Developments in Estimation and Control for Cloud-Enabled Automotive Vehicles

by

Zhaojian Li

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
2016

Doctoral Committee:

Associate Professor Ella M. Atkins, Co-Chair
Professor Ilya V. Kolmanovsky, Co-Chair
Assistant Professor James R. Forbes
Professor Jing Sun

A journey of a thousand miles begins with a single step

—Laozi, *Tao Te Ching*

To my parents,
Yucai and Yongfan,
with gratitude and love

ACKNOWLEDGMENTS

Firstly and most importantly, I would like to sincerely thank my advisors, Prof. Atkins and Prof. Kolmanovsky, for their guidance and support, for giving me the opportunity to do research with flexibility, and for their mentoring and encouragement that helped me overcome challenges and difficulties. I am privileged to work with them and to have the opportunities to learn from their brilliance, dedication and kindness. They have offered the best I can expect.

Secondly, I would like to thank my committee members, Prof. Sun and Prof. Forbes, for their insightful comments and constructive suggestions throughout my dissertation writing. I would like to thank my Ford advisors, Jianbo Lu and Dimitar Filev, for their guidance and help, and for enabling me to work on both theoretical and applied research. I would like to thank Ford-University of Michigan Alliance for supporting my Ph.D. program.

I would like to acknowledge all of my co-authors and co-workers, who have helped me in accomplishing my research goals. From Ford: Kun, Quansheng, Xingping, Hamid, Baitao, Steve, John, Mahmoud and Engin; from the University: Rohit, Uros, Wu, Dianyun, Kevin, Xun, Swee, Pedro, Ehsan and Hossein.

I would also like to thank my best friends, Xiang, Cao, Xunyuan, Tiezhi, Liang, Pengchuan, Connie, Yuchen and Xianjun, for their support and encouragement.

Finally, the biggest thanks to my family. I heartfully thank my wife, Cheng, for her love, support and sacrifice. Thank my son, Lucas, for the extra motivation and inspiration he brought to me.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGMENTS iii

LIST OF FIGURES viii

LIST OF TABLES xi

LIST OF ABBREVIATIONS xii

LIST OF SYMBOLS xiv

ABSTRACT xv

CHAPTER

1. Introduction 1

1.1 Motivation 1
1.2 Overview of V2C2V 3
1.3 Comparison with Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) 5
1.4 Research Objectives and Approach 6
1.5 Contributions and dissertation outline 7
1.6 Innovations 9

Part I Road profile and anomaly information crowdsourcing

2. Pothole response modeling and detection 10

2.1 Introduction 10
2.2 Pothole analysis and a multi-phase dynamic model 11
2.2.1 Pothole characterization 11
2.2.2 Multi-phase dynamic model 13
2.3 Model validation with FTire 19
2.4 Model-based pothole detection 22

2.4.1 Mode confidence . . . 22
2.4.2 Estimation with Unscented Kalman Filter (UKF) . 22
2.4.3 Detection example . . . 23
2.5 Summary and discussion . . . 24

3. Simultaneous road profile estimation and anomaly detection 26

3.1 Introduction . . . 26
3.2 Background . . . 27
3.2.1 Dynamic model . . . 27
3.2.2 Input observer . . . 30
3.3 JDP-based estimation and comparison with the Kalman filter (KF) . . . 31
3.3.1 JDP-based estimation . . . 31
3.3.2 Simulation results . . . 34
3.4 Anomaly detection algorithm and experimental results . . . 36
3.4.1 Road anomaly detection algorithm . . . 36
3.4.2 Experimental results . . . 38
3.5 Summary and discussion . . . 41

4. Cloud-enabled anomaly reports clustering . . . 43

4.1 Introduction . . . 43
4.2 Mahalanobis distance and χ^2 distribution . . . 46
4.2.1 Math preliminaries . . . 47
4.3 Anomaly Report Stream Clustering Algorithm (ARSCA) . . 47
4.3.1 Cluster features . . . 48
4.3.2 Cluster maintenance algorithm . . . 50
4.3.3 Recursive computation of matrix inverse . . . 53
4.3.4 Parameter selection discussion . . . 53
4.4 Simulation demonstration . . . 54
4.5 Summary and discussion . . . 58

Part II Cloud-aided automotive applications

5. Cloud-aided semi-active suspension control . . . 59

5.1 Introduction . . . 59
5.2 Suspension dynamic model and road input characterization . 61
5.2.1 Dynamic model of semi-active suspension systems . 61
5.2.2 Road input modeling . . . 63
5.2.3 Pothole response modeling . . . 64
5.3 Performance prediction and mode selection . . . 66
5.4 Towards V2C2V implementation . . . 69
5.5 Simulation results . . . 72

5.6 Summary and discussion 73

6. Cloud-aided safety-based route planning 74

6.1 Introduction 74
6.2 Road risk modeling 78
6.2.1 Road risk index 78
6.2.2 HSIS database 79
6.2.3 Crash rate prediction 81
6.3 Model sensitivity analysis 85
6.4 Real-time factors influencing road risk 87
6.4.1 Weather 87
6.4.2 Time of day, day of the week 88
6.4.3 Driving style and vehicle conditions 89
6.5 Safety based route planning 89
6.5.1 Problem Formulation 90
6.5.2 Optimal route planning by Mixed Integer Programming 91
6.6 Route planning case study 91
6.6.1 Time optimal route 92
6.6.2 $\alpha = 0.2$ 92
6.6.3 Route changes with time of day, day of the week . . 92
6.6.4 Route changes with weather conditions 93
6.7 Summary and discussion 94

7. Cloud-aided comfort-based route planning 97

7.1 Introduction 97
7.2 Comfort-related factors and metrics for route planning 100
7.2.1 Comfort-related vibration and acceleration factors . 100
7.2.2 Comfort metrics for route planning 103
7.3 Comfort-based route planning 106
7.3.1 Problem Formulation 106
7.3.2 Optimal route planning by Extended Dijkstra's algorithm 107
7.4 Route planning case study 107
7.4.1 Time optimal route 110
7.4.2 $\alpha_1 = 0.06$ case 111
7.4.3 $\alpha_1 = 0.1$ case 111
7.4.4 Routes with minimum anomaly impact 111
7.5 Summary and discussion 112
7.6 Acknowledgement 113

