud of any sample that likely corresponds to normal road conditions.

We first demonstrate the efficacy of the filter. Figure 5.30 shows the ground-truth bank angle signal from the simulated data set along with the estimated bank angles calculated from samples from 100 vehicles driving at an average of 50 km/h in an idealized situation with no added GPS or acceleration noise. The majority of the calculated sample bank angle values closely match the ground truth signal (the ground truth data overlaps the majority of the samples on the graph), indicating the calculated angles are correct. The samples that deviate from the ground truth are generally isolated to the pothole locations, where perturbations cause discrepancies between the expected and measured acceleration vectors. The angle values differ from the ground-truth since the algorithm attempts to rotate the expected acceleration vector from the inertial frame into a measured anomalous

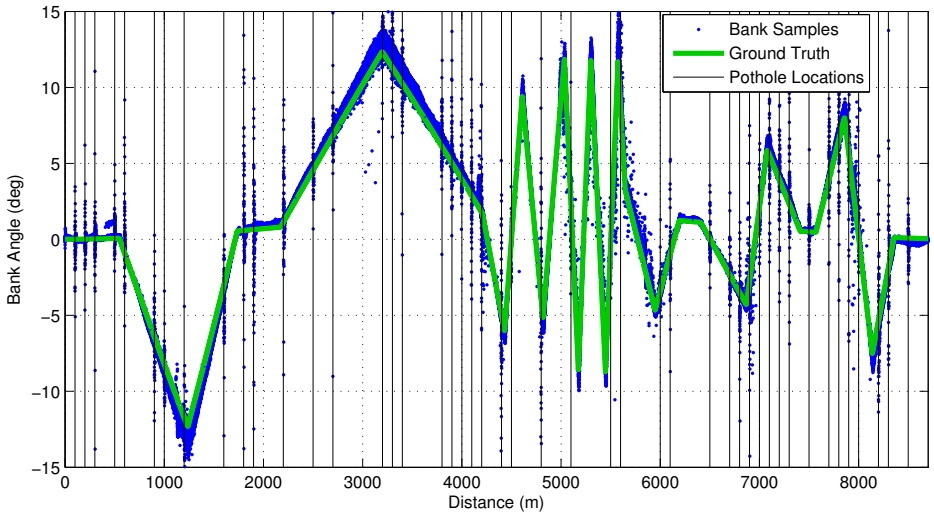


Figure 5.30: Road bank angle estimates from vehicle acceleration measurements in the presence of potholes.

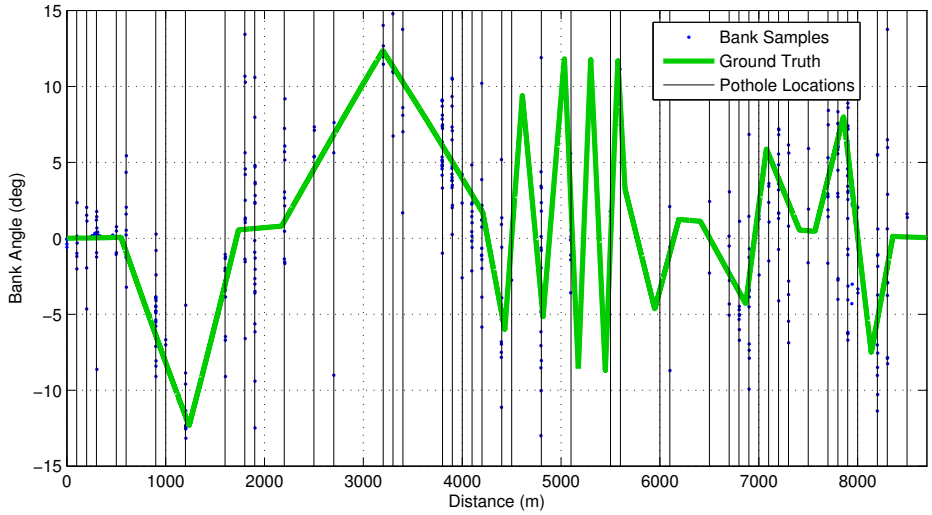


Figure 5.31: Road bank angle estimates from vehicle acceleration measurements which pass the threshold filter, $\tau = 2$ m/s^2.

vehicle state.

The discrepancies between the acceleration vectors are made more obvious when the vehicle data filter is applied with $\tau = 2$ m/s^2. The resulting measurements that pass the filter are shown in Figure 5.31. Here, all of the data which passes the filter correspond to pothole locations. Therefore this filter can be extremely effective in isolating the pothole data on each vehicle before transmission to the Cloud.

It should be noted that the situation isn't necessarily as clean as displayed in these figures. Sensor noise or other perturbations, such as speedbumps, could cause an acceleration measurement to still pass the vehicle filter. This is why crowdsourcing and the Cloud SVM detector are still necessary.

The transmission bandwidth is a function of the normed acceleration difference threshold, τ, being used in the vehicles to determine how much of the data is filtered out before transmission. For the example simulated road, with vehicles traveling at an average of 50 km/h, Figure 5.32 shows the resulting percentage of data that passes the filter for transmission. As the threshold initially increases above zero, the amount of data transmitted rapidly falls off as the suspected normal road data is held from transmission. The transmission percentage then levels off as only extreme suspected pothole data is able to pass the vehicle filter. The exact transmission value for a given threshold depends on the specific vehicle and road environment, however the remainder of the results will be in reference to our simulated course.

Assessing the value of decreasing the amount of data transmitted only makes sense in the context of the pothole detection rates. In the following section, we analyze the impact of reducing the amount of transmitted data and analyze the tradeoffs in the number of sampling vehicles on the pothole detection rate and localization values.

5.2.5.2 Simulated Multi-lane Detection Results

Due to the multi-lane environment we examine three categories of pothole detections:

1. *True Detection, Correct Lane* - Correctly detecting the pothole in the lane the respective pothole is residing.
2. *True Detection, Adjacent Lane* - Correctly detecting a pothole, however due to GPS error, detecting it in an adjacent lane from its true location. This is generally a result of GPS errors erroneously locating the discerning pothole data in an incorrect lane.

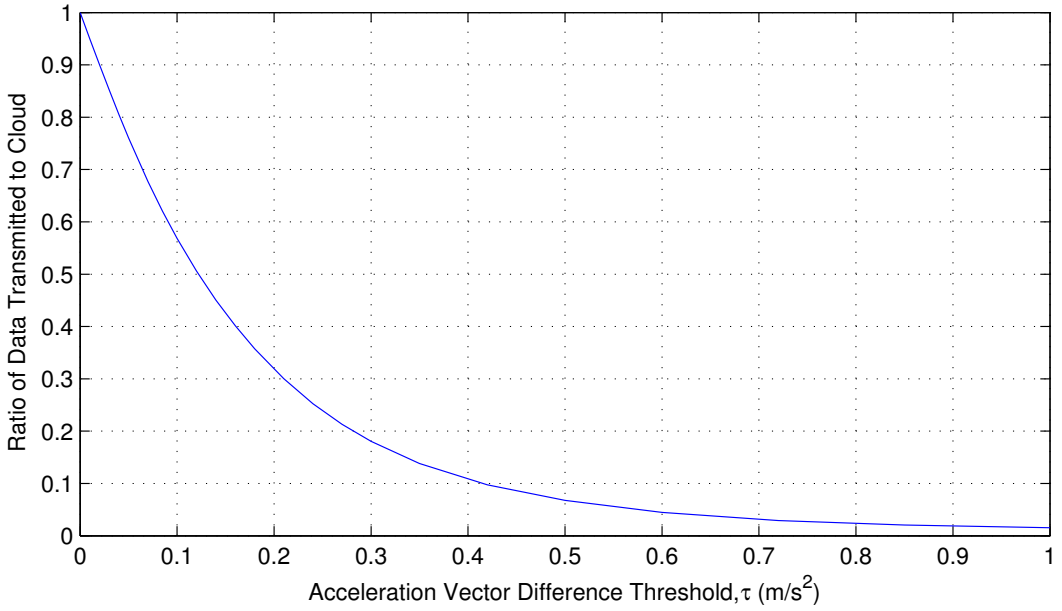


Figure 5.32: Percentage of data transmitted from vehicle to Cloud in *Filtered Multi-Stage Detection* as function of threshold τ for two-lane 50 km/h scenario.

Although these are technically incorrect detections, we classify them separately. For any application of the detection system requiring manual inspection, such as repairing the pothole, inspectors would be able to immediately see the correct pothole lane. This detection class does not apply to single-lane roads.

3. *False Alarms* - Positive detections where there is no pothole in the vicinity.

The detection rates for the correct and adjacent lane classifications are shown in Figure 5.33 and Figure 5.34 respectively as a function of the number of vehicles transmitting data and the ratio of data that passes the filter on each vehicle for the 50 km/h scenario. For any transmission ratio, the correct lane detection rate increases as the number of vehicles increases. Increasing the filtering threshold on the vehicle (and therefore decreasing the transmission ratio) initially improves detection as the normal road data is eliminated, however the detection rate then declines for high thresholds (low transmission ratios) as the increasing threshold eliminates discerning pothole data in addition to normal road data. The system performs best when only 30% of the vehicle data is transmitted. Therefore the

benefits of decreasing the amount of data transmitted are twofold:

1. The required bandwidth of the system is decreased as less data is transmitted.
2. The detection rate improves since there is less normal road data mixing in with the pothole data since the normal road data is not being transmitted.

Detections in adjacent lanes follow a similar pattern to the correct lane detection with respect to the vehicle filtering threshold, however the behavior differs with respect to the number of vehicles. The adjacent lane detection rate is initially high for a low number of vehicles, where a few vehicle traces with extreme GPS errors could wield a large influence. However as the number of vehicles involved in detection increases, the distribution of traces better clusters around their true lanes and the detector can better discern in which lane the pothole actually resides. For a large number of vehicles where the correct lane detection rate is highest, the adjacent lane detection rate is close to zero.

The false alarms per true pothole rate for the varying vehicle numbers and transmission ratios are shown in Figure 5.35. Similar to adjacent lane detections, the false alarm rate initially increases for a small number of vehicles, where any individual vehicle's corrupt or noisy measurements has a large influence, but then decreases once the number of vehicles increases the confidence in the group knowledge. The false alarm rate is low overall, particularly for the conditions where the detection rate is highest.

The ability to localize the pothole on the road, not just to a specific lane, but also by longitudinal distance on the road is important, particularly to any automated location-based control algorithm. The root-mean-square error (RMSE) of the localization accuracy for our proposed detection scheme is shown in Figure 5.36. The pattern is similar to the correct detection rate. The accuracy increases as the number of vehicles increases, reaching a minimum of about 2 m under tested values where the correct lane detection rate is highest. This is significantly better than the 10-15 m range of prior works dictated by the length of the sliding windows.

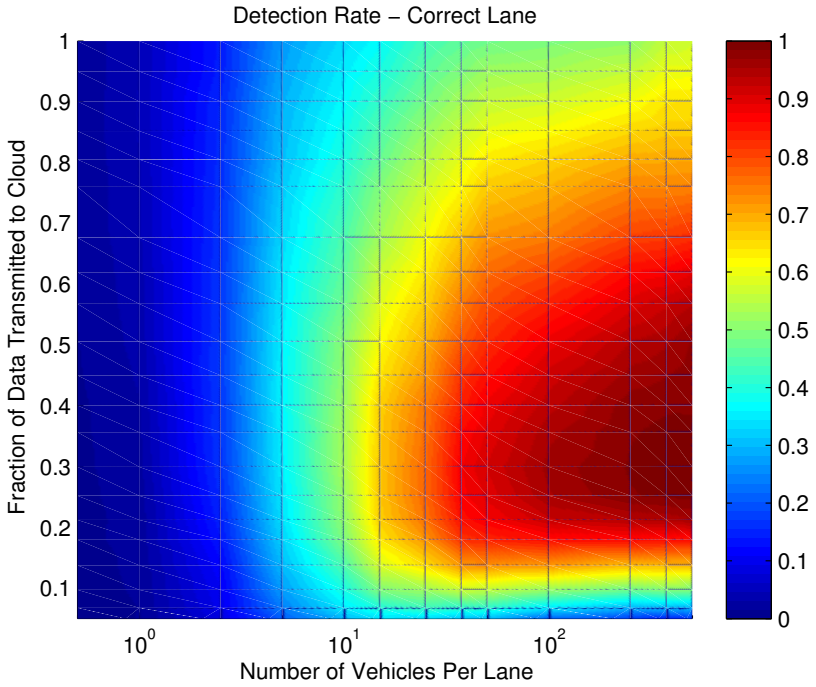


Figure 5.33: Detection rate for potholes on two-lane road in their correct lane - 50 km/h

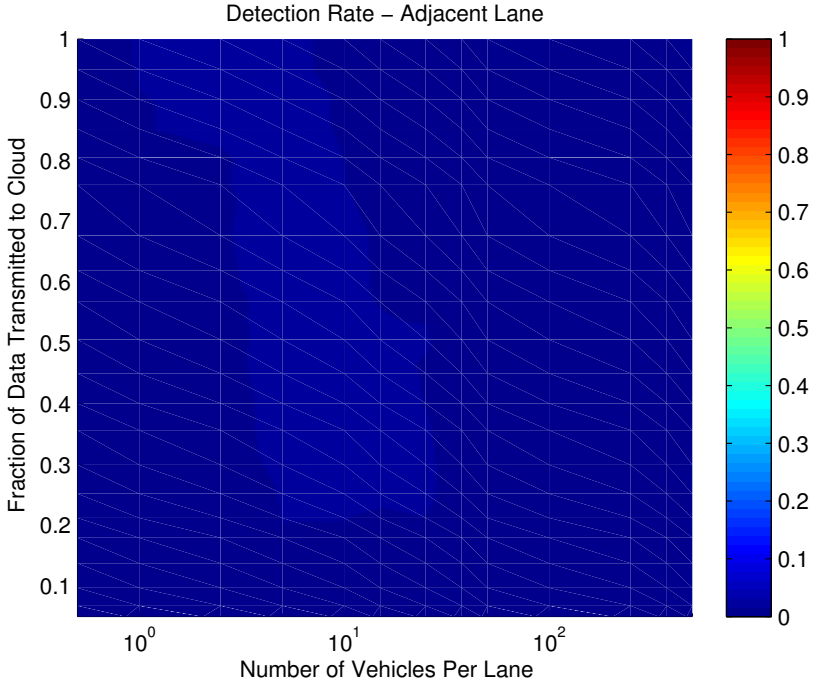


Figure 5.34: Detection rate for potholes on two-lane road in an adjacent lane to their true location - 50 km/h

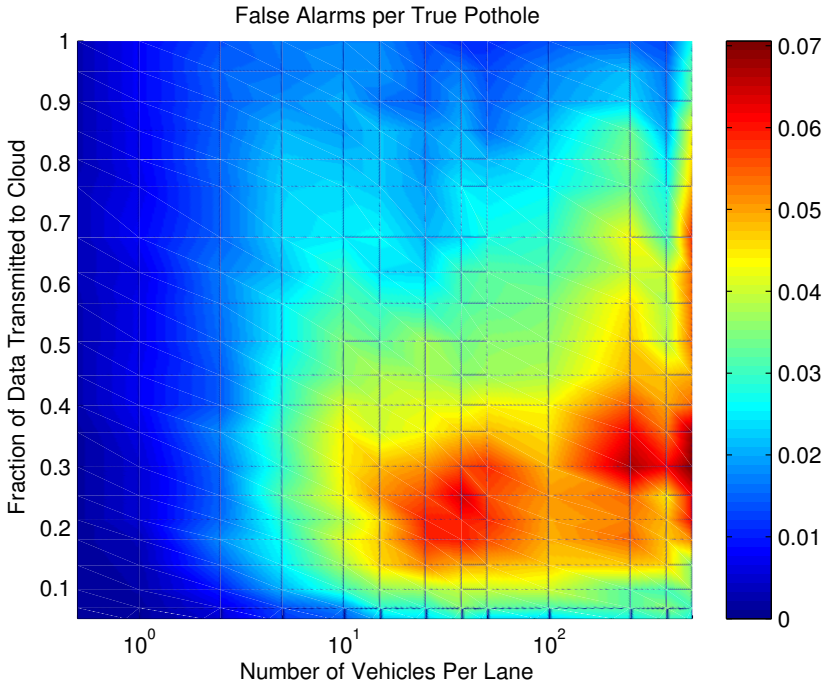


Figure 5.35: False Alarms per true pothole for two-lane pothole detection - 50 km/h

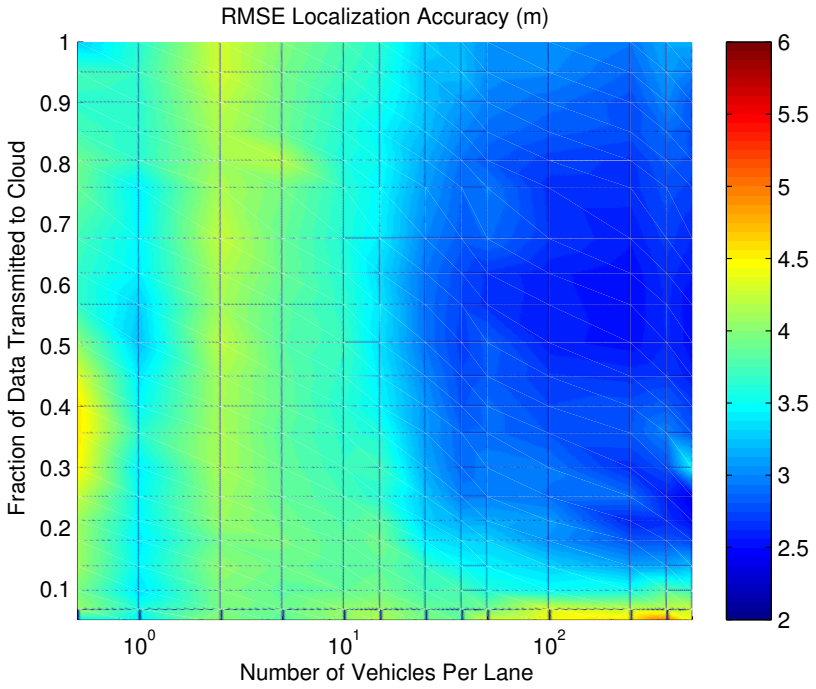


Figure 5.36: Root Mean Square Error on pothole localization distance on two-lane road - 50 km/h

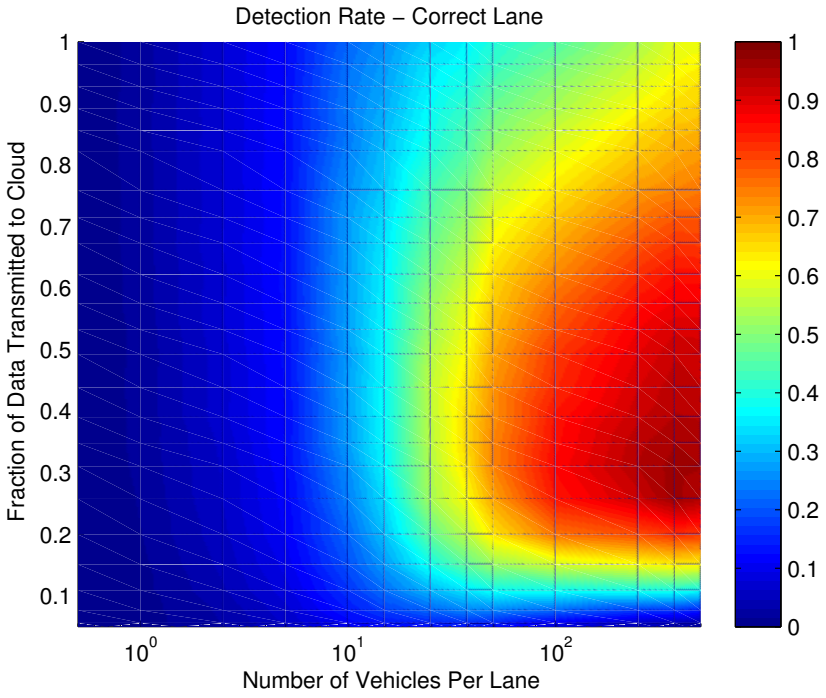


Figure 5.37: Detection rate for potholes in their correct lane on two-lane road - 100 km/h

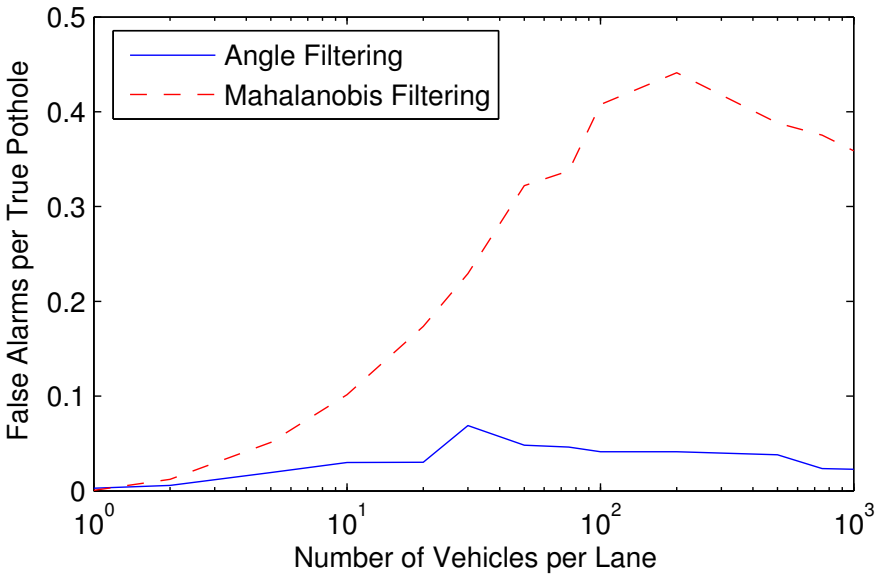


Figure 5.38: Comparison of false alarm rates for multi-lane pothole detection when transmitting 30% of the data when filtering by both the angle determination method of this work and the Mahalanobis distance threshold of previous works - 50 km/h.

The detection rates are also a function of the speed of the vehicles. Higher vehicle speeds increase the detection difficulty due to a number of factors. For one, since the sensors operate at a constant temporal frequency, the spatial sampling rate (the domain of interest) is decreased as the speed increases. Therefore less data is available for a given number of vehicle traces at higher speeds. Secondly, as the speed increases, normal road vehicle behavior starts to resemble pothole behavior [3], as demonstrated in Section 5.2.4.2. Vehicles tend to bounce more on normal roads at higher speeds, while they are also able to more easily glide over potholes. This somewhat inverted behavior means that the calculated detection features are less discerning at higher speeds.

This contrasting speed behavior is shown for the correct lane detection rate in Figure 5.37 for simulated vehicles driving on the course at an average of 100 km/h. Note that it requires many more vehicles to achieve similar results than what is shown for the 50 km/h scenario in Figure 5.33. False alarms and localization results are also similarly degraded.

We also compare this in-vehicle filtering algorithm to one from previous work [3], which uses the comparison of the Mahalanobis distance of the accelerometer features to a threshold. Although the correct lane detections are only slightly improved compared to the Mahalanobis scheme, the false alarm rate is significantly improved for our system, particularly for the multi-lane environment as shown in Figure 5.38. The Mahalanobis distance threshold allows through samples with large acceleration components, even if those components are to be expected from normal driving. Given the winding nature of our simulated course, this leads to many false positives.

These results were for multi-lane weights calculated using the autocorrelated weighting model where all the samples from a single vehicle are given equal weight. However, this involves estimating additional parameters of the covariance matrix, which we assumed was known for these results. For comparison, the detection results for the multi-lane road

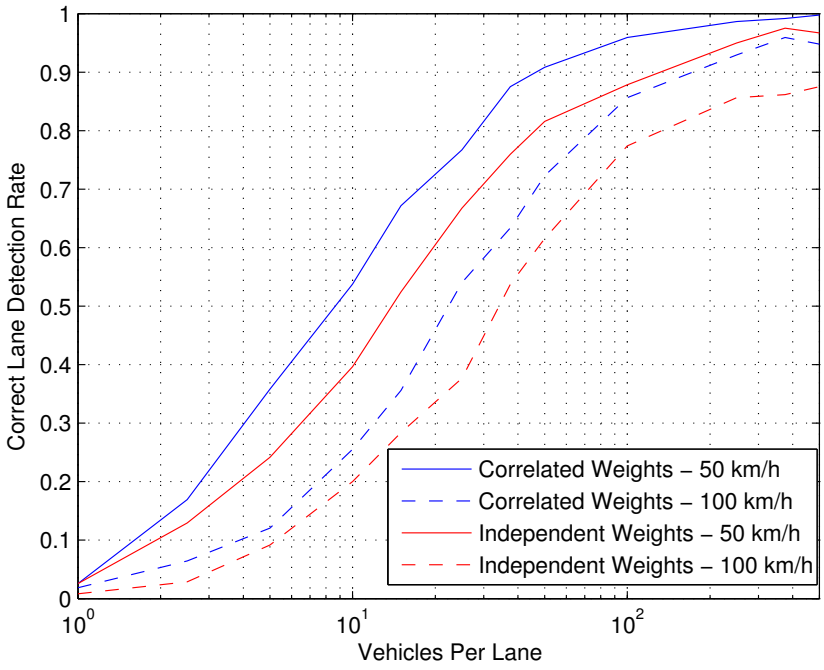


Figure 5.39: Comparison of correct lane detection results for using autocorrelated multi-lane weights per vehicle against independent weights for each data sample

for using independent weights for each data sample are shown in Figure 5.39 for vehicles traveling at an average of 50 km/h and 100 km/h. The detection results for the correlated weights are better than those for the independent weights as information from the entire group is used to enhance the weight determination process. There is also a significant improvement for the adjacent lane detection rate when using the autocorrelated weights, as shown in Figure 5.40.

5.2.5.3 Simulated Data Single Lane Detection Comparison

We compare the detection results of the previous section to those obtained from crowd-sourcing vehicles from a single lane environment. The detection rate and false alarm per true pothole rate are shown in Figure 5.41 and Figure 5.42 respectively, and the RMS localization error is shown in Figure 5.43 for an average vehicle speed of 50 km/h. The results are significantly improved over the multi-lane scenario as there is no longer normal

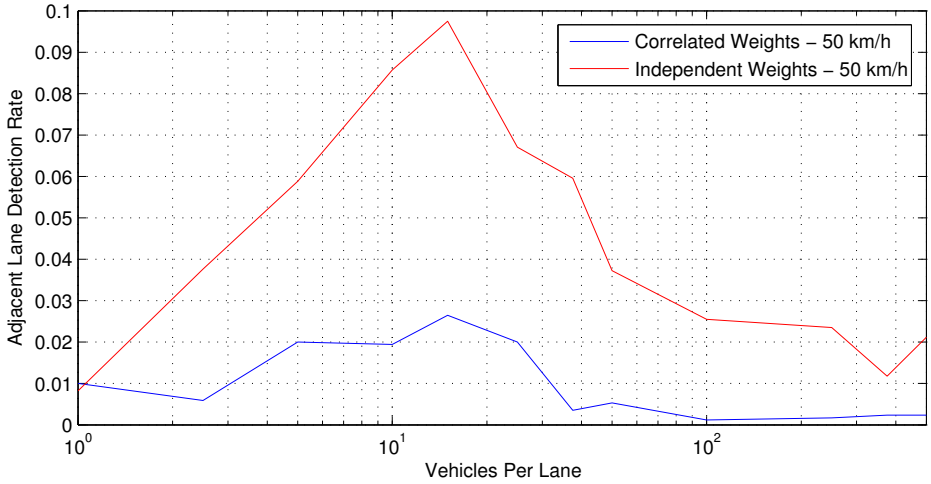


Figure 5.40: Comparison of adjacent lane detection results for using autocorrelated multi-lane weights per vehicle against independent weights for each data sample

road data from adjacent lanes being mixed in with the pothole data. Significantly fewer vehicles are required with only a single lane to obtain similar detection rates.

A comparison of the multi-lane and single lane detection rates for 30% of data passing the in-vehicle filter for both 50 km/h and 100 km/h roads is shown in Figure 5.44. At 50 km/h it only takes about 20 vehicles to reach a 90% detection rate for a single-lane road. It takes over 100 vehicles per lane to reach the same detection threshold for a two-lane road. The results would continue to degrade as the number of lanes increases and there is more normal road data from adjacent lanes to obfuscate a pothole. However on most city roads it does not take a significant amount of time to obtain such vehicle numbers.

5.2.5.4 Real-world Data

We applied the SVMs trained from the simulated data to the real-world data outlined in Section 5.2.3.2. Overall, the detectors derived from simulation perform adequately when applied to real-world data and compare reasonably well to the simulated results given the varied conditions.

The number of detections for each classification type for the real-world data is shown in

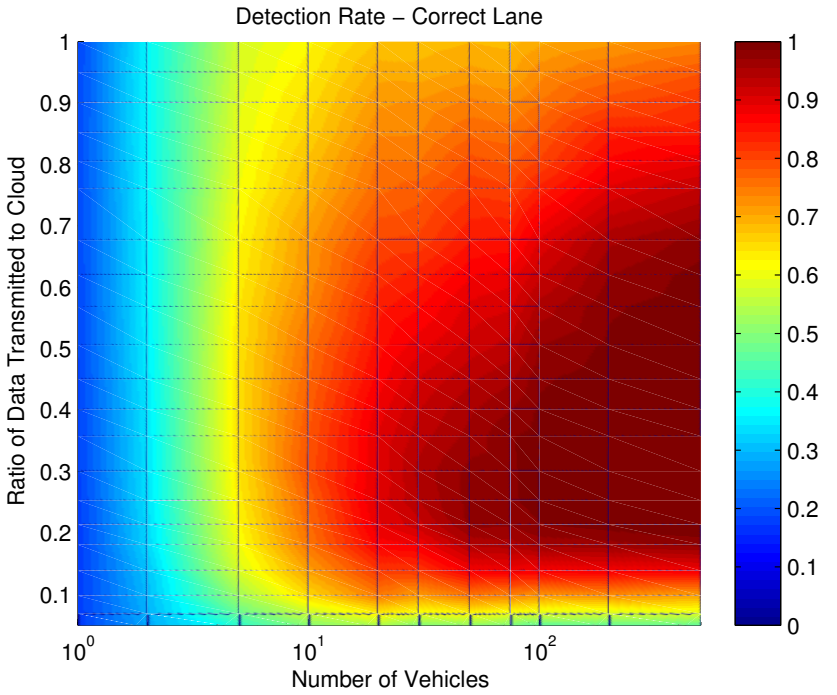


Figure 5.41: Detection rate for potholes in single-lane scenario - 50 km/h

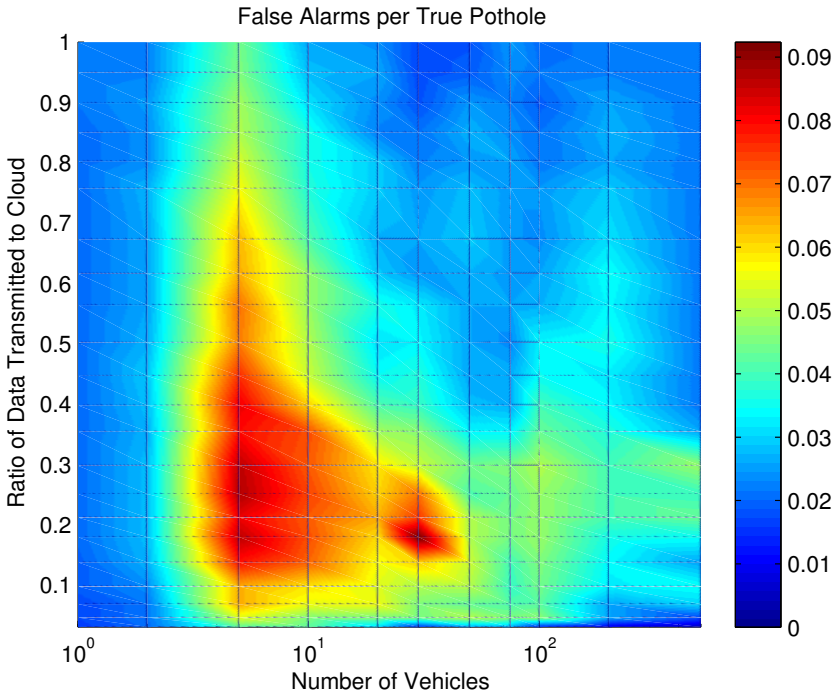


Figure 5.42: False alarm rate per true pothole for single-lane scenario - 50 km/h