8. Conclusions and future work 114

8.1 Summary 114

8.2 Conclusions 114
8.2.1 Benefits of V2C2V 114
8.2.2 Pothole response modeling and detection 115
8.2.3 Simultaneous road profile estimation and anomaly detection 115
8.2.4 Cloud-enabled anomaly reports clustering 116
8.2.5 Cloud-aided semi-active suspension control 116
8.2.6 Cloud-aided safety-based route planning 116
8.2.7 Cloud-aided comfort-based route planning 117
8.3 Future work 117
8.3.1 Pothole response modeling and detection 117
8.3.2 Simultaneous road profile estimation and anomaly detection 118
8.3.3 Cloud-enabled anomaly reports clustering 118
8.3.4 Cloud-aided semi-active suspension control 118
8.3.5 Cloud-aided safety-based route planning 119
8.3.6 Cloud-aided comfort-based route planning 119

APPENDICES 120

BIBLIOGRAPHY 122

LIST OF FIGURES

Figure

1.1	Main ECUs in a modern automobile	2
1.2	A schematic V2C2V architecture diagram	3
1.3	A schematic V2V diagram [1]	5
2.1	Small pothole geometry	12
2.2	Large pothole geometry	13
2.3	Dynamic model of running on flat ground	14
2.4	Dynamic model of running about front edge	15
2.5	Dynamic model of hitting and running about rear edge	16
2.6	A typical relation between slip ratio and friction coefficient	16
2.7	Switching diagram of hitting a small pothole	19
2.8	Simulink setup with FTire	20
2.9	Responses of multi-phase model vs. FTire: $l = 0.25\,m$, $r = 0.05\,m$, $v = 30\,mph$	20
2.10	Responses of multi-phase model vs. FTire: $l = 0.15\,m$, $r = 0$, $v = 30\,mph$	21
2.11	Responses of multi-phase model vs. FTire: $l = 0.25\,m$, $r = 0$, $v = 40\,mph$	21
2.12	Estimate $p(y_k\|m^j, Y_{k-1})$ with UKF	23
2.13	Fictitious vertical acceleration measurements from FTire	24
2.14	Mode confidence scores	24
3.1	Seven DoF full-car model	28
3.2	2 DoF half-car model	29
3.3	Test vehicle and main available sensors	29
3.4	Input estimation with the JDP estimator	35
3.5	State estimation comparison with KF and JDP estimator	35
3.6	Input estimation with the KF	36
3.7	Input estimation with the JDP estimator	37
3.8	State estimation error comparison with KF and JDP estimator	37
3.9	Input estimation with the KF	38
3.10	Three road surface types; roughness increases from left to right	39
3.11	Input estimation on the three road surfaces	40
3.12	Real-time dSPACE ControlDesk interface	40

3.13 Input estimation when driving over a small pothole followed by a large pothole . . . 41
3.14 Detected anomalies in Dearborn with our test vehicle . . . 42
4.1 Vehicle-to-Cloud-to-Vehicle anomaly detection and information sharing 45
4.2 Clusters in forms of ellipsoids . . . 49
4.3 Road anomalies and false alarms . . . 55
4.4 Snapshot at the end of day 1 . . . 55
4.5 Information compression for anomalies 1, 2 and 3 . . . 56
4.6 Snapshot at the end of day 5 . . . 56
4.7 Snapshot at the end of day 10 . . . 57
4.8 Snapshot at the end of day 15 . . . 57
5.1 V2C2V Suspension Control System. . . . 61
5.2 Semi-active suspension dynamics. . . . 62
5.3 Pothole analysis. . . . 64
5.4 Kinematics when hitting the rear edge. . . . 66
5.5 Road disturbances ($\dot{r}_0 + w$). . . . 72
6.1 Architecture of cloud-aided safety-based route planning . . . 76
6.2 Safety-based route planning overview . . . 77
6.3 The function $\rho_\tau(x)$ in 6.6 . . . 82
6.4 Neural network for crash rate prediction . . . 83
6.5 A hybrid neural network model . . . 84
6.6 RRI dependence on roughness, speed limit and segment length . . . 85
6.7 RRI dependence on number of lanes, AADT and lane width . . . 85
6.8 RRI dependence on curvature and grade . . . 86
6.9 Edge RRI composition of multiple homogeneous road segments . . . 87
6.10 Accident distribution over day of the week . . . 89
6.11 Accident distribution over time of day . . . 89
6.12 AADT distribution over time of day (1996-2006) [2] . . . 90
6.13 Optimal routes with different α's . . . 93
6.14 Abstracted road maps . . . 94
6.15 Route changes with real-time RRI adjustments . . . 95
7.1 Architecture of cloud-aided comfort-based route planning . . . 100
7.2 Three road surface types; roughness increases from left to right. The roughness indices (RI) are 0.62, 0.85 and 1.03, respectively. . . . 101
7.3 Road anomaly types . . . 102
7.4 Left and right turns at intersections . . . 102
7.5 An example Sigmoid function (7.3) with $a = 0.8$, $b = 1.2$, $x_0 = 0.08$, and $k = 80$ . . . 104
7.6 Route planning from Ford Research and Innovation Center to Ford Rouge Factory Tour. A time-optimal route is highlighted on Google Maps. . . . 108
7.7 Abstracted road network map; each edge is assigned a cost triple (expected travel time, roughness index and road anomaly cost) . . . 108
7.8 Trace of the experimental drive and detected anomalies . . . 109

7.9 Abstracted road network map; each edge is labeled by a tuple (RWT, RAC) 110
7.10 Optimal comfort route with $\alpha_1 = 0.06$ 111
7.11 Optimal comfort route with $\alpha_1 = 0.1$ 112
7.12 Route with minimum anomaly impact ($\alpha_1 = 10$) 112

LIST OF TABLES

Table

3.1	JDP estimator parameters	34
4.1	Cross-reference table of p-value, CI and sigma band for $n = 1$, and χ^2 values for $n = 1, 2, \cdot, 6$	47
4.2	Parameters for simulation	55
5.1	Simulation parameters	72
5.2	Damping Coefficients C_s (N$\cdot$ s/m)	73
6.1	Explanatory and response variable statistics	81
6.2	Model performance comparison	84
6.3	Average daily accidents under different weather conditions	88
7.1	Driver comfort metric questionnaire for road anomalies	104
7.2	Driver comfort metric questionnaire for intersections	105

LIST OF ABBREVIATIONS

NIST	National Institute of Standards and Technology
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
ECU	Electronic Control Unit
V2C2V	Vehicle-to-Cloud-to-Vehicle
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
UKF	Unscented Kalman Filter
JDP	Jump Diffusion Process
HSIS	Highway Safety Information System
RRI	Road Risk Index
MIP	Mixed Integer Programming
PID	Proportional-Integral-Derivative
OLS	Ordinary Least Squares
QR	Quantile Regression
PR	Poisson Regression
ANN	Artificial Neural Network
GPS	Global Positioning System
EoM	Equations of Motion

DoF	Degree of Freedom
CG	Center of Gravity
KF	Kalman Filter
CI	Confidence Interval
US	United States
VIN	Vehicle Identification Number
IRI	International Roughness Index
mph	mile per hour
RMSE	Root Mean Square Error
NVH	Noise, Vibration and Harshness
BS	British Standard
ISO	International Organization for Standardization
PSD	Power Spectral Density
VDV	Vibration Does Values
RWT	Roughness-Weighted Time
IIC	Intersection-Induced Cost
RAC	Road Anomaly Cost
LP	Left Pothole
SP	Small Pothole
SB	Speed Bump
RJ	Road Joint
LT	Left Turn
RT	Right Turn
SG	Stop and Go
GS	Go Straight

LIST OF SYMBOLS

$\mathbb{R}$: Set of real numbers

$\mathbb{R}^n$: Set of n-dimensional vectors of real elements

$\mathbb{R}^{n\times m}$: Set of n-by-m real matrices

I_m : Identity matrix of dimension m

0_m : Zero square matrix of dimension m

Let $x \in \mathbb{R}^n$,

$\|x\|_p$: $p-$norm, *i.e.*, $\|x\|_p^p = \sum_{i=1}^{n} |x_i|^p$ for $1 \leq p < \infty$

$\|x\|_Q$: $Q-$norm, *i.e.*, $\|x\|_Q^2 = \frac{1}{2} x^{\mathrm{T}} Q x$ for positive-definite $Q \in \mathbb{R}^{n\times n}$

Let S be a set,

$|S|$: Cardinality of set S

ABSTRACT

Developments in Estimation and Control for Cloud-Enabled Automotive Vehicles

by

Zhaojian Li

Co-Chairs: Ella M. Atkins and Ilya V. Kolmanovsky

Cloud computing is revolutionizing access to distributed information and computing resources that can facilitate future data and computation intensive vehicular control functions and improve vehicle driving comfort and safety. This dissertation investigates several potential Vehicle-to-Cloud-to-Vehicle (V2C2V) applications that can enhance vehicle control and enable additional functionalities by integrating onboard and cloud resources.

Firstly, this thesis demonstrates that onboard vehicle sensors can be used to sense road profiles and detect anomalies. This information can be shared with other vehicles and transportation authorities within a V2C2V framework. The response of hitting a pothole is characterized by a multi-phase dynamic model which is validated by comparing simulation results with a higher-fidelity commercial modeling package. A novel framework of simultaneous road profile estimation and anomaly detection is developed by combining a jump diffusion process (JDP)-based estimator and a multi-input observer. The performance of this scheme is evaluated in an experimental vehicle. In addition, a new clustering algorithm is developed to compress anomaly information by processing anomaly report streams.

Secondly, a cloud-aided semi-active suspension control problem is studied demonstrating for the first time that road profile information and noise statistics from the cloud can be used to enhance suspension control. The problem of selecting an optimal

damping mode from a finite set of damping modes is considered and the best mode is selected based on performance prediction on the cloud.

Finally, a cloud-aided multi-metric route planner is investigated in which safety and comfort metrics augment traditional planning metrics such as time, distance, and fuel economy. The safety metric is developed by processing a comprehensive road and crash database while the comfort metric integrates road roughness and anomalies. These metrics and a planning algorithm can be implemented on the cloud to realize the multi-metric route planning. Real-world case studies are presented. The main contribution of this part of the dissertation is in demonstrating the feasibility and benefits of enhancing the existing route planning algorithms with safety and comfort metrics.

CHAPTER 1

Introduction

1.1 Motivation

Cloud computing has become an indispensable part of our everyday life. Gmail, Google Drive, iCloud and Facebook are all examples of cloud computing applications. Unlike traditional desktop computing that runs software and stores documents locally, cloud computing is an internet-based service that runs programs and stores data on a shared pool of resources connected to the global network. This pervasive network provides ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [3].

Three cloud computing service models are defined by the National Institute of Standards and Technology (NIST): Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (Baas) [3]. SaaS refers to application-based services that are accessible from client devices such as a web browser or a program interface. PaaS provides customers the capability to develop and deploy applications using programming languages, libraries, and tools, but customers do not control or manage the underlying cloud infrastructure. Baas provides consumers the ability to provision processing, storage, and network resources for the consumer to deploy and run arbitrary software.

Interest in employing cloud computing for automotive applications is growing to support computation and data intensive tasks [4, 5, 6, 7, 8]. The cloud can provide access to "big data" as well as real-time crowd-sourced information. Smart utilization of on-demand cloud resources can increase situation awareness and provide additional functionality. In addition, computation and data intensive tasks can be outsourced to the cloud, enabling advanced and high-fidelity algorithms to be implemented in

real time. While embedded vehicle processors remain essential for time-critical applications, cloud computing can extend current control functionalities with additional functions to enhance performance [4].

The modern automobile hosts a large number of embedded Electronic Control Units (ECUs) for control and diagnostic functions [9]. The main ECUs in an automobile are shown in Figure 1.1. These ECUs include sophisticated sensors and micro-controllers that can be used not only for traditional engine, transmission, and vehicle electronic stability control but also for road and environmental sensing in real time, *e.g.,* traffic density estimation [10], road friction coefficient estimation [11] and pothole detection [12]. Cloud-enabled vehicles can thus be used as mobile sensors to crowd-source road and environment data that can be shared in real time [7].