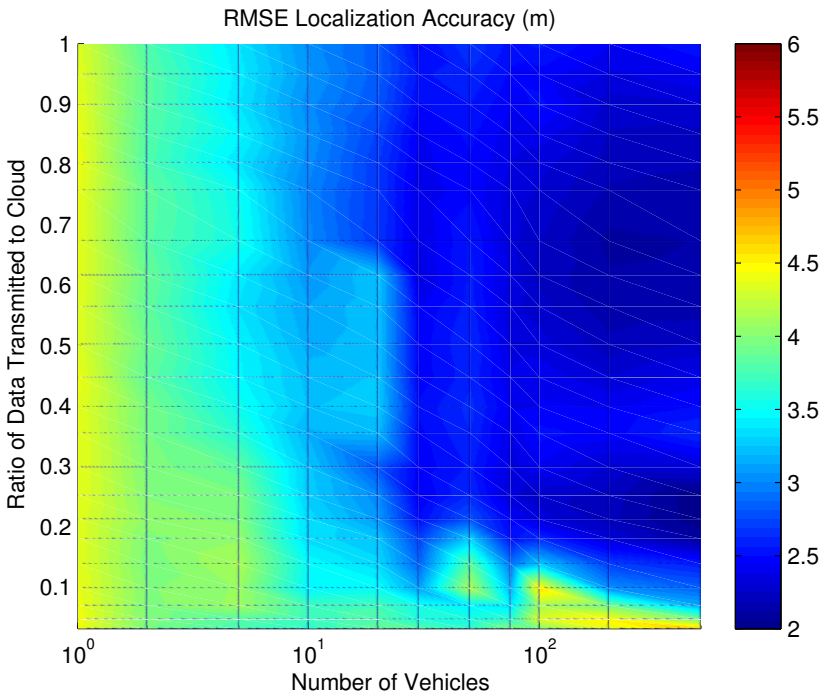


Figure 5.43: Root Mean Square Error on pothole localization distance for single-lane scenario - 50 km/h

Table 5.8. Overall, 60% of the potholes were detected in either the correct (51%) or adjacent (9%) lanes. The detection rates from the two-lane simulation results for independent weights for 8 vehicles per lane at an average of 50 km/h are 35% detection in the correct lane, 8% in the adjacent lane, and 0.04 false alarms per true pothole. The correct and adjacent lane detection results are comparable, although slightly higher than what would be expected for this many vehicles given the results from the simulated data. This is likely due to the increased sampling frequency from the lateral acceleration sensor and from the lengthier pothole regions.

The false alarm rate is higher than expected though. This is due partially to the extremely rough and broken paving of the road, and the slightly higher speeds than 50 km/h. It should also be noted that three of the false alarms correspond to manhole covers.

We also demonstrate how the detection rate increases as a function of the number of vehicles transmitting their data. By using only a subset of the traces from the real-world

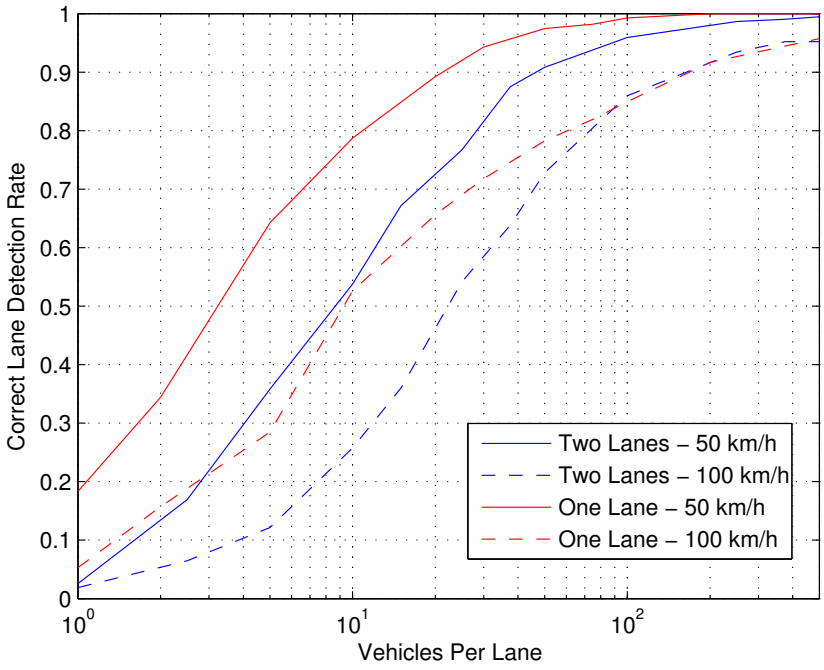


Figure 5.44: Comparison of correct lane detection rates for single and multi-lane environments at different speeds when 30% of the data passes the in-vehicle data filter and is transmitted to the Cloud.

data set, Figure 5.45 shows the detection rates for aggregating data from a varied number of vehicles per lane. The overall increasing detection rate trend is similar to that from the simulated data and we expect the detection rate to continue to increase as data from more vehicles is collected.

5.2.5.5 Temporal Effects

The impact of the temporal weights, defined in Section 4.3, are tested by analyzing how the detection system responds to dynamically changing pothole conditions.

We rerun the simulated environment by aggregating samples from 40 vehicles per time unit (with average speeds of 50 km/h), and running the detection system after every time unit. After 25 time units, the potholes were simulated to be repaired, i.e., data after this time threshold was drawn from simulations without potholes. The time decay constant

Table 5.8: Pothole detection results for real-world multi-lane data

Road	True Potholes	Correct Lane Detection	Adjacent Lane Detection	False Alarms
1	18	6	1	3
2	28	14	4	7
3	31	14	3	6
4	13	12	0	3
Total	90	46	8	19

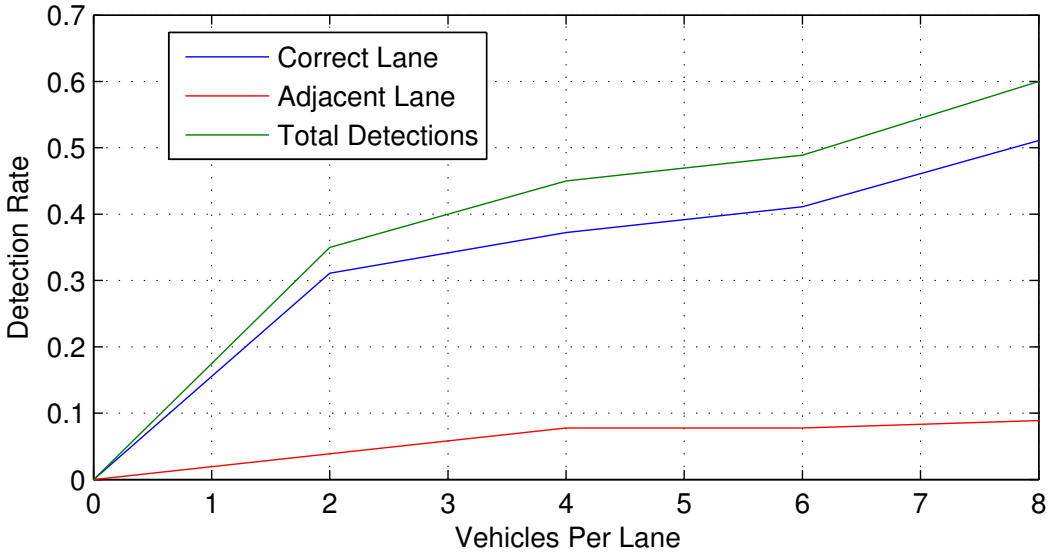


Figure 5.45: Real-world detection rates as a function of the number of vehicles per lane providing data.

for weighting the data samples was set to $\tau_T = 6$ time units. The threshold for the in-vehicle filter allowed for approximately 30% of the data to be transmitted when the potholes existed. The detection results for both the correct and adjacent lanes are shown in Figure 5.46. Note that after $t = 25$, the detection rates are still shown using the correct and adjacent lane detection terminology. After this threshold time, all the detections would technically be false alarms, however continuing to show the detection results in terms of where the potholes previously were provides appropriate context. This works well since no new potholes were added.

Also shown in Figure 5.46 is the temporally weighted number of vehicles used in detection. The value for this experiment reaches a stable value at approximately 124 vehicles per lane. This is a function of the number of vehicles added per time unit and the decay rate of the weightings. Other temporal rates would result in a different stable value (or fluctuating if the rates fluctuated), however the stable rules were used for this experiment so that the detection rates would stabilize, effectively removing the number of added or removed vehicles as a variable, since everything can be evaluated in the context of the stable detection rate. Note that for this scenario, the detection rate reaches a stable value of about 0.89, which is just slightly below the rate of 0.95 shown in Figure 5.44 for 124 vehicles per lane. The detection rate declines fairly rapidly after the potholes are removed, with no more detections in the pothole regions by $t = 32$. This quick dropoff in detections is due to the relatively large rate of decay of the weights from the given time constant. A sample collected at $t = 24$ (the last time instant the potholes existed) would only be weighted at a factor of 0.26 when running the detection process at $t = 32$.

The opposite of this scenario is detecting potholes where none existed before (and data was already aggregated indicating as such). The simulation settings were set the same as for the previous experiment, however there were no potholes initially and at $t = 25$ potholes were created at the 85 simulated locations. Detection results are shown in Figure 5.47. These results are similar to those for the pothole being repaired in that detections begin to increase when the potholes exist. Note that the detection rate increase is slower than that shown in Figure 5.46 where the potholes exist at $t = 0$. This is because in the second scenario when the potholes begin to be detected there remains data from previous times when there was no pothole incorrectly influencing the detection result. The system has to wait for the non-pothole information to decay out of the system using the temporal weighting scheme before the detection rate can reach the target level.

In general, the aggregated data from a single vehicle would not be indicated by a single

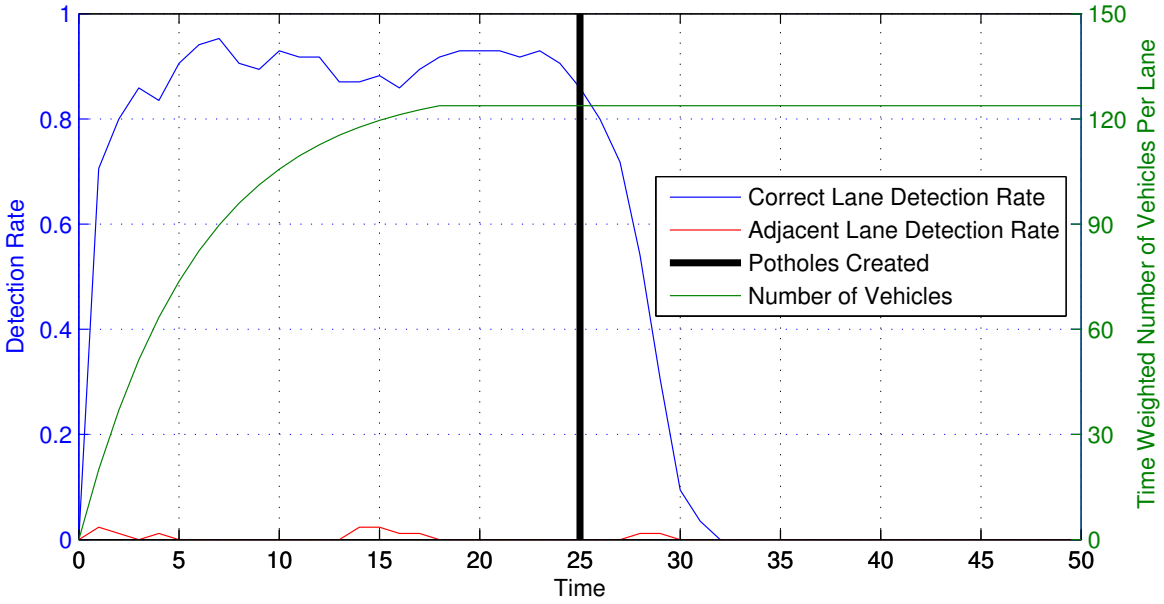


Figure 5.46: Detection results for simulated experiment where potholes are repaired at $t = 25$.

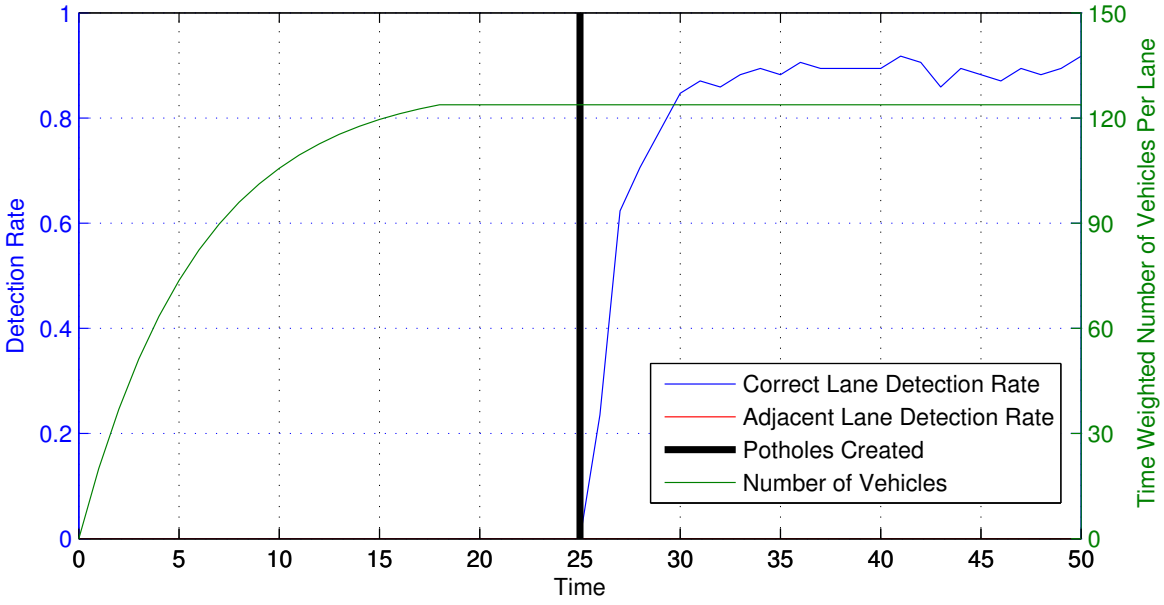


Figure 5.47: Detection results for simulated experiment where potholes are created at $t = 25$.

timestamp, but instead be a function of the speed of the vehicle along the road. However these results still illustrate the larger point of how the detection system is able to adapt to temporally changing conditions.

5.2.5.6 Training Data

The *Filtered Multi-stage Pothole Detector* is designed to isolate the acceleration components on the vehicle that correspond to the pothole interaction, so that the aggregated data set on the Cloud consists only of acceleration components unrelated to normal road driving. This process allows for the creation of a generic detection model that is applicable to diverse environments. Generic models are important as tuning embedded sensors or detection parameters for specific situations may be extremely difficult.

To further demonstrate the importance of using extensive data, we simulated vehicles driving on four additional road courses, where instead of a continually changing road environment, as in the road course described in Section 5.2.3.1, each of these roads has only a single type of non-pothole environmental feature. Details of these four road courses are provided in Table 5.9. The road roughness and types of vehicles used were similar to previous simulations.

To demonstrate the importance of using extensive data from a wide variety of circumstances and to try and isolate the anomalous acceleration components, we trained new SVMs by using the simulated raw acceleration data from the vehicles on these four road courses. Five different SVMs were trained: four from training using data from each road separately, and one using data aggregated from all four courses. These SVMs were then used to test and detect potholes on each of the different courses. The detection rate and false alarms per true pothole are shown in Table 5.10 and Table 5.11, respectively, for each training and test combination of the road courses with comparisons to training the SVM using all roads. These results are for vehicles traveling at an average of 50 km/h, using

Table 5.9: Description of simulated road courses to compare for model training.

Course	Road Style	Road Features (100 m spacing)
1	0.4 km long Straight, Flat	100 cm long, 8 cm deep pothole 40 cm long, 3.5 cm tall speedbump 120 cm long, 3.5 cm tall speedbump
2	3.1 km long Straight, Flat	30 potholes 2 cm - 10 cm deep 20 cm - 120 cm long
3	3.1 km long Straight, 5° incline	30 potholes 2 cm - 10 cm deep 20 cm - 120 cm long
4	3.1 km long Sinusoidally curving, Flat	30 potholes 2 cm - 10 cm deep 20 cm - 120 cm long

data aggregated from 100 vehicles.

Depending on how different the environmental conditions of the courses are, the classifiers trained on only one course are inaccurate when applied to other courses. The speedbumps on Course 1, for example, are continually detected as potholes by the other SVMs and the curves on Course 4 mask the potholes from the classifiers trained on the straight-line courses. The SVM boundaries are calculated to maximize the margin on the training course without any consideration as to their impact on features from other courses. The SVM trained on data from all the courses does not always beat the SVM trained on individual courses when tested on those respective courses likely due to overfitting, but overall is better able to handle diverse road conditions.

These results reinforce the need to capture training data representing as many diverse scenarios as possible to create a general model. Simulations are used as the size and detail of the data set would be too costly to empirically collect.

Table 5.10: True detection rates for pothole detection on different road courses (see Table 5.9) when trained on individual courses

		Testing Course			
		1	**2**	**3**	**4**
Training Course	**1**	1.00	0.99	1.00	0.42
	2	1.00	0.99	0.97	0.35
	3	1.00	1.00	1.00	0.43
	4	1.00	0.85	0.87	0.95
	All	1.00	1.00	0.99	0.98

Table 5.11: False alarms per true pothole for pothole detection on different road courses (see Table 5.9) when trained on individual courses

		Testing Course			
		1	**2**	**3**	**4**
Training Course	**1**	0.20	0.01	0.03	0.23
	2	2.00	0.04	0.04	0.05
	3	2.60	0.14	0.05	0.04
	4	2.00	0.04	0.01	0.007
	All	1.00	0.05	0.01	0.03

5.3 Chapter Summary

In this thesis chapter, we demonstrated how the vehicle-Cloud information extraction framework could be applied to determine road incline and bank angles, and potholes from undersampled and asynchronous vehicle sensor data. Algorithms were developed for each of the event types, then tested on both simulated and real-world data. Vehicle filtering algorithms were presented to limit the required network bandwidth of the detection system. It was shown that systematically constructed data filters can not only limit the required bandwidth, but also improve detection results. It was shown how using the MSVR detection algorithms presented in Chapter 4 and the GPS location error models in Chapter 3

were critical to enhancing the detection process. We also demonstrated the benefits of an extensive training data set and how the algorithms are able to adapt to temporally changing information.

Chapter 6

Conclusions and Future Work

6.1 Thesis Summary

In this thesis, we presented a framework with examples to extract environmental road information from asynchronous and undersampled vehicle sensor data. The framework for the interconnected vehicle-Cloud detection system was presented in Chapter 2. Developments and limitations of vehicle sensing technology were presented that both necessitated and made feasible the aggregation of data from multiple vehicles onto the Cloud in order to satisfactorily detect road environmental information. Various components of the detection framework were introduced, including the sensors, vehicle and Cloud data filters, event type descriptions, information feedback, and parameter optimization. The importance of each of these components and their relationships within the framework and to example information detection algorithms were described.

Chapter 3 introduced the unique sampling conditions that arise when using embedded sensors from multiple vehicles as a mobile sensor network to detect environmental road information. These sampling conditions are referred to as *multi-source, variable-rate* (MSVR) sampling conditions. Sampling error models were introduced, specifically with regards to GPS location error, which is used to spatially index the measurements from

all other sensors. Different frameworks for the error models were introduced depending on various assumptions and how the error model was to be applied to any detection algorithm.

Chapter 4 introduced detailed algorithms to specifically detect both binary and continuous types of road information, based on the sampling conditions presented in Chapter 3 as applied within the context of the framework of Chapter 2. The signal reconstruction algorithms were analyzed for various sampling conditions. Adaptations for time-varying information, improved localization accuracy, and optimizing the detection parameters are presented with respect to the different detection algorithms.

Chapter 5 applied the detection algorithms of Chapter 4 to two specific types of event detection, those being potholes, and road incline and bank angles. Specific detection algorithms were derived for each of the events, and results were presented for both simulated and real-world data. These results helped to validate both the specific event detection algorithms and the application of the aggregated data detection algorithms within the overall information extraction framework.

This final chapter will summarize the contributions of the thesis in Section 6.2 and present directions for future related work in Section 6.3.

6.2 Thesis Contributions

The main contributions of the thesis are summarized as follows:

- **Information Extraction Framework:** We developed a framework to extract road environmental information from asynchronous and undersampled vehicle sensor data. The framework is developed for information with varying spatial and temporal granularities. Data filters and the use of side information are introduced to both reduce the required bandwidth of the system and to improve the detection results. We developed algorithms to estimate the parameters of the detection algorithms based on the data, including temporal information decay parameters and signal reconstruction

bandwidth. We analyzed the tradeoffs for the number of sampling vehicles compared to the detection accuracy for the various algorithms and applications presented.

- **Multi-source, Variable-rate Sampling:** We introduced the concept of multi-source variable-rate (MSVR) sampling, specifically within the context of vehicle sampling conditions. Samples obtained by vehicles are indexed by their GPS locations which have spatially and temporally correlated error models, as detailed in Chapter 3. We derived signal reconstruction models that specifically account for these types of error models. The MSVR framework was extended to joint signal reconstructions, with adaptations for temporally changing information, and optimizations presented for the algorithmic parameters. Applications of the MSVR models were also introduced for binary event detection, such as using them to determine a multi-lane weighting scheme to distinguish samples obtained from each lane.
- **Road Incline and Bank Angle Estimate:** We developed an algorithm that estimates the road incline and bank angles from individual acceleration and GPS measurements. By crowdsourcing data from multiple vehicles and using the derived signal reconstruction algorithms on these MSVR samples, we were able to reconstruct the road incline and bank angle signals for entire roads.
- **Pothole Detection:** We applied the information extract framework to detect potholes from vehicle sensor data. This was done by crowdsourcing the MSVR sampled data to create a detector on the Cloud. To the best of our knowledge, this is the first work that has performed detection on multi-lane roads with an attempt to finely localize the potholes given the GPS location noise. Road angle information was further used to filter data to both reduce the required network bandwidth of the detection system and to enhance the detection results.

6.3 Future Work

In this section, we present four possible directions for future work. Section 6.3.1 presents additional possible detection process parameters to optimize. Section 6.3.2 presents an introduction and examples of multi-dimensional MSVR signal reconstructions, specifically as applied to vehicle sampling conditions. Section 6.3.3 introduces reconstructing continuous signals from only binary data. Section 6.3.4 provides an example of using the time decay algorithm for continuous reconstructions to estimate a time-varying temperature signal.

6.3.1 Parameter Optimization

There are number of parameters presented in this work for which a more generalizable tuning procedure could be implemented. One such example is the in-vehicle data filtering threshold presented in Section 5.2.4.1 for pothole detection. Although results in Section 5.2.5 identified the best performing threshold, the determined value was generalized across our simulation conditions. Due to the age and quality of the vehicle or sensor system, different vehicles will perform differently with respect to the detection and filtering algorithms and should therefore have different filtering thresholds associated with them. A feedback algorithm could be implemented to attempt to determine the impact of any given vehicle's measurements on the performance of the detection system, which could be used to reinforce the detection capabilities of individual vehicles. The reinforcement could involve adjusting filtering thresholds for each vehicle. Additionally, a performance weighting scheme could be implemented, similar to the temporally dynamic feature algorithm presented in Section 4.3, to further enhance the feature calculation process.

Furthermore, the data transmission rate from the vehicles to the Cloud could be regulated as a function of the detection rate or detection confidence level on the Cloud. For example, for a static signal such as the road incline angle, once the signal reconstruction variance is reduced below a desired threshold, there is no need to continue to aggregate

additional data as it would not provide any additional information. For a signal exhibiting more variation in time, the transmission rate from the vehicle could still be regulated such that transmission rates increase during periods the signal is believed to be more dynamic and decrease when the signal is relatively slowly changing. Such a procedure could be used in conjunction with a temporal weighting algorithm to reduce the required network bandwidth in addition to improving the accuracy of the information extraction process.

6.3.2 Multi-dimensional MSVR Signal Reconstructions

Beyond reconstructing one-dimensional signals as functions of the distance along a road, vehicle sensor data can be used to reconstruct multi-dimensional functions of environmental information. By aggregating data from vehicles traveling on multiple roads, larger maps can be developed containing information such as a temperature distribution or terrain topography, inferred in spaces between the roads where no vehicles traveled. The signal reconstruction algorithms presented in Section 4.1 could therefore be extended to multiple dimensions for these signal reconstruction purposes. One example is the nonuniform image reconstruction algorithm with no sample location noise in [99].

The same MSVR challenges highlighted in Chapter 3 extend to the two-dimensional situation, however one must also account for the correlations between the multiple spatial dimensions, for example as described by the GDOP covariance matrix in Equation 3.1. There is also an additional signal reconstruction challenge in two dimensions when using vehicular-based sensors. It was shown in Section 4.1.2.5 that a Voronoi weighting scheme provided good results for MSVR sampled signal reconstructions. This weighting scheme is also applicable two two-dimensional signals. However, using the area of the Voronoi regions as weights is not as logically transferable to the domain of roadway sampling. Unlike in the one-dimensional road case, the sample locations for a two-dimensional network of roads never approximates a uniform distribution over the domain. When aggregating samples

from vehicles on a road, there is a relatively small spacing between samples along the tangential direction to the road. However, there is a large spacing in the sample distance between different parallel roads in the regions where there are no roads from which to obtain samples. Using a Voronoi weighting scheme, samples that are obtained along the edge of the roadway (due to true locations or noise) have a disproportionately large weight for the reconstruction, even though the samples may not be in any more of a sparse region along the road direction than any other sample.

To demonstrate the weighting behavior, Figure 6.1 shows the Voronoi diagram (where the area of each region represents the weight of each sample) for two sampling conditions. In the first scenario, samples are drawn from a uniform distribution over the entire domain. In the second scenario, samples are obtained from a uniform distribution over the indicated roads, with i.i.d. sample location error (zero-mean, $\sigma_x = \sigma_y = 0.1$) added. The figures were constructed with 2000 samples in each scenario. Observe that by using Voronoi weights, the samples in the road sampling case that are obtained at the largest perpendicular distance to each road have a significantly larger weight (larger area) in the reconstruction scheme than other samples, and the samples in those respective regions are usually closer to the borders of the Voronoi regions than the centers.

Using these sampling schemes, we attempt to reconstruct the following function of two variables,

$$z = \sin(2\pi(0.189)y)\sin(2\pi(0.132695)x + 2.1) + 0.05(y-10)^2, \tag{6.1}$$

over the domain $x, y \in [0, 20]$. The ground-truth image is shown in Figure 6.2. The road sampling method using Voronoi weights, as shown in Figure 6.1, does not produce a viable reconstruction for 2000 samples. However, we repeat the simulation by adding approximately three times as many roads to sample over, thus decreasing the sparsity of many of the regions of the domain. Reconstructions (bandlimit M_x=M_y=0.4) for each of the sampling schemes are shown in Figure 6.3. The RMSE for the reconstructions are 0.45 for the uniform sampling and 0.77 for the road sampling. This is a fairly substantial

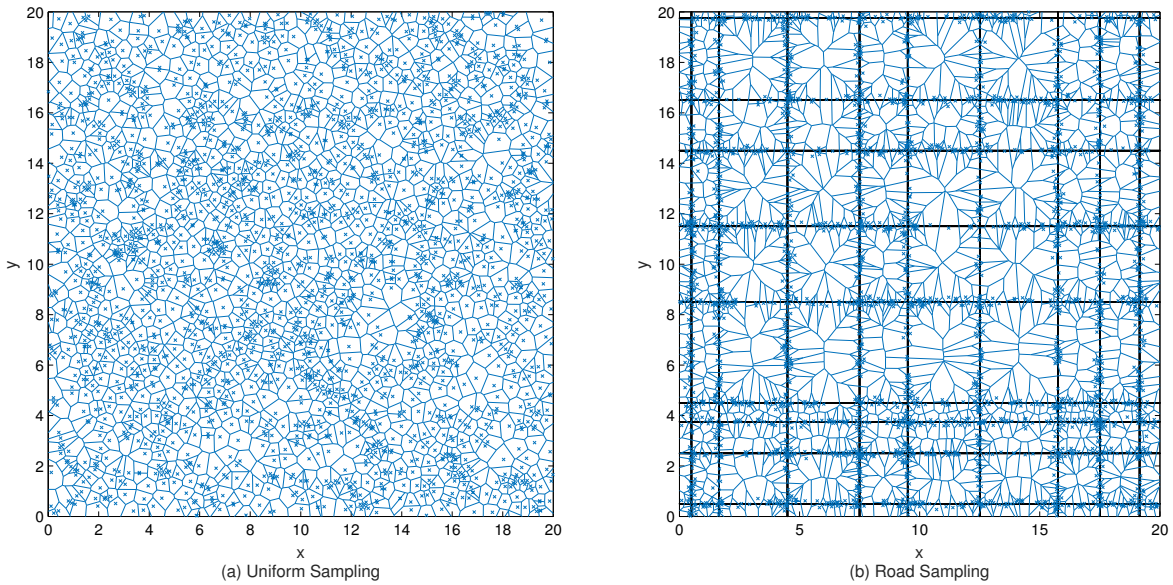


Figure 6.1: Voronoi areas for 2D samples taken from (a) uniform distribution over the domain, (b) uniform distribution over roads

difference given that the RMS value of the signal is 1.70. The reconstruction would be increasingly challenging for the road sampling conditions as higher frequency signals make the regions between the roads more dynamic.

In the future one should investigate adapting the other noisy MSVR sampling reconstruction algorithms given in Section 4.1 to multi-dimensional signals, and be able to design those reconstructions to be computationally viable and efficient. Another possible research direction is to investigate designing a more logical weighting system for the road sampling environment that does not disproportionately assign a high weight to a sample just because its sampled location is proximate to a road's edge.