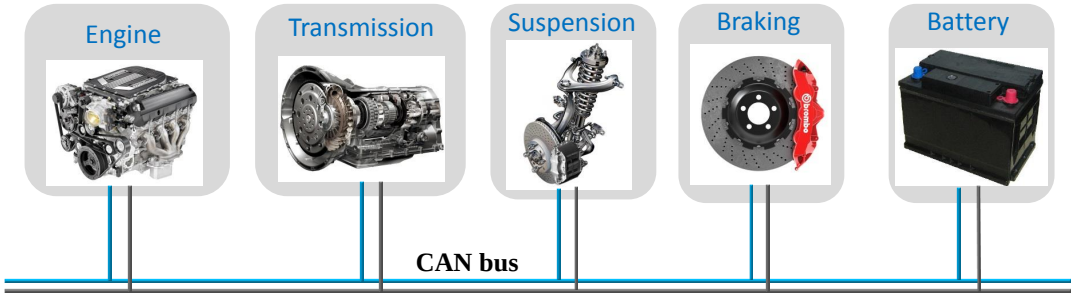


Figure 1.1: Main ECUs in a modern automobile

Cloud-aided automobile applications have been limited to-date. Insurance companies, specifically Progressive and StateFarm, use, with permission, MyRate and In-drive driving-monitoring devices, respectively, to determine driver insurance premium [13, 14]. MyFord Mobile utilizes cloud services to provide infotainment and telematics features [15]. In [16], cloud-based driving speed optimization is studied to improve fuel efficiency for everyday driving. An agent-based urban traffic management system is proposed in [17]. A cloud-based adaptive fleet vehicle routing system is proposed in [18, 19]. A system to complement vehicle sensor measurements with weather station data is considered in [20].

This thesis investigates several potential cloud-aided automotive applications that can enhance vehicle control and enrich vehicle functionality by integrating cloud and onboard resources. These applications are proposed within a Vehicle-to-Cloud-to-Vehicle (V2C2V) [4] framework.

1.2 Overview of V2C2V

A schematic diagram of the V2C2V architecture is illustrated in Figure 1.2. Cloud resources including computation, storage, and web services are pooled and managed by a cloud infrastructure managing system, enabling V2C2V applications to access these resources and augment onboard ECU capabilities. Onboard ECUs support real-time sensing, control, and diagnostic functions for time and safety critical applications.

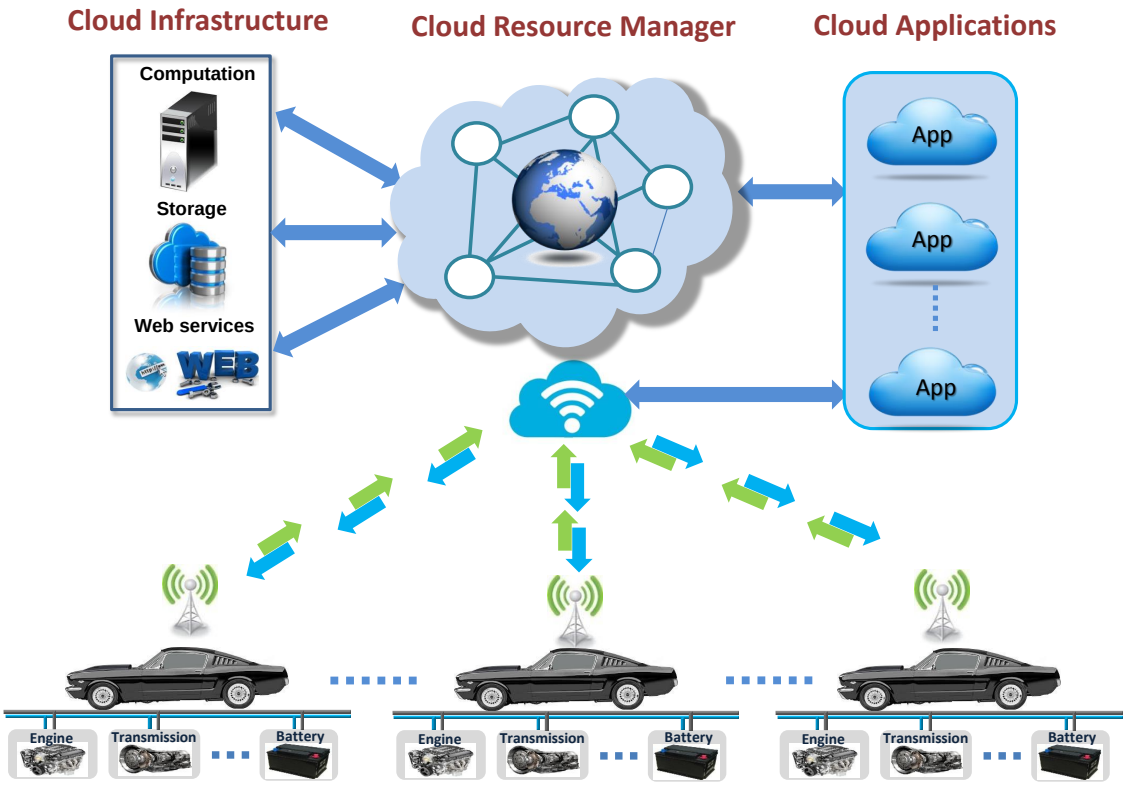


Figure 1.2: A schematic V2C2V architecture diagram

Onboard sensors and software provide the real-time data and processing necessary for safe vehicle control. Onboard sensors can also be used to sense real-time road and environment information [10, 11, 12]. Raw and processed data can be transmitted from the vehicle to the cloud to populate and update databases that aggregate information about vehicles, roads and drivers. Data fusion between onboard and cloud-based sources allows control computations to be performed at the optimal location (onboard vs. cloud) and maximizes situational awareness onboard and across the connected vehicle network. Real-time updates to cloud databases ensure

environment information (*e.g.*, road, weather) stays current.

Wireless networks support the communication between vehicles and cloud. Cloud-enabled vehicles can access web services for enhanced infotainment as well as travel planning and driving tasks. While the cloud can in principle provide unlimited data access and computing power, its use is limited by network availability, bandwidth, and cost. Network-induced imperfections such as time delays and packet dropouts must be carefully considered [21, 22] to ensure V2C2V systems do not reduce safety when the cloud is unavailable.

The main benefits of the V2C2V framework are summarized as follows:

- *Computation outsourcing.* The V2C2V framework enables vehicles to access powerful computing resources on the cloud to execute computation-intensive algorithms. For instance, the cloud implementation can facilitate optimal multi-objective route planning with route segment dependent mode or speed optimization. Onboard real-time controllers can display or in the case of future autonomous vehicles follow prescribed routes based on cloud data.

- *Information enrichment.* Cloud-enabled vehicles can access "big data" and real-time information through web services. These data can enhance situational awareness, improve controller performance, and enhance occupant infotaiment.