6.3.3 Continuous Signal Reconstruction from Binary Data

In this thesis, we demonstrated how both binary events and continuous signals could be reconstructed from aggregated raw sensor data. Algorithms were also presented to limit the data transmitted from the vehicles to the Cloud in an attempt to reduce the required

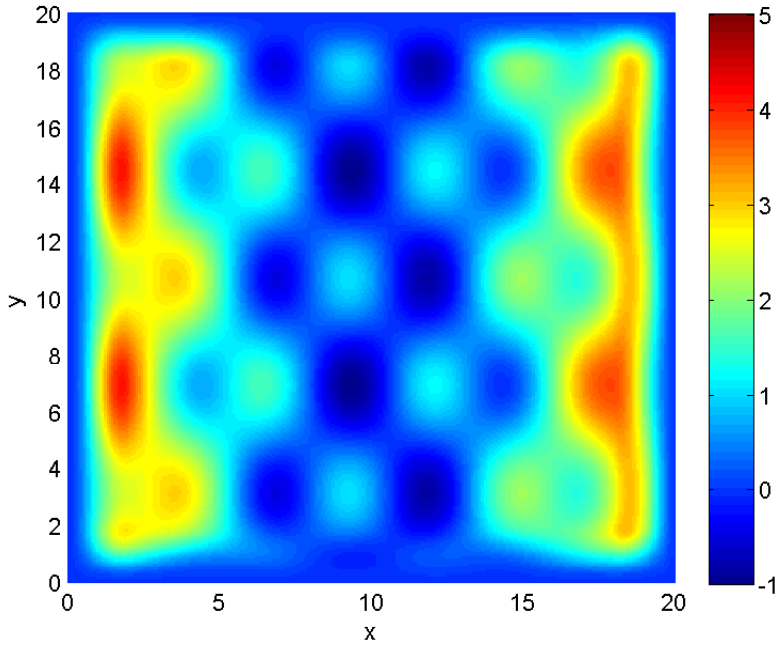


Figure 6.2: Ground truth example two-dimensional signal, given by Equation 6.1, with the indicated color representing the z value

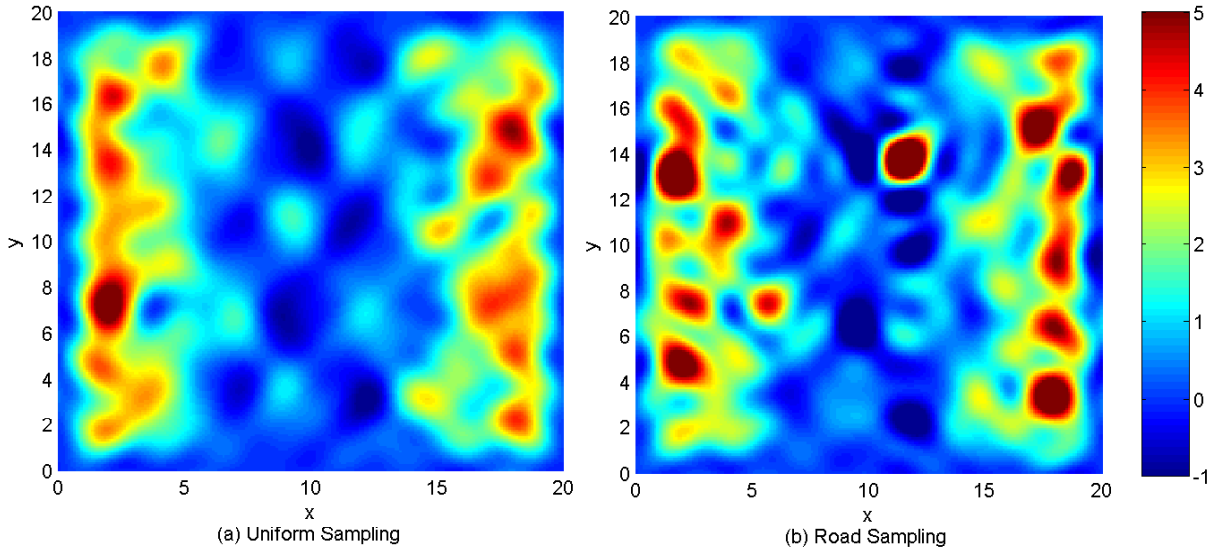


Figure 6.3: Reconstructed two-dimensional signal of Equation 6.1 using (a) uniform sampling, and (b) road sampling conditions

network bandwidth of the detection framework. A natural extension to bandwidth reduction is to further limit the amount of transmitted data by only transmitting binary data instead of the raw sensor data from the vehicles to the Cloud. An initial binary classification could be performed within the vehicle, with only the binary result transmitted to the Cloud where the entire event signal would be reconstructed from the aggregated binary data.

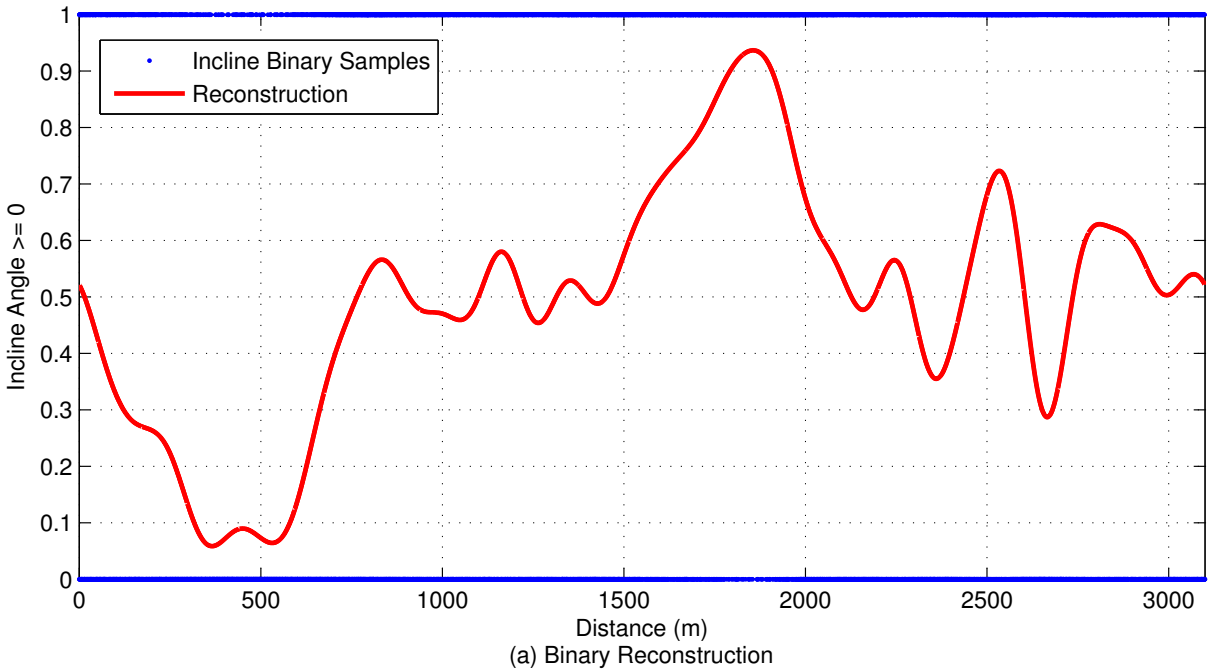
We demonstrate an example of full signal reconstruction from binary data using the road angle detection algorithm, initially presented in Section 5.1.3. In this scheme, individual vehicles first calculate the road angles for each of their measured samples. The road angles are converted to binary values by indicating whether or not the angle is greater than or equal to a threshold of zero degrees. These binary values, indexed by their GPS locations, would then be transmitted to the Cloud where similar results from other vehicles would be aggregated. The MSVR signal reconstruction algorithms is then performed on the binary data, where the reconstructed signal represents the probability of a data sample exceeding the zero degree threshold. With knowledge of the original noise process, the probability signal could then be mapped back to the road angle value that produces the respective probability.

There are a number of challenges involved in this process. A first challenge is determining the reverse mapping algorithm required to estimate the original angle values from the reconstructed probability signal. The algorithm requires both detailed knowledge of the original angle calculation process and also how noise is manifested in the system. The noise is a critical element of the reverse mapping. However unlike reconstructions directly from raw data, binary reconstructions prove to be more challenging when the noise is limited. Consider a situation where the vehicles used noiseless accelerometer sensors and the angle detection algorithm exactly determines the respective road angles. A zero degree binary threshold would not be reverse mappable in this situation. If the vehicles all measured the

incline angle at one road location at 2° and another location at 5°, the aggregated binary data would be identical for both locations.

Measurement noise is therefore beneficial in such a system as it produces non-zero probability distributions of measuring different angles for a given true value. However, detailed knowledge of the noise model is required in the Cloud to adequately reverse map the probability distribution to an angle value. As an example, consider the simulated road from Section 5.1.4. Two example scenarios attempting to reconstruct continuous signals from binary data are presented in Figure 6.4 and Figure 6.5 for data aggregated from 40 vehicles traveling at an average of 50 km/h, with zero-mean white noise on the angle errors with standard deviations $\sigma_y = 3°$ and $\sigma_y = 1.5°$ respectively. The road angle signal is reconstructed fairly well for the example with larger sample value error, since the probability of a measured sample being above or below the 0° threshold is non-negligible for the given road angles. With the lower measurement error, there are ranges where the probability of obtaining measurements on both sides of the threshold are negligible, resulting in reconstructions that fluctuate closely to zero or one. At this level, it is difficult to reconstruct the angle signal since any small error in the probability signal is equivalent to many standard deviations when mapped back to the angle signal. Also, since the reconstruction process treats the binary data as it would any other data, the signal reconstruction is not restricted to having values only in the range between zero and one. Exceeding the probability range results in an undefined mapping back to the angle values, as indicated for some of the extreme regions in Figure 6.5.

To maintain a defined mapping to angle values, the reconstructed probability signal needs to be restricted to the range zero to one. One possible solution is to re-frame the optimization problem determining the frequency coefficients $\mathbf{a}$ as a Karush-Kuhn-Tucker (KKT) [100, 101] conditioned optimization problem such that over the reconstruction domain, $0 < p(x) < 1$.



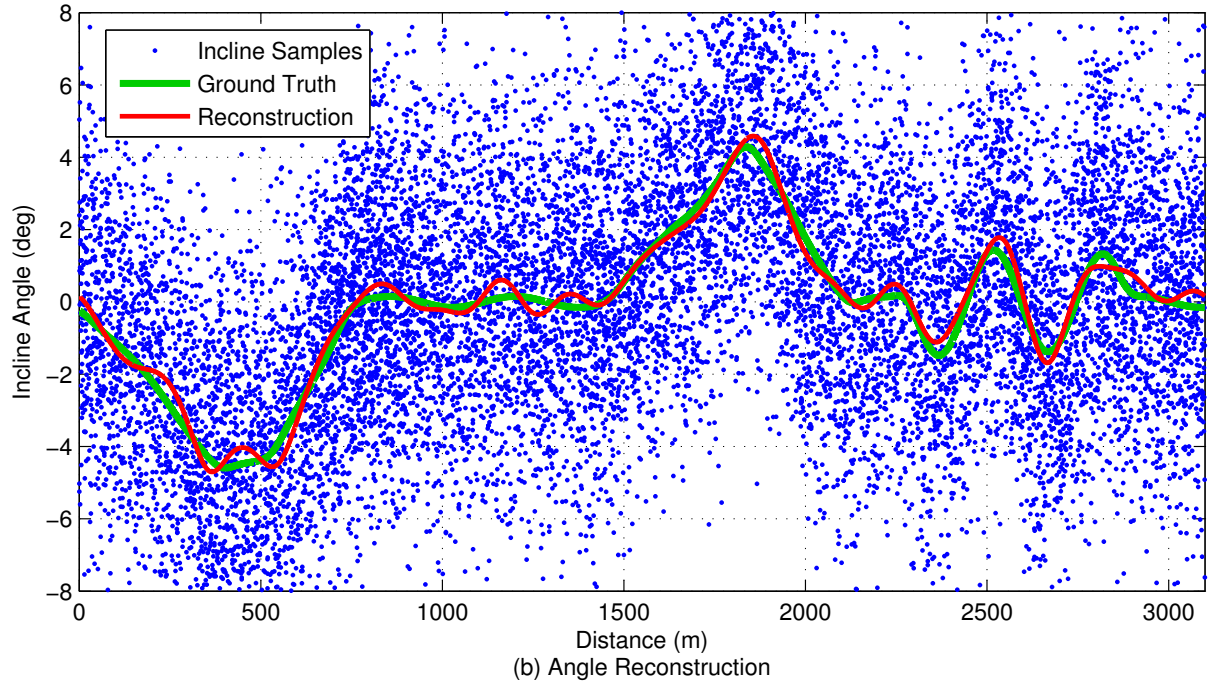


Figure 6.4: Incline angle reconstruction from (a) binary thresholded data, and (b) converted angle signal, with $\sigma_y = 3°$

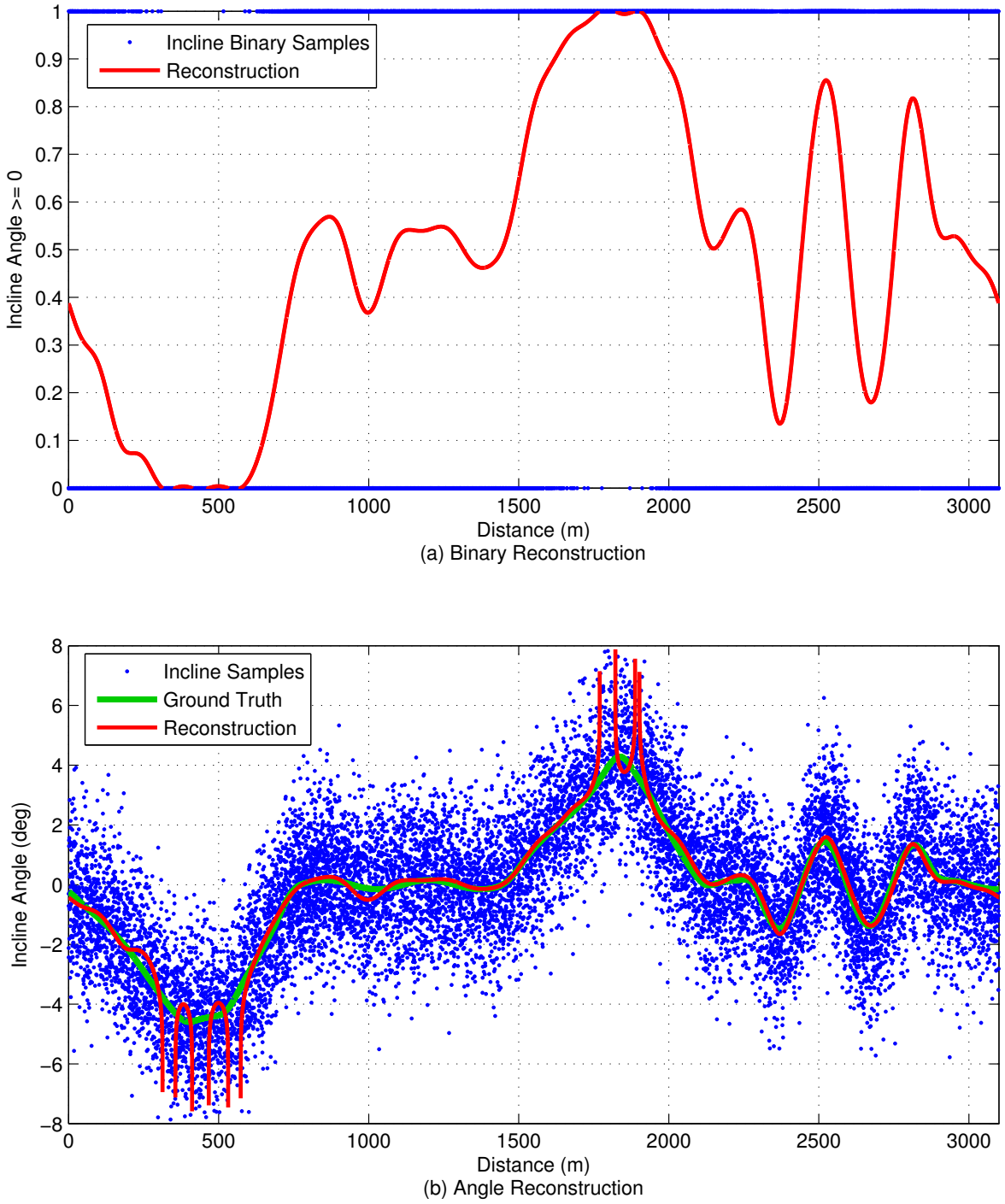


Figure 6.5: Incline angle reconstruction from (a) binary thresholded data, and (b) converted angle signal, with $\sigma_y = 1.5°$

With focus on reconstruction Algorithm 4 with no sample location error, presented in Section 4.1.2.4, the reconstruction problem is modified as follows. Let $\boldsymbol{\xi}$ be a vector of fine-grained locations over the reconstruction domain, $x \in [0, L]$, with spacing δ such that

$$\boldsymbol{\xi} = \begin{bmatrix} 0 & \delta & 2\delta & \dots & L-2\delta & L-\delta & L \end{bmatrix}^T. \tag{6.2}$$

Let $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ be $\left(\frac{L}{\delta}+1\right) \times 1$ sized vectors of KKT multipliers corresponding to each location in $\boldsymbol{\xi}$. Therefore the Lagrange minimization problem is,

$$\min_{\mathbf{a}} \sum_{v=1}^{V} \sum_{j=1}^{n_v} w_j^v \left| p(x_j^v) - y_j^v \right|^2 - \boldsymbol{\lambda}_1^T \sum_{k=-ML}^{ML} a_k e^{\frac{2\pi i k \boldsymbol{\xi}}{L}} + \boldsymbol{\lambda}_2^T \left(\left[\sum_{k=-ML}^{ML} a_k e^{\frac{2\pi i k \boldsymbol{\xi}}{L}} \right] - \mathbf{1}_{\left(\frac{L}{\delta}+1\right)\times 1} \right). \tag{6.3}$$

The solution for $\mathbf{a}$ to create the dual problem is similar to that presented in Section 4.1.2.4, however the vector $\mathbf{b}$ in Equation 4.59 is replaced by a vector with components,

$$b_k = (\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2)^T e^{\frac{-2\pi i k \boldsymbol{\xi}}{L}} + \sum_{v=1}^{V} \sum_{j=1}^{n_v} w_j^v y_j^v e^{\frac{-2\pi i k x_j^v}{L}}. \tag{6.4}$$

This KKT formulation is applied to the $\sigma_y = 1.5°$ situation that without the KKT conditions produced the result in Figure 6.5. The binary reconstruction and resulting incline angle signal are shown in Figure 6.6. The KKT formulation improves the reconstruction and allows it to be defined over the entire domain. However the benefits of this method still become more limited as the noise is reduced and differences in values of the binary signal reconstruction near zero and one represent many standard deviations when mapped back to the angle values.