- *Data crowd-sourcing.* Modern vehicles are embedded with a suite of sophisticated sensors that can be exploited to characterize road, driver and environmental properties. Sensed information can be transmitted to the cloud to populate and update databases. Crowd-sourced data can then be shared with other vehicles and interested parties.

- *Connected vehicle.* The V2C2V-implemented vehicles are connected via the cloud. They can communicate useful information with each other in real-time, using the cloud as a media. The in-vehicle communication can be used to enhance cruise control and fleet management.

- *Diagnostics and prognostics.* Cloud-based diagnostic and prognostic prediction models can be used to infer vehicle degradation trends from onboard sensors. This enables vehicles to optimally schedule maintenance actions.

1.3 Comparison with Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I)

Complementary architectures to V2C2V have been developed, including V2V and V2I. V2V refers to a wireless network where automobiles can communicate with neighboring vehicles as shown in Figure 1.3. V2V communications exploit Dedicate Short-Range Communications (DSRC) with a link range up to 300 meters [23]. Vehicles share data including location, speed, steering and braking to nearby vehicles to warn drivers and enhance vehicle functions such as adaptive cruise control [24].

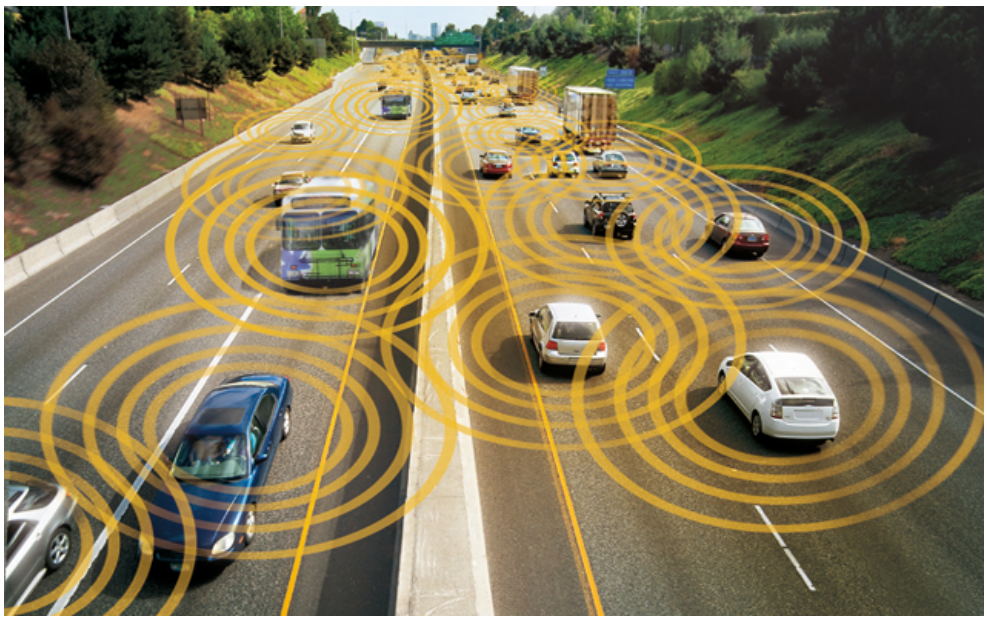


Figure 1.3: A schematic V2V diagram [1]

Similar to V2C2V, V2V enables vehicles data to be shared. However, there are defining differences. First, the main goals of V2V are to enable real-time and safety-critical warning and control functions while V2C2V has to-date focused on non-safety-critical applications due to communication reliability limitations. Second, vehicles with V2V technology are locally connected while V2C2V can connect vehicles globally via cloud media. Third, V2V-equipped vehicles typically share direct measurements from the CAN bus such as speed, steering and braking signals while V2C2V can employ onboard algorithms to process CAN bus data then send the processed data to the cloud. Finally, V2C2V offers access to significant computation and storage resources on the cloud.

Another vehicular communication framework is Vehicle-to-Infrastructure (V2I), where vehicles can communicate with roadway infrastructures such as traffic lights

to improve traffic safety and efficiency. V2I also relies on DSRC for communication. V2C2V can be seen as an augmented form of V2I with the cloud as an infrastructure.

1.4 Research Objectives and Approach

This thesis investigates potential automotive applications that can exploit the benefits of V2C2V. The fist application is motivated by the idea of crowd-sourcing mobile vehicle sensor data to characterize and update road condition information database on the cloud. Specifically, the problem of road profile estimation and anomaly detection from onboard sensors is studied.

Road anomalies such as potholes and bumps can cause discomfort for occupants and even vehicle damage. If available, road anomaly information can be used to warn drivers, enhance route planning, and notify road maintenance crews. This thesis proposes the exploitation of onboard vehicle sensors to detect anomalies autonomously. A multi-phase dynamic model is developed to characterize the response of hitting a pothole. This complex response is empirically decomposed into several phases, each of which corresponds to a simplified model. The developed model is validated by comparing simulation results with FTire, a commercial simulation package that supports tire-anomaly interaction analysis. This work has been published in a conference paper [25].

Based on a multi-phase model, a pothole detection algorithm is developed which combines an Unscented Kalman Filter (UKF) and Bayesian estimation. A simulation-based analysis demonstrates algorithm performance. The proposed detection scheme requires high-bandwidth sensors that are not commonly available in vehicles. To resolve this issue, a systematic framework of simultaneous road profile estimation and anomaly detection is developed in this thesis. A jump diffusion process (JDP)-based estimator is used to determine system states; the JDP-based estimator is compared with a Kalman filter when anomalies are present. A multi-input observer is used along with the JDP-based estimator to estimate the road profile. An anomaly detection algorithm is further developed to detect and label the anomalies (pothole, bump, road joints, *etc.*). The algorithm is implemented in a test vehicle to enable experimental validation. This work has appeared in [26].

With GPS and anomaly detection, cloud-enabled vehicles can report anomaly location data to the cloud. To aggregate reports, the cloud must convert the raw data stream into concise anomaly information that can be shared with other vehicles and road agencies. This thesis develops a clustering algorithm to achieve this goal.

The anomaly reports are processed in a single-pass fashion. Two cluster types, a main cluster and an outlier cluster, are defined to deal with outliers due to false alarms, as well as capture the evolving nature of road anomalies. Each cluster is defined by a feature vector and is maintained online. A simulation is presented to demonstrate the efficacy of the proposed algorithm. This work has appeared in [27].