It would be beneficial in the future to be able to continue to design such a framework so that continuous signals could be reconstructed from aggregated MSVR binary sample values without such detailed knowledge of the noise model. This overall framework would be beneficial from a bandwidth perspective as transmitting a binary value could be considerably more network friendly than transmitting potentially large amounts of raw sensor data.

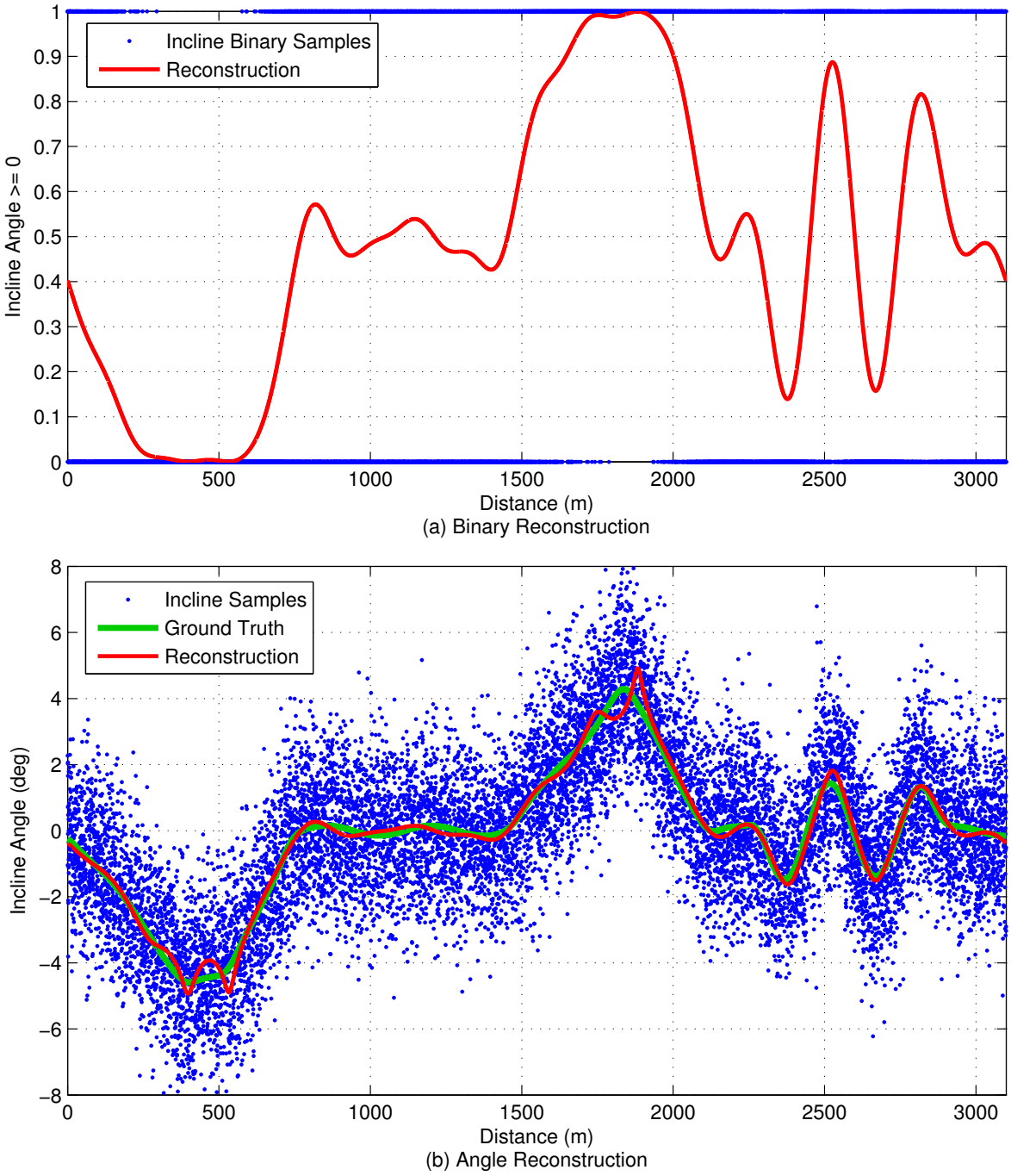


Figure 6.6: Incline angle reconstruction from (a) binary thresholded data, and (b) converted angle signal, with $\sigma_y = 1.5°$ using KKT conditions

6.3.4 Temperature Map

Weather and temperature information is critical environmental knowledge. Reported temperatures on websites for example are usually produced by weather stations spread throughout the country. However weather stations can be sparsely spaced, particularly in rural areas, and may not update their measurements frequently enough to provide current forecasts. However, using the embedded thermometers in vehicles, the Intelligent Transportation System can be leveraged to create current weather maps. Although values can be interpolated in any region, any vehicle that drives in a low density region provides critical data to reconstruct the signal. By adding a temporal factor to the data, as given by the algorithm in Section 4.3.1.1, the reconstructed temperature function can be continually updated, even in regions where data may be sparse or old.

As an example of such reconstructions, data was scraped from 82 weather stations [102] spread between Buffalo, NY, USA and Boston, MA, USA near Highway I-90 between March 1, 2016 - March 7, 2016, spanning 718 km. The weather stations output data at an average rate of one sample per 7.6 minutes over the course of the week, with a minimum station sampling rate of one sample per 31.7 minutes and a maximum rate of one sample per 3.5 minutes (averaged over the week). The data was randomly split such that 95% of the data was used as reconstruction data and 5% of the data was used as test data. To emulate vehicles driving, the reconstruction data was replicated 25 times with simulated zero-mean noise added with standard deviations $\sigma_x = 10$ km and $\sigma_y = 0.3$ °C. This is in addition to any inherent errors that may already exist in the data set. At every instance a test data sample was received, the test data was compared to the reconstruction using the temporal decay algorithm detailed in Section 4.3.1.1. This produced a RMSE value of 0.87 °C over the test data set. The reconstructions for temperature data for two example times over this period are shown in Figure 6.7. Only the data samples displayed have a temporal weighting that exceeds the threshold. The color of the data sample indicates its age. Data

samples older than 70 minutes were not used in the reconstruction for these two times. The time threshold and decay rate depend on the acquisition rate of new data and how much the new data varies from the old data.

In the future it would be beneficial to test the temperature map reconstruction using vehicle data as test data, with comparisons being made to weather stations as an approximation of the ground-truth data. Since the vehicles were only being emulated in this scenario, there were still sparse regions on the reconstruction wherever the weather stations were sparsely spaced, since there were no mobile vehicles to collect data from to fill in the gaps.

It would also be beneficial to reconstruct the temperature map as a two-dimensional function over an area, as per the extensions in Section 6.3.2, instead of doing so in one dimension along a road. This would also provide for a better comparison to the weather stations. Although most of the weather stations the data was aggregated from were within 10 km of the highway, many of the stations were intentionally located near bodies of water or elevated regions, where the changes in environmental conditions were likely to produce more drastic changes in weather than what would be expected in other localized regions. Using vehicle sensors as the data source would allow one to better map out these changes to more accurately reconstruct the temperature distribution for any location.

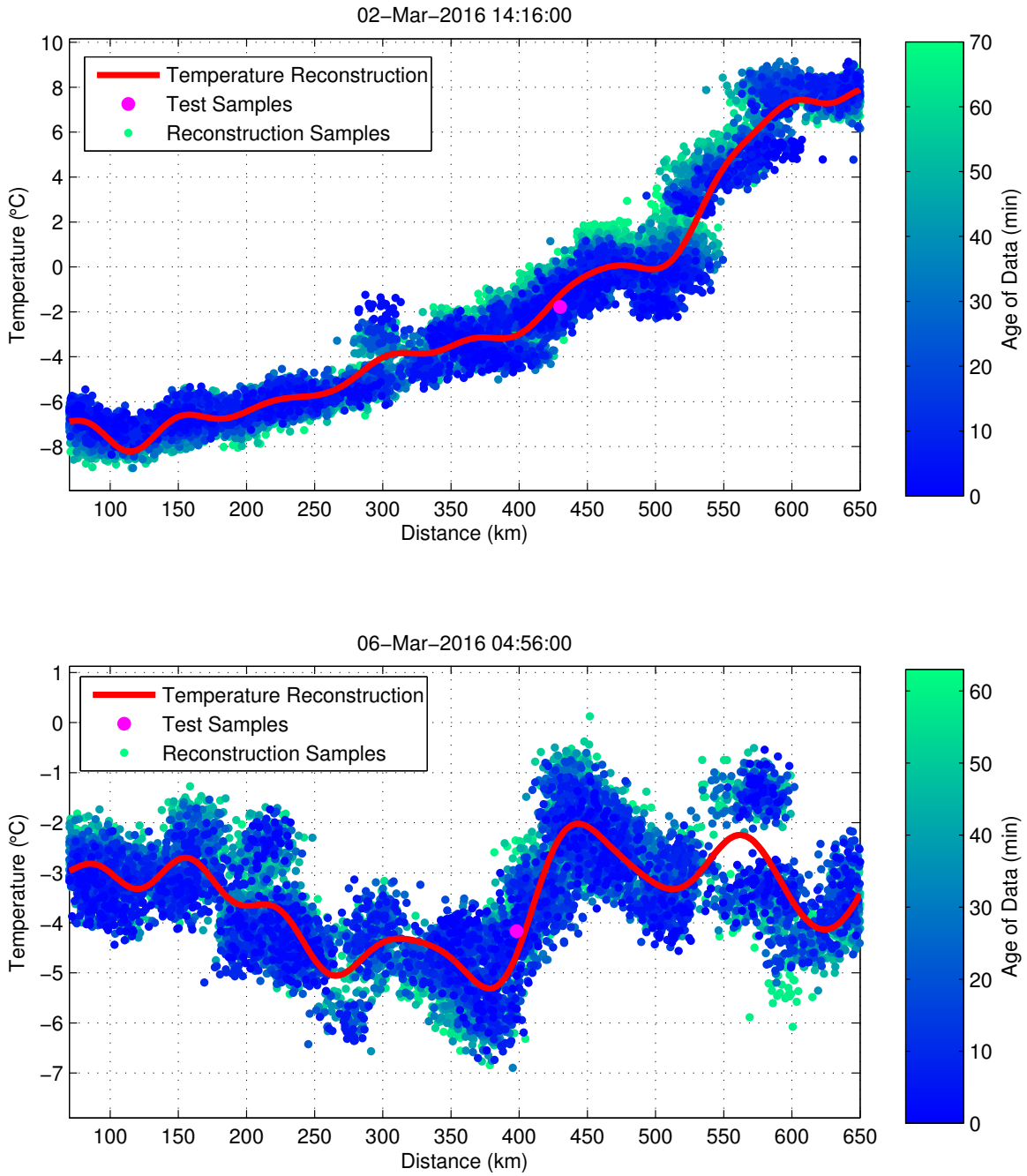


Figure 6.7: Example temperature signal reconstructions at two times, as compared to test data

Appendix A

Joint Reconstruction LMA Equations

A.1 Autocorrelated Error Model

The following derives the LMA equations for minimizing the cost function of Equation 4.65 to jointly reconstruct signals using the autocorrelated error model.

Following the LMA solution, the $(GN + N) \times 1$ vector of cost functions is,

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_{11}^{1T} & \dots & \mathbf{r}_{1\,1}^{VT} & \dots & \mathbf{r}_{1G}^{1T} & \dots & \mathbf{r}_{1\,G}^{VT} & \mathbf{r}_{2}^{1T} & \dots & \mathbf{r}_{2}^{VT} & r_3^1 & \dots & r_3^V \end{bmatrix}^T. \quad \text{(A.1)}$$

The vector $\mathbf{r}_{1g}^v$ is an $n_v \times 1$ vector of cost functions for the sample value error for signal g composed of elements,

$$\mathbf{r}_{1g_j}^v = \sqrt{\frac{\sigma_a^2}{\sigma_{y_g}^2} w_j^v} \left(\sum_{k=-M_gL}^{M_gL} a_{g_k} e^{\frac{2\pi i k (x_j^v - \Delta_j^v)}{L}} - y_{g\,j}^v \right). \quad \text{(A.2)}$$

The vector $\mathbf{r}_2^v$ is a $(n_v - 1) \times 1$ vector for the difference in spacing of consecutive samples from source v, composed of elements

$$r_{2j}^v = \sqrt{\frac{G}{1 - (\rho_{j,j+1}^v)^2}} \left(\sqrt{w_j^v} \Delta_j^v - \sqrt{w_{j+1}^v} \Delta_{j+1}^v \right). \quad \text{(A.3)}$$

The scalar r_3^v represents the cost function for the location error for the final sample from source v, given by

$$r_3^v = \sqrt{G w_{n_v}^v}\ \Delta_{n_v}^v. \quad \text{(A.4)}$$

The Jacobian matrix is defined similarly to Equation 4.25, however $\mathbf{J}$ is a $(GN+N)\times(GN+\sum_{g=1}^{G}(2M_gL+1))$ block matrix. The block matrix $\mathbf{J}_1$ is a $GN\times N$ block matrix composed as

$$\mathbf{J}_1 = \begin{bmatrix} \mathbf{J}_{1\,1} \\ \vdots \\ \mathbf{J}_{1\,G} \end{bmatrix}, \tag{A.5}$$

The block matrices $\mathbf{J}_{1\,g}$ are $N\times N$ diagonal block matrices, where

$$\mathbf{J}_{1\,g} = \begin{bmatrix} \mathbf{J}^1_{1\,g} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}^2_{1\,g} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}^V_{1\,g} \end{bmatrix}, \tag{A.6}$$

and $\mathbf{J}^v_{1\,g}$ are diagonal matrices composed of elements

$$J^v_{1\,g_{jj}} = -\sqrt{\frac{\sigma_a^2}{\sigma_{y_g}^2}w_j^v}\left(\sum_{k=-M_gL}^{M_gL} a_{g_k}\left(\frac{2\pi ik}{L}\right)e^{\frac{2\pi ik(x_j^v-\Delta_j^v)}{L}}\right). \tag{A.7}$$

The matrix $\mathbf{J}_2$ is a block matrix composed similarly to Equation 4.28, however the component block matrices, $\mathbf{J}_2^v$ have elements on the diagonal and superdiagonal given by

$$J^v_{2\,j,j} = \sqrt{\frac{Gw_j^v}{1-(\rho^v_{j,j+1})^2}}, \tag{A.8}$$

$$J^v_{2\,j,j+1} = -\sqrt{\frac{Gw_{j+1}^v}{1-(\rho^v_{j,j+1})^2}}. \tag{A.9}$$

The matrix $\mathbf{J}_3$ is defined similarly to Equation 4.31, however the non-zero entry in the block matrices, $\mathbf{J}_3^v$ is defined as

$$J^v_{3\,1,n_v} = \sqrt{Gw^v_{n_v}}. \tag{A.10}$$

The matrix $\mathbf{A}$ is a $GN \times \left(\sum_{g=1}^{G}(2M_gL+1)\right)$ block diagonal matrix defined as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_G \end{bmatrix}, \tag{A.11}$$

where similar to Equation 4.33,

$$\mathbf{A}_g = \begin{bmatrix} \mathbf{A}_g^{1T} & \dots & \mathbf{A}_g^{VT} \end{bmatrix}^T. \tag{A.12}$$

The matrix $\mathbf{A}_g^v$ is composed of elements,

$$A_{gjl}^v = \sqrt{\frac{\sigma_a^2}{\sigma_{y_g}^2} w_j^v} e^{\frac{2\pi i(l-M_gL)\left(x_j^v - \Delta_j^v\right)}{L}}. \tag{A.13}$$

A.2 Correlated Error Model

The following derives the LMA equations for minimizing the cost function of Equation 4.67 to jointly reconstruct signals using the correlated error model.

Following the LMA solution, we define the $(GN+N+V)$-dimensional vector of cost functions $\mathbf{r}$ as

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_{11}^{1T} & \dots & \mathbf{r}_{1\,1}^{VT} & \dots & \mathbf{r}_{1G}^{1T} & \dots & \mathbf{r}_{1\,G}^{VT} & \mathbf{r}_2^{1T} & \dots & \mathbf{r}_2^{VT} & r_3^1 & \dots & r_3^V \end{bmatrix}^T. \tag{A.14}$$

The vector $\mathbf{r}_{1g}^v$ is an $n_v \times 1$ sized vector representing the sample value error cost functions for the samples from the v^{th} vehicle and g^{th} signal, composed of elements

$$\mathbf{r}_{1g_j}^v = \sqrt{\frac{w_j^v}{\sigma_{y_g}^2}} \left(\sum_{k=-M_gL}^{M_gL} a_{g_k} e^{\frac{2\pi ik\left(x_j^v - \Delta_j^v\right)}{L}} - y_{g\,j}^v \right). \tag{A.15}$$

The vector $\mathbf{r}_2^v$ is an $n_v \times 1$ sized vector of cost functions composed of elements,

$$r_{2j}^v = \sqrt{G(A_v - B_v)w_j^v}\,\Delta_j^v. \tag{A.16}$$

The components r_3^v are scalar cost functions where

$$r_3^v = \sqrt{GB_v} \sum_{j=1}^{n_v} \sqrt{w_j^v} \Delta_j^v. \tag{A.17}$$

The Jacobian matrix is defined similarly to Equation 4.40, however $\mathbf{J}_1$ is a $GN \times N$ block matrix defined similarly to Equation A.5. The block matrices $\mathbf{J}_{1g}$ are $N \times N$ diagonal block matrices, where

$$\mathbf{J}_{1g} = \begin{bmatrix} \mathbf{J}_{1g}^1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{1g}^2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}_{1\,g}^V \end{bmatrix}, \tag{A.18}$$

and $\mathbf{J}_{1g}^v$ are diagonal matrices composed of elements

$$J_{1\,g_{jj}}^v = -\sqrt{\frac{w_j^v}{\sigma_{y_g}^2}} \left(\sum_{k=-M_g L}^{M_g L} a_{g_k} \left(\frac{2\pi i k}{L} \right) e^{\frac{2\pi i k (x_j^v - \Delta_j^v)}{L}} \right). \tag{A.19}$$

The matrix $\mathbf{J}_2$ is defined by block diagonal matrices similarly to Equation 4.28, however the diagonal values for the block matrices $\mathbf{J}_2^v$ are now equal to

$$J_{2\,jj}^v = \sqrt{G(A_v - B_v) w_j^v}. \tag{A.20}$$

The matrix $\mathbf{J}_3$ is also defined similarly to Equation 4.31 as given previously however the composition block matrices $\mathbf{J}_3^v$ are now composed of elements

$$J_{3\,1j}^v = \sqrt{GB_v w_j^v}. \tag{A.21}$$

The matrix $\mathbf{A}$ is composed similarly to that of Equation A.11, but with components

$$A_{g_{jl}}^v = \sqrt{\frac{w_j^v}{\sigma_{y_g}^2}} e^{\frac{2\pi i (l - M_g L)\left(x_j^v - \Delta_j^v\right)}{L}}. \tag{A.22}$$

A.3 Fixed Source Error Model

The following derives the LMA equations for minimizing the cost function of Equation 4.68 to jointly reconstruct signals using the fixed source error model.

Following the LMA optimization framework, the $(GN + V)$-dimensional vector of cost functions, $\mathbf{r}$, is given by

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_{11}^{1T} & \dots & \mathbf{r}_{1\,1}^{VT} & \dots & \mathbf{r}_{1G}^{1T} & \dots & \mathbf{r}_{1\,G}^{VT} & r_2^{1T} & \dots & r_2^{VT} \end{bmatrix}^T. \tag{A.23}$$

The vector $\mathbf{r}_{1g}^v$ is an $n_v \times 1$ sized vector representing the sample value error cost functions for the samples from the v^{th} vehicle and g^{th} signal, composed of elements

$$\mathbf{r}_{1g_j}^v = \sqrt{\frac{\sigma_v^2}{\sigma_{y_g}^2} w_j^v} \left(\sum_{k=-M_g L}^{M_g L} a_{g_k} e^{\frac{2\pi i k (x_j^v - \Delta^v)}{L}} - y_{g\,j}^v \right). \tag{A.24}$$

The scalar value r_2^v is equal to,

$$r_2^v = \sqrt{G \sum_{j=1}^{n_v} w_j^v}\ \Delta^v. \tag{A.25}$$

The Jacobian matrix, $\mathbf{J}$ is defined similarly to Equation 4.49 however with $\mathbf{J}_1$ and $\mathbf{A}$ composed of block matrices similar to Equations A.5 and A.11, respectively. As for the components of the block matrices in the Jacobians, $\mathbf{J}_{1g}$ is an $N \times V$ matrix composed similarly to Equation A.6, where $\mathbf{J}_{1g}^v$ is an $n_v \times 1$ matrix composed of elements,

$$\mathbf{J}_{1g_{j1}}^v = -\sqrt{\frac{\sigma_v^2}{\sigma_{y_g}^2} w_j^v} \left(\sum_{k=-M_g L}^{M_g L} a_{g_k} \left(\frac{2\pi i k}{L} \right) e^{\frac{2\pi i k (x_j^v - \Delta^v)}{L}} \right). \tag{A.26}$$

$\mathbf{J}_2$ is a $V \times V$ diagonal matrix where,

$$J_{2vv} = \sqrt{G \sum_{j=1}^{n_v} w_j^v}. \tag{A.27}$$

The block matrices $\mathbf{A}_g$ are also composed by block matrices in the style of Equation A.12 where the elements of $\mathbf{A}_g^v$ are equal to,

$$A_{g_{jl}}^v = \sqrt{\frac{\sigma_v^2}{\sigma_{y_g}^2} w_j^v} e^{\frac{2\pi i (l - M_g L)\left(x_j^v - \Delta^v\right)}{L}}. \tag{A.28}$$

A.4 Asynchronous Data Collection

The following derives the LMA equations for minimizing the cost function of Equation 4.71 to jointly reconstruct signals using the fixed source error model for asynchronous data collection. The vector of cost functions, $\mathbf{r}$, is defined similarly to Equation A.23, however the vector $\mathbf{r}_{1g}^{v}$ is now of size $n_{vg} \times 1$ with components

$$\mathbf{r}_{1g_j}^{v} = \sqrt{\frac{\sigma_v^2}{\sigma_{y_g}^2} w_{g_j}^{v}} \left(\sum_{k=-M_g L}^{M_g L} a_{g_k} e^{\frac{2\pi i k \left(x_{g_j}^{v} - \Delta^{v} \right)}{L}} - y_{g_j}^{v} \right), \tag{A.29}$$

and the scalar value r_2^v from the same Equation A.23 is now given by

$$r_2^{v} = \sqrt{\sum_{g=1}^{G} \sum_{j=1}^{n_{vg}} w_{g_j}^{v}} \ \Delta^{v}. \tag{A.30}$$

The Jacobian of Equation 4.49 has dimensions $(V + \sum_{g=1}^{G} N_g) \times \left(V + \sum_{g=1}^{G} (2M_g L + 1) \right)$ where the $\mathbf{J}_{1g}$ block matrices are of size $N_g \times V$, and are constructed as block diagonal matrices similar to Equation A.6, where the matrix $\mathbf{J}_{1g}^{v}$ is an $n_{vg} \times 1$ matrix with components

$$J_{1\, g_{j1}}^{v} = -\sqrt{\frac{\sigma_v^2}{\sigma_{y_g}^2} w_{g_j}^{v}} \left(\sum_{k=-M_g L}^{M_g L} a_{g_k} \left(\frac{2\pi i k}{L} \right) e^{\frac{2\pi i k (x_{g_j}^{v} - \Delta^{v})}{L}} \right). \tag{A.31}$$

The matrix $\mathbf{J}_2$ is a $V \times V$ diagonal matrix defined similarly to Equation A.27, but with components

$$J_{2vv} = \sqrt{\sum_{g=1}^{G} \sum_{j=1}^{n_{vg}} w_{g_j}^{v}} \ . \tag{A.32}$$

The matrix $\mathbf{A}$ is defined similarly to Equation A.11, however the block matrix $\mathbf{A}_g$ is of size $N_g \times (2M_g L + 1)$ where its block matrix components $\mathbf{A}_g^v$ (constructed as in Equation A.13) are the $n_{vg} \times (2M_g L + 1)$ matrices composed of elements

$$A_{g_{jl}}^{v} = \sqrt{\frac{\sigma_v^2}{\sigma_{y_g}^2} w_{g_j}^{v}} e^{\frac{2\pi i (l - M_g L)(x_{g_j}^{v} - \Delta^{v})}{L}}. \tag{A.33}$$

Appendix B

Error Propagation Derivation

Given the MSVR sampling reconstruction algorithm in Section 4.1.2.4, we can frame the reconstruction steps and the energy calculation as the following series of functions:

$$p(x) = f_1(\mathbf{a}) \qquad \text{Equation 4.10} \tag{B.1}$$

$$\mathbf{a} = f_2(\mathbf{T}^{-1}, \mathbf{b}) \qquad \text{Equation 4.62} \tag{B.2}$$

$$\mathbf{b} = f_3(\mathbf{x}, \mathbf{y}) \qquad \text{Equation 4.59} \tag{B.3}$$

$$\mathbf{T}^{-1} = f_4(\mathbf{T}) \qquad \text{Matrix Inverse} \tag{B.4}$$

$$\mathbf{T} = f_5(\mathbf{x}) \qquad \text{Equation 4.60} \tag{B.5}$$

$$E_p = f_6(\mathbf{a}) \qquad \text{Energy Calculation} \tag{B.6}$$

As a function of the sampling error, we can set the Jacobian matrices for the frequency coefficient calculation and energy calculation as

$$\mathbf{J}_a = \frac{\partial f_2}{\partial \mathbf{b}} \frac{\partial f_3}{\partial \mathbf{y}}, \tag{B.7}$$

$$\mathbf{J}_E = \frac{\partial f_6}{\partial \mathbf{a}} \mathbf{J}_a, \tag{B.8}$$

$$\tag{B.9}$$

where the partial derivative notation indicates the Jacobian matrix for the respective function with respect to the indicated vector of variables.

We next need to construct the Jacobian matrices for each of these functions. First note that given the vector construction of complex values, that we can represent $\mathbf{x} = x_1 + ix_2$ in matrix form for multiplication purposes using the function $\mathbf{G}(\mathbf{x})$ such that,

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} x_1 & -x_2 \\ x_2 & x_1 \end{bmatrix}. \tag{B.10}$$

The Jacobian $\frac{\partial f_2}{\partial \mathbf{b}}$ is a $2(2ML+1) \times 2(2ML+1)$ sized matrix composed of the following 2×2 block matrices

$$\frac{\partial f_2}{\partial \mathbf{b}} = \begin{bmatrix} \frac{\partial f_{2-ML}}{\partial b_{-ML}} & \frac{\partial f_{2-ML}}{\partial b_{-ML+1}} & \cdots & \frac{\partial f_{2-ML}}{\partial b_{ML}} \\ \frac{\partial f_{2-ML+1}}{\partial b_{-ML}} & \frac{\partial f_{2-ML+1}}{\partial b_{-ML+1}} & \cdots & \frac{\partial f_{2-ML+1}}{\partial b_{ML}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{2ML}}{\partial b_{-ML}} & \frac{\partial f_{2ML}}{\partial b_{-ML+1}} & \cdots & \frac{\partial f_{2ML}}{\partial b_{ML}} \end{bmatrix}. \tag{B.11}$$

The component block matrices $\frac{\partial f_{2_k}}{\partial b_l}$ are the 2×2 Jacobian matrices corresponding to differentiating the frequency component a_k with respect to b_l, determined as,

$$\frac{\partial f_{2k}}{\partial b_l} = \mathbf{G}(T_{kl}^{-1}). \tag{B.12}$$

The Jacobian matrix $\frac{\partial f_3}{\partial \mathbf{y}}$ is a $2(2ML+1) \times 2N$ sized block Jacobian matrix, composed as,

$$\frac{\partial f_3}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial f_{3-ML}}{\partial \mathbf{y}^1} & \frac{\partial f_{3-ML}}{\partial \mathbf{y}^2} & \cdots & \frac{\partial f_{3-ML}}{\partial \mathbf{y}^V} \\ \frac{\partial f_{3-ML+1}}{\partial \mathbf{y}^1} & \frac{\partial f_{3-ML+1}}{\partial \mathbf{y}^2} & \cdots & \frac{\partial f_{3-ML+1}}{\partial \mathbf{y}^V} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{3ML}}{\partial \mathbf{y}^1} & \frac{\partial f_{3ML}}{\partial \mathbf{y}^2} & \cdots & \frac{\partial f_{3ML}}{\partial \mathbf{y}^V} \end{bmatrix}. \tag{B.13}$$

The block matrices $\frac{\partial f_{3_k}}{\partial \mathbf{y}^v}$ are $2 \times 2n_v$ block matrices, composed as

$$\frac{\partial f_{3k}}{\partial \mathbf{y}^v} = \begin{bmatrix} \frac{\partial f_{3k}}{\partial y_1^v} & \frac{\partial f_{3k}}{\partial y_2^v} & \cdots & \frac{\partial f_{3k}}{\partial y_{n_v}^v} \end{bmatrix}, \tag{B.14}$$

where $\frac{\partial f_{3_k}}{\partial y_j^v}$ is a 2×2 Jacobian matrix of the derivative of b_k with respect to y_j^v, such that

$$\frac{\partial f_{3_k}}{\partial y_j^v} = \begin{bmatrix} w_j^v \cos\left(\frac{2\pi k x_j^v}{L}\right) & w_j^v \sin\left(\frac{2\pi k x_j^v}{L}\right) \\ -w_j^v \sin\left(\frac{2\pi k x_j^v}{L}\right) & w_j^v \cos\left(\frac{2\pi k x_j^v}{L}\right) \end{bmatrix}. \tag{B.15}$$

The Jacobian matrix $\frac{\partial f_6}{\partial \mathbf{a}}$ is a $2 \times 2(2ML+1)$ block matrix, composed as

$$\frac{\partial f_6}{\partial \mathbf{a}} = \begin{bmatrix} \frac{\partial f_6}{\partial a_{-ML}} & \frac{\partial f_6}{\partial a_{-ML+1}} & \cdots & \frac{\partial f_6}{\partial a_{ML}} \end{bmatrix}, \tag{B.16}$$

where $\frac{\partial f_6}{\partial a_k}$ is the 2×2 Jacobian matrix of the derivative of the signal energy with respect to a_k such that,

$$\frac{\partial f_6}{\partial a_k} = \begin{bmatrix} 2\mathrm{Re}(a_k) & 2\mathrm{Im}(a_k) \\ 0 & 0 \end{bmatrix}. \tag{B.17}$$

Only the $(1,1)$ element of J_E is relevant and non-zero since E_p is purely real.

References

[1] A. Albert, "Comparison of event-triggered and time-triggered concepts with regard to distributed control systems," *Embedded World*, vol. 2004, pp. 235–252, 2004.

[2] "E/E diagnostic test modes (J1979)," Society of automotive engineers, 2014.

[3] A. Fox, B. V. K. Vijaya Kumar, J. Chen, and F. Bai, "Crowdsourcing undersampled vehicular sensor data for pothole detection," in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (IEEE SECON 2015)*, Seattle, USA, Jun. 2015, pp. 515–523.

[4] M. Jain, A. P. Singh, S. Bali, and S. Kaul, "Speed-Breaker Early Warning System," in *Proceedings of the 6th USENIX/ACM Workshop on Networked Systems for Developing Regions*. New York, NY, USA: ACM, 2012.

[5] D. Pomerleau, "Visibility estimation from a moving vehicle using the RALPH vision system," in *Intelligent Transportation System, 1997. ITSC '97., IEEE Conference on*, Nov 1997, pp. 906–911.

[6] J. Yoon, B. Noble, and M. Liu, "Surface street traffic estimation," in *MobiSys '07: 5th International conference on Mobile systems, applications and services*. New York, NY, USA: ACM, 2007, pp. 220–232.

[7] J. Choi, B. T. Ahn, and I. S. Kweon, "Crosswalk and traffic light detection via integral framework," in *Frontiers of Computer Vision, (FCV), 2013 19th Korea-Japan Joint Workshop on*, Jan 2013, pp. 309–312.