The automotive suspension system is critical to ride comfort and road handling. A cloud-aided semi-active suspension control system is proposed to optimally select the best damping modes by exploiting the road information on the cloud as a preview. Road disturbance is modeled as a combination of a known road profile, an unmeasured stochastic road profile and potholes. Stochastic evaluation is conducted to select the best damping mode based on road profile and anomaly information from the cloud. This work has been published in conference papers [28, 21].

While modern vehicle navigation systems are able to generate optimal routes in terms of travel time, distance or fuel economy, this thesis next augments these traditional cost terms with safety and comfort metrics. A road and accident database from the Highway Safety Information System (HSIS) is mined to predict road risk indices (RRIs). Real-time factors such as time of day, day of the week and weather are also considered. A comfort metric is developed by integrating road roughness, road anomalies and intersections. Multi-metric route planning is formulated as a multi-objective network flow problem and further reduced to a Mixed Integer Programming (MIP) problem. A V2C2V implementation is proposed to facilitate access to real-time information and computing resources. Real world case studies are investigated. This work has appeared in [29, 30, 31].

1.5 Contributions and dissertation outline

This dissertation is divided into two parts. Part I investigates the use of vehicles to crowd-source road profile and anomaly data. Part II studies several cloud-aided automotive route planning and control applications. Individual contributions are listed below in the order of their corresponding chapters.

- A multi-phase dynamic model is developed to capture the complex response of hitting a pothole. A pothole detection algorithm is developed and demonstrated based on the multi-phase model using an Unscented Kalman Filter (UKF) and Bayesian estimation. These contributions are presented in Chapter 2 [25].

- A systematic framework of simultaneous road profile estimation and anomaly

detection is developed. The theory of JDP-based estimation is extended to the vector disturbance case as well as the case where the jump diffusion disturbance also impacts measurement data. The JDP-based estimation is applied to the road anomaly detection problem and evaluated in comparison to the Kalman Filter. The road profile estimation and anomaly detection algorithm is implemented and evaluated on an experimental vehicle. These contributions are detailed in Chapter 3 [26].

- An evolving clustering algorithm is developed to process aggregated anomaly reports. The scheme of o-cluster and m-cluster is exploited to reject outliers and handle the road anomaly changes over time. The developed algorithm can accurately determine isolated anomaly locations and can compress clustered anomaly information by processing pothole report stream. These contributions are presented in Chapter 4 [27].

- A cloud-aided semi-active suspension control system is developed by exploiting road profile data as a preview. The response of hitting potholes is explicitly considered, demonstrating how crowd-sourced road data and cloud computing can be used in an automotive control application. These contributions are detailed in Chapter 5 [28, 21].

- A cloud-aided safety-based route planner is investigated. A crash rate prediction model is developed with a hybrid Artificial Neural Network (ANN) and compared with existing models. Real-time factors such as weather, time of day and day of the week are incorporated as correction factors to a static road risk index. Real-world case studies are explored to demonstrate the applicability of the proposed route planning framework. These contributions are presented in Chapter 6 [29, 30].

- A cloud-aided comfort-based route planning problem is formulated as an extension or alternative to safety-based route planning. Objective comfort-related metrics are developed and applied in route planning, demonstrating how the crowd-sourced road profile and anomaly information can be used to improve information for drivers as well as enhancing control system performance. Real-world case studies are presented to demonstrate the efficacy of the proposed route planning framework. These contributions are detailed in Chapter 7 [31].

1.6 Innovations

Key innovations of this work include:

- A novel multi-phase dynamic model formulation to capture the complex response of hitting a pothole. Previous tire-anomaly interaction models such as Finite Element Analysis models are too computationally expensive for onboard algorithms.

- Simultaneous road profile estimation and anomaly detection using a JDP-based estimator and a multi-input observer. Previous methods focused on either road profile estimation or pothole detection. The existing anomaly detection algorithms exploited machine learning techniques.

- Use of road information from the cloud as a preview in semi-active suspension control. The effects of potholes are for the first time explicitly considered in the suspension control system. Previously the driver manually adjusted suspension or the road profile was assumed to be a Gaussian process.

- Use of a novel road risk index in safety-based route planning. Traditional route planner only considers time, distance or fuel economy.

- Development and use of comfort metrics including roughness weighted time, road anomaly cost, and intersection-induced cost. These comfort-related costs have not been considered in existing route planning framework.

Part I

Road profile and anomaly information crowdsourcing

CHAPTER 2

Pothole response modeling and detection

2.1 Introduction

Potholes can lead to passenger discomfort due to sudden acceleration change as well as vehicle damage such as tire puncture, suspension fatigue, and wheel misalignment. Pothole location information, if available, can be exploited to warn drivers and plan road maintenance by transportation authorities. It can also be exploited by suspension control systems as will be investigated in Chapter 5. This chapter analyzes the response of hitting a pothole and proposes an onboard pothole detector that is able to detect potholes with onboard sensors such as the speedometer and accelerometers. With a robust detector, pothole locations (*e.g.*, Global Positioning System (GPS) coordinates) can then be transmitted, processed and later shared through a Vehicle-to-Cloud-to-Vehicle (V2C2V) architecture [28, 29].

Recent studies have focused on pothole detection with mobile agents. Artis *et al.* [32] developed a mobile pothole detector with accelerometers embedded in Android smartphones. Four detection algorithms were proposed which exploited signal processing techniques and had an average true positive rate of around 80 percent. Jakob *et al.* [12] developed a pothole detector with three external accelerometers using machine learning techniques. The detector was implemented in seven taxis in metropolitan Boston and 39 out of 48 detected potholes were true potholes. Recently,

Jaguar and Land Rover has launched a "Pothole Alert" project to use onboard sensors to detect potholes [33].

In this chapter, we first develop a dynamic model to characterize the responses of hitting a pothole. Unlike previous approaches based on Finite Element Analysis (FEA) [34] which is computationally complex for onboard use, in this study, a novel multi-phase dynamic model is developed that represents a multi-mode switching system. Responses are empirically broken down into phases and each phase is represented by a simpler dynamic model. A pothole detection algorithm is then proposed based on the developed model which exploits Bayesian estimation and the Unscented Kalman Filter (UKF).

This chapter is organised as follows. The pothole characterization and a multi-phase dynamic model are discussed in Section 2.2. Section 2.3 presents model validation with FTire. A model-based pothole detection algorithm is developed and simulation results are reported in Section 2.4. Section 2.5 concludes the chapter.