[8] H. Gomez-Moreno, S. Maldonado-Bascon, P. Gil-Jimenez, and S. Lafuente-Arroyo, "Goal evaluation of segmentation algorithms for traffic sign recognition," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 4, pp. 917–930, Dec 2010.

[9] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 108–118, Jun 2000.

[10] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, "ParkNet: Drive-by sensing of road-side parking statistics," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10. New York, NY, USA: ACM, 2010, pp. 123–136.

[11] A. Fox, B. V. K. Vijaya Kumar, and F. Bai, "Multi-source variable-rate sampled sig-

nal reconstructions in vehicular CPS," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016)*, San Francisco, USA, Apr. 2016, pp. 946–954.

[12] W. Zhang and V. Taliwal, "Using lane tracker data to improve lane-level digital maps created with probe vehicle data," in *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, vol. 1, Oct 2003, pp. 585–589.

[13] M. Eichner and T. Breckon, "Integrated speed limit detection and recognition from real-time video," in *Intelligent Vehicles Symposium, 2008 IEEE*, June 2008, pp. 626–631.

[14] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "GreenGPS: A participatory sensing fuel-efficient maps application," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10. New York, NY, USA: ACM, 2010, pp. 151–164.

[15] K. H. Johansson, M. Törngren, and L. Nielsen, *Handbook of Networked and Embedded Control Systems*. Boston, MA: Birkhäuser Boston, 2005, ch. Vehicle Applications of Controller Area Network, pp. 741–765.

[16] Y. Jiang, H. Qiu, M. McCartney, W. G. J. Halfond, F. Bai, D. Grimm, and R. Govindan, "CARLOG: A platform for flexible and efficient automotive sensing," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, SenSys '14. New York, NY, USA: ACM, 2014, pp. 221–235.

[17] W. Viriyasitavat, F. Bai, and O. Tonguz, "Dynamics of network connectivity in urban vehicular networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 3, pp. 515–533, March 2011.

[18] J. Fernandez, K. Borries, L. Cheng, B. V. K. Vijaya Kumar, D. Stancil, and F. Bai, "Performance of the 802.11p physical layer in vehicle-to-vehicle environments," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 1, pp. 3–14, Jan 2012.

[19] B. Yu and F. Bai, "ETP: Encounter transfer protocol for opportunistic vehicle communication," in *INFOCOM, 2011 Proceedings IEEE*, April 2011, pp. 2201–2209.

[20] I. Leontiadis, P. Costa, and C. Mascolo, "Persistent content-based information dissemination in hybrid vehicular networks," in *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, March 2009, pp. 1–10.

[21] X. Liu, J. Biagioni, J. Eriksson, Y. Wang, G. Forman, and Y. Zhu, "Mining large-scale, sparse GPS traces for map inference: Comparison of approaches," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12. New York, NY, USA: ACM, 2012, pp. 669–677.

[22] J. Davics, A. Beresford, and A. Hopper, "Scalable, distributed, real-time map generation," *Pervasive Computing, IEEE*, vol. 5, no. 4, pp. 47–54, Oct 2006.

[23] A. Hofleitner, R. Herring, and A. Bayen, "Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning," *Transportation*

Research Part B: Methodological, vol. 46, no. 9, pp. 1097 – 1122, 2012.

[24] *CarSim: Quick Start Guide*, Mechanical Simulation Corporation, Ann Arbor, MI, 2013. [Online]. Available: http://www.carsim.com/

[25] J. Wilkinson, C. W. Mousseau, and T. Klingler, "Brake response time measurement for a HIL vehicle dynamics simulator," SAE Technical Paper, Tech. Rep., 2010.

[26] B. Parkinson and J. Spilker, *Global Positioning System: Theory and Applications*, Global Positioning System: Theory and Applications. American Institute of Aeronautics and Astronautics, 1996, no. v. 1.

[27] G. Blewitt, "Basics of the GPS technique: observation equations," *Geodetic applications of GPS*, pp. 10–54, 1997.

[28] G. Strazdins, A. Mednis, R. Zviedris, G. Kanonirs, and L. Selavo, "Virtual ground truth in vehicular sensing experiments: How to mark it accurately," in *SENSORCOMM 2011, The Fifth International Conference on Sensor Technologies and Applications*, 2011, pp. 295–300.

[29] A. Kleusberg and R. B. Langley, "The limitations of GPS," *GPS World*, vol. 1, no. 2, 1990.

[30] M. Weiss, "Apparent diurnal effects in the global positioning system," *IEEE Transactions on Instrumentation and Measurement*, vol. 38, no. 5, pp. 991–997, Oct 1989.

[31] M. C. Olynik, "Temporal characteristics of GPS error sources and their impact on relative positioning," Master's thesis, University of Calgary, Calgary, Alberta, Canada, 2002.

[32] K. L. Senior, J. R. Ray, and R. L. Beard, "Characterization of periodic variations in the GPS satellite clocks," *GPS Solutions*, vol. 12, no. 3, pp. 211–225, 2008.

[33] M. G. Ferguson, "Global positioning system (GPS) error source prediction," Master's thesis, Air Force Institute ofTechnology, March 2000.

[34] B. W. Parkinson and P. K. Enge, "Differential GPS," *Global Positioning System: Theory and applications.*, vol. 2, pp. 3–50, 1996.

[35] J. Klobuchar, "Ionospheric effects on GPS," *Global Positioning System: Theory and applications.*, vol. 1, pp. 485–515, 1996.

[36] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.

[37] R. B. Langley, "Dilution of precision," *GPS world*, vol. 10, no. 5, pp. 52–59, 1999.

[38] R. J. Carroll, *Measurement error in nonlinear models: a modern perspective*, Monographs on statistics and applied probability. Boca Raton: Chapman & Hall/CRC, 2006.

[39] A. El-Rabbany, *The Effect of Physical Correlations on the Ambiguity Resolution and Accuracy Estimation in GPS Differential Positioning*. Department of Geodesy and Geomatics Engineering, University of New Brunswick, 1994.

[40] J. Rankin, "An error model for sensor simulation GPS and differential GPS," in

Position Location and Navigation Symposium, 1994., IEEE, Apr 1994, pp. 260–266.

[41] P. Jansson and C.-G. Persson, "The effect of correlation on uncertainty estimates - with GPS examples," *Journal of Geodetic Science*, vol. 3, pp. 111–120, Sep. 2013.

[42] T. Strohmer and J. Tanner, "Fast reconstruction algorithms for periodic nonuniform sampling with applications to time-interleaved ADCs," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 3, April 2007, pp. 881–884.

[43] S. Roques and C. Thiebaut, "Spectral contents of astronomical unequally spaced time-series: contribution of time-frequency and time-scale analyses," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, April 2003, pp. VI–473–6 vol.6.

[44] F. Marvasti, *Nonuniform sampling: theory and practice*. Springer Science & Business Media, 2012.

[45] N. Levinson, *Gap and Density Theorems*, American mathematical society colloquium publications. American Mathematical Society, 1940.

[46] P. Butzer and G. Hinsen, "Reconstruction of bounded signals from pseudo-periodic, irregularly spaced samples," *Signal Processing*, vol. 17, no. 1, pp. 1–17, 1989.

[47] J. M. Whittaker, *Interpolatory function theory*. The University Press, 1935, vol. 33.

[48] E. T. Whittaker, "XVIII.On the functions which are represented by the expansions of the interpolation-theory," *Proceedings of the Royal Society of Edinburgh*, vol. 35, pp. 181–194, Jan 1915.

[49] C. E. Shannon, "Communication in the presence of noise," *Proc. Institute of Radio Engineers*, vol. 37, no. 1, pp. 10–21, 1949.

[50] T. Strohmer and J. Tanner, "Fast reconstruction methods for bandlimited functions from periodic nonuniform sampling," *SIAM J. Numer. Anal.*, vol. 44, no. 3, pp. 1073–1094, Mar. 2006.

[51] T. Strohmer, "Numerical analysis of the non-uniform sampling problem," *Journal of computational and applied mathematics*, vol. 122, no. 1, pp. 297–316, 2000.

[52] K. Gröchenig, "A discrete theory of irregular sampling," *Linear Algebra and its applications*, vol. 193, pp. 129–150, 1993.

[53] H. G. Feichtinger, K. Gröchenig, and T. Strohmer, "Efficient numerical methods in non-uniform sampling theory," *Numer. Math.*, vol. 69, no. 4, pp. 423–440, Feb. 1995.

[54] L. Wasserman, *All of Nonparametric Statistics (Springer Texts in Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[55] S. Efromovich, *Nonparametric curve estimation: methods, theory, and applications*. Springer Science & Business Media, 2008.

[56] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly Journal of Applied Mathmatics*, vol. II, no. 2, pp. 164–168, 1944.

[57] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parame-

ters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[58] M. K. Transtrum and J. P. Sethna, "Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization," *arXiv preprint arXiv:1201.5885*, 2012.

[59] F. de Hoog, "A new algorithm for solving Toeplitz systems of equations," *Linear Algebra and its Applications*, vol. 8889, pp. 123–138, 1987.

[60] G. Voronoi, "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs." *Journal für die reine und angewandte Mathematik*, vol. 134, pp. 198–287, 1908.

[61] F. Aurenhammer, "Voronoi diagrams - a survey of a fundamental geometric data structure," *ACM Comput. Surv.*, vol. 23, no. 3, pp. 345–405, Sep. 1991.

[62] A. Fox, V. Sharma, and B. V. K. Vijaya Kumar, "Signal reconstruction for multi-source variable-rate samples with autocorrelated errors in variables," in *2016 IEEE Statistical Signal Processing Workshop (SSP 2016)*, Palma de Mallorca, Spain, Jun. 2016.

[63] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09. New York, NY, USA: ACM, 2009, pp. 85–98.

[64] T. Strohmer, "On the estimation of the bandwidth of nonuniformly sampled signals," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 4, Mar 1999, pp. 2047–2050.

[65] B. Hall, "On the propagation of uncertainty in complex-valued quantities," *Metrologia*, vol. 41, no. 3, pp. 173–177, 2004.

[66] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[67] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001.

[68] J. Ahn, Y. Wang, B. Yu, F. Bai, and B. Krishnamachari, "RISA: Distributed road information sharing architecture," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 1494–1502.

[69] H. E. Tseng, B. Ashrafi, D. Madau, T. A. Brown, and D. Recker, "The development of vehicle stability control at Ford," *IEEE/ASME Transactions on Mechatronics*, vol. 4, no. 3, pp. 223–234, Sep 1999.

[70] C. R. Carlson and J. C. Gerdes, "Optimal rollover prevention with steer by wire and differential braking," in *ASME 2003 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers, 2003, pp. 345–354.

[71] D. Yanakiev and I. Kanellakopoulos, "Speed tracking and vehicle follower control

design for heavy-duty vehicles," *Vehicle System Dynamics*, vol. 25, no. 4, pp. 251–276, 1996.

[72] M. Druzhinina, L. Moklegaard, and A. G. Stefanopoulou, "Compression braking control for heavy-duty vehicles," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 4, 2000, pp. 2543–2547.

[73] D. Yanakiev and I. Kanellakopoulos, "Nonlinear spacing policies for automated heavy-duty vehicles," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 4, pp. 1365–1377, Nov 1998.

[74] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 566–580, May 2007.

[75] P. Sahlholm and K. H. Johansson, "Road grade estimation for look-ahead vehicle control using multiple measurement runs," *Control Engineering Practice*, vol. 18, no. 11, pp. 1328–1341, 2010.

[76] H. E. Tseng, L. Xu, and D. Hrovat, "Estimation of land vehicle roll and pitch angles," *Vehicle System Dynamics*, vol. 45, no. 5, pp. 433–443, 2007.

[77] H. Grip, L. Imsland, T. Johansen, J. Kalkkuhl, and A. Suissa, "Estimation of road inclination and bank angle in automotive vehicles," in *American Control Conference, 2009. ACC '09.*, June 2009, pp. 426–432.

[78] J.-O. Hahn, R. Rajamani, S.-H. You, and K. I. Lee, "Real-time identification of road-bank angle using differential GPS," *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 4, pp. 589–599, July 2004.

[79] H. S. Bae, J. Ryu, and J. C. Gerdes, "Road grade and vehicle parameter estimation for longitudinal control using GPS," in *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, 2001, pp. 166–171.

[80] J. Parviainen, J. Hautamäki, J. Collin, and J. Takala, "Barometer-aided road grade estimation," in *Proceedings of the World Congress of the International Association of Institutes of Navigation.*, 2009.

[81] G. Wahba, "A least squares estimate of satellite attitude," *SIAM Review*, vol. 7, no. 3, pp. 409–409, 1965.

[82] F. L. Markley, "Attitude determination using vector observations and the singular value decomposition," *The Journal of the Astronautical Sciences*, vol. 36, no. 3, pp. 245–258, 1988.

[83] M. Shuster, "Three-axis attitude determination from vector observations," *J. Guidance and Control*, vol. 4, no. 1, pp. 70–77, 1981.

[84] *Design Manual for Roads and Bridges, Highway Link Design*, 6th ed., The Highways Agency, Williams Lea, 2002.

[85] C. Mertz, S. Varadharajan, S. Jose, K. Sharma, L. Wander, and J. Wang, "City-wide road distress monitoring with smartphones," in *Proceedings of ITS World Congress*, September 2014.

[86] J. Ryu and J. Gerdes, "Estimation of vehicle roll and road bank angle," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 3, June 2004, pp. 2110–2115.

[87] F. Saeed, S. Qamariatul, M. Rahman, and A. Woodside, "The state of pothole management in UK local authority," *Bituminous Mixtures and Pavements VI*, pp. 153–159, 2015.

[88] H. Han, J. Yu, H. Zhu, Y. Chen, J. Yang, Y. Zhu, G. Xue, and M. Li, "SenSpeed: Sensing driving conditions to estimate vehicle speed in urban environments," in *INFOCOM, 2014 Proceedings IEEE*, April 2014, pp. 727–735.

[89] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The Pothole Patrol: Using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*. New York, NY, USA: ACM, 2008, pp. 29–39.

[90] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*. New York, NY, USA: ACM, 2008, pp. 323–336.

[91] K. Chen, M. Lu, G. Tan, and J. Wu, "CRSM: Crowdsourcing based road surface monitoring," in *Proc. of the 11th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC)*, November 2013, pp. 2151–2158.

[92] M. Perttunen, O. Mazhelis, F. Cong, M. Kauppila, T. Leppänen, J. Kantola, J. Collin, S. Pirttikangas, J. Haverinen, T. Ristaniemi, and J. Riekki, "Distributed road surface condition monitoring using mobile phones," in *Proceedings of the 8th International Conference on Ubiquitous Intelligence and Computing*, UIC'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 64–78.

[93] C. Koch and I. Brilakis, "Pothole detection in asphalt pavement images," *Adv. Eng. Inform.*, vol. 25, no. 3, pp. 507–515, Aug. 2011.

[94] J. Karuppuswamy, V. Selvaraj, M. M. Ganesh, and E. L. Hall, "Detection and avoidance of simulated potholes in autonomous vehicle navigation in an unstructured environment," *Proc. SPIE*, vol. 4197, pp. 70–80, 2000.

[95] M. Ausloos and D. H. Berman, "A multivariate Weierstrass-Mandelbrot function," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 400, no. 1819, pp. 331–350, 1985.

[96] J. J. Dawkins, "Terrain characterization and roughness estimation for simulation and control of unmanned ground vehicles," Ph.D. dissertation, Auburn University, Auburn, Alabama, December 2011.

[97] J. Wong, *Theory of Ground Vehicles*. John Wiley & Sons, 1993.

[98] Clifton Associates Ltd., "Pothole identification, assessment, and repair guidelines," Municipalities of Prince Albert, Saskatoon, Tech. Rep., August 2012.

[99] T. Strohmer, "Computationally attractive reconstruction of bandlimited images from irregular samples," *IEEE Transactions on Image Processing*, vol. 6, no. 4, pp. 540–

548, Apr 1997.

[100] W. Karush, "Minima of Functions of Several Variables with Inequalities as Side Constraints," Master's thesis, Dept.˜of Mathematics, Univ.˜of Chicago, 1939.

[101] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950*. University of California Press, Berkeley and Los Angeles, 1951, pp. 481–492.

[102] "Weather underground," https://www.wunderground.com/.