2.2 Pothole analysis and a multi-phase dynamic model

In the literature on vehicle control with road disturbances, road inputs are often modeled as either deterministic (*e.g.*, given by sinusoids) or stochastic (*e.g.*, given by white noise) without an explicit representation for hitting a pothole [35, 36, 37]. In this chapter, we explicitly consider the response of hitting a pothole by exploiting a multi-phase dynamic model.

2.2.1 Pothole characterization

Before we develop the dynamic model, we first analyze pothole geometries on which the dynamic model is based. In this work, potholes are characterized as *small* or *large* based on whether the tire touches the bottom of the pothole when the vehicle runs over it. For small potholes the tire does not touch the bottom of the pothole. Geometry of the interaction between a wheel and a small pothole at the instant of hitting the rear edge is illustrated in Figure 2.1. According to the direction of motion (left to right), w_1 and w_2 are defined as the front and rear edges, respectively. The two main parameters for a small pothole are the diameter of the pothole l and the vertical difference between the rear and front edges of the pothole r. Note that in Figure 2.1 the height of the rear edge (right) is higher than that of the front edge. Similar analysis can be performed if the front edge is higher than the rear edge. Given

the wheel effective radius R, the following equations are satisfied:

$$\begin{aligned} l &= l_1 + l_2, \\ r &= \sqrt{R^2 - l_1^2} - \sqrt{R^2 - l_2^2}. \end{aligned} \tag{2.1}$$

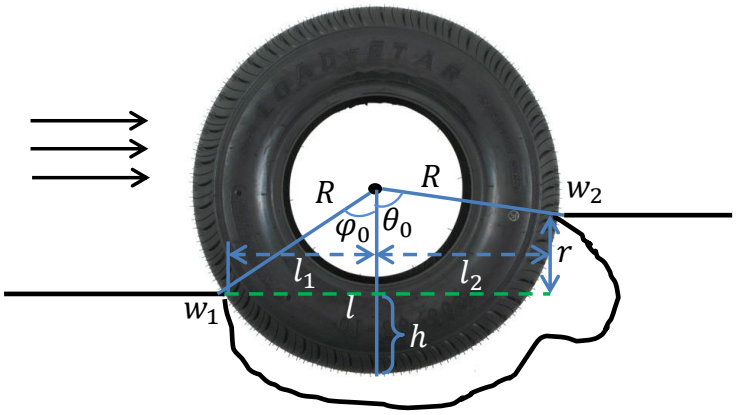


Figure 2.1: Small pothole geometry

From (2.1), l_1, l_2 can be determined if l, r and R are known. Then the angle φ_0 in Figure 2.1 can be computed as:

$$\varphi_0 = \sin^{-1} \frac{l_1}{R}. \tag{2.2}$$

Angle θ_0 in Figure 2.1 can be represented as:

$$\theta_0 = \sin^{-1} \frac{l_2}{R}. \tag{2.3}$$

Note that if the vertical difference between the front and rear edges r is zero, then

$$l_1 = l_2 = \frac{l}{2}.$$

By large potholes we refer to those with a large diameter such that the wheel hits the bottom of the pothole when running over it. As illustrated in Figure 2.2, the pothole is sufficiently wide that the wheel will touch the bottom of the pothole. In Figure 2.2, w_1, w_2, l, θ_0 and φ_0 are defined the same as for the small pothole; r_1 and r_2 are the heights relative to the bottom of the front and rear edges, respectively; d is the distance of the wheel travelling on the bottom of the pothole. These variables

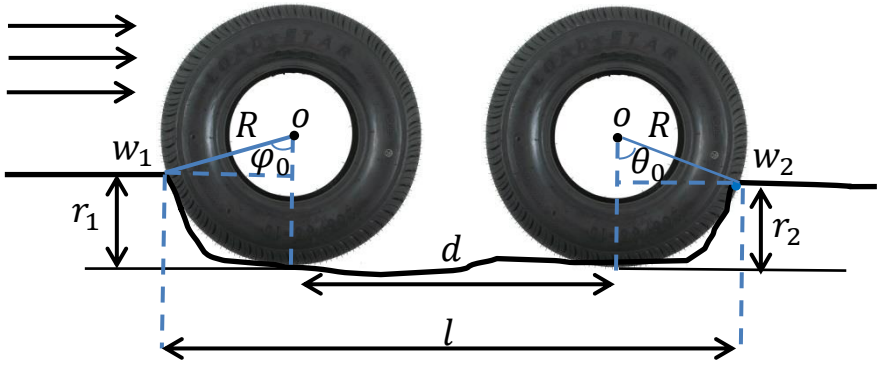


Figure 2.2: Large pothole geometry

are related as follows:

$$\varphi_0 = \cos^{-1}\frac{R - r_1}{R}, \tag{2.4a}$$

$$\theta_0 = \cos^{-1}\frac{R - r_2}{R}, \tag{2.4b}$$

$$d = l - \sin\varphi_0 \cdot R - \sin\theta_0 \cdot R. \tag{2.4c}$$

With the above variable descriptions, we next develop a dynamic model to represent the responses of a quarter car running over a pothole.

2.2.2 Multi-phase dynamic model

In this study, the overall response is empirically decomposed into four phases and each phase is modeled as a simple dynamic system. The phases are as follows:

1. *Rolling on flat ground.* The vehicle rolls on flat ground before hitting a pothole.

2. *Rolling about front edge w_1.* The wheel starts to hit the pothole by rolling about the front edge w_1 as in Figure 2.1. The wheel drops and w_1 is the only contact point in this phase. During this phase, the angle between Ow_1 and the reference vertical line changes from 0 to φ_0, which is represented in (2.2).

3. *Rolling about rear edge w_2.* The second phase ends when the wheel hits the rear edge of the pothole w_2. w_2 then becomes the only contact point and the wheel rolls about that point. During this phase, the angle between Ow_2 and the reference vertical line changes from θ_0 to 0.

4. *Rolling on flat ground.* After phase 3, the wheel rolls out of the pothole and proceeds on flat ground.

The responses of hitting a large pothole are similar to hitting a small one except that the wheel hits and rolls on the bottom after rolling about w_1 before hitting w_2 in the large pothole case. The response is also composed of the above basic phases. Since phase 1 and 4 are the same, there are 3 modes (*rolling on flat ground, rolling about front edge and hitting and rolling about rear edge*) corresponding to the distinct phases. The dynamic models are illustrated in Figures 2.3 and 2.5.

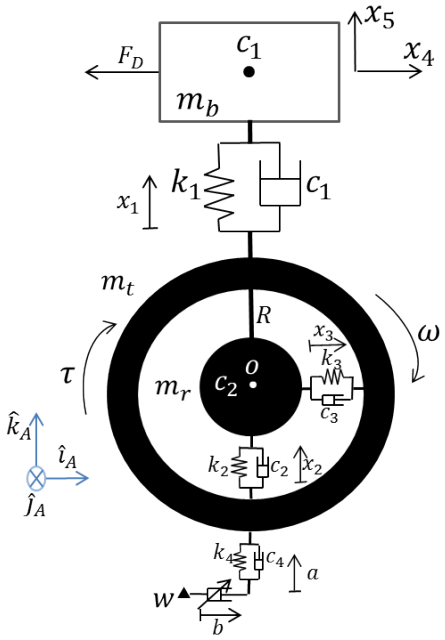


Figure 2.3: Dynamic model of running on flat ground

Since most pothole impacts involve only one wheel at a time, a quarter car model is used. The body of a quarter car is modeled as a rigid body with mass m_b. Vehicle suspension is modeled as a spring and damper system with constant spring stiffness k_1 and damping coefficient c_1. The point O (see Figures 2.3 – 2.5) is the center of the rim (wheel carrier) and we assume there is no horizontal displacement between the rim and the quarter car body. A simplified rigid ring tire model is used where the tire interacts with the rim through a vertical spring and damper system with stiffness k_2 and damping coefficient c_2, and a horizontal spring and damper system with stiffness k_3 and damping coefficient c_3. The interaction between the tire and road surface is modeled as a rotating spring and damper through the contact point and the center of wheel rim O with stiffness k_4 and damping coefficient c_4. Note that w_1 and w_2 are

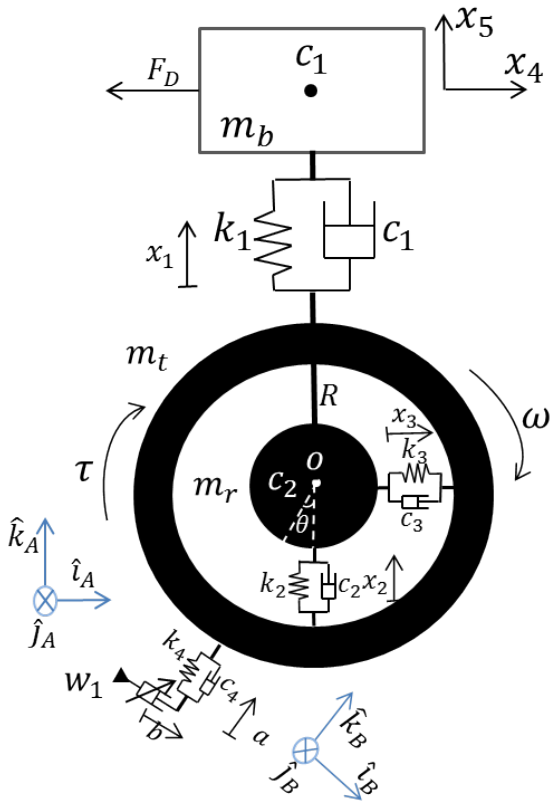


Figure 2.4: Dynamic model of running about front edge

the only contact points in phase 2 and phase 3, respectively. The angle θ is the angle between the vertical line and the line connecting center of rim O and the contact point during phase 2 and 3 as in Figures 2.4 and 2.5.

The variables x_1, x_2, x_3, and a are the displacements from equilibrium of springs k_1, k_2, k_3 and k_4, respectively. These displacements are defined to be positive when the corresponding spring is extended. The variables x_4 and x_5 are the horizontal and vertical displacements relative to an inertial point, respectively. The variable τ is the driving torque; b represents the tire-road slip displacement and $\dot{b}$ is the slip velocity. A commonly used slip-friction relation is illustrated in Figure 2.6, which is described by the following expression,

$$\mu(\kappa) = \frac{2\mu_p \kappa_p}{\kappa_p^2 + \kappa^2}\kappa, \tag{2.5}$$

where κ is the slip ratio defined as $\kappa = \frac{v_{sp}}{\max(v_t, \omega R)}$, v_{sp} is the slip velocity ($v_{sp} = \dot{b}$ in our case), and v_t is the wheel tangential velocity. Since we are considering the driving

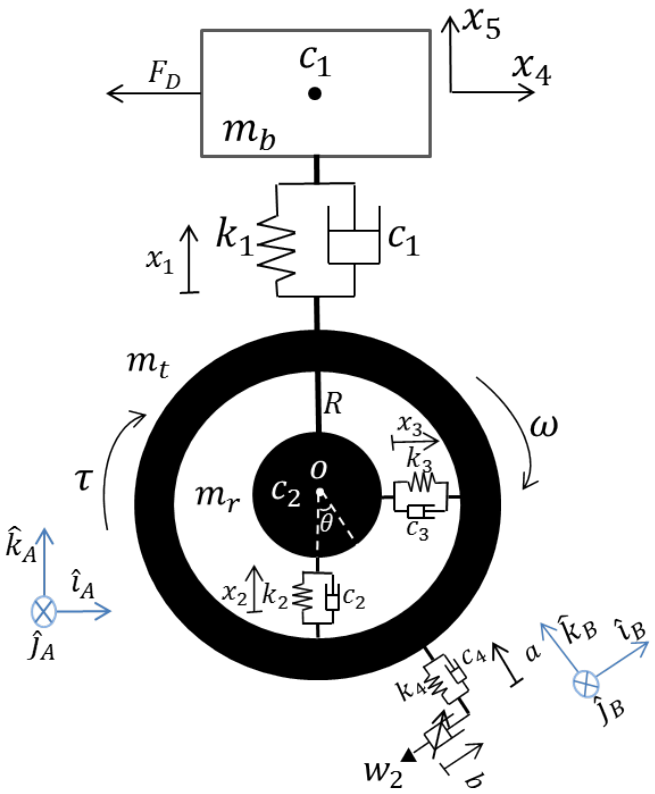


Figure 2.5: Dynamic model of hitting and running about rear edge

wheel, ωR is normally greater than v_t. As a result, our formulation yields

$$\kappa = \frac{\dot{b}}{\omega R}. \tag{2.6}$$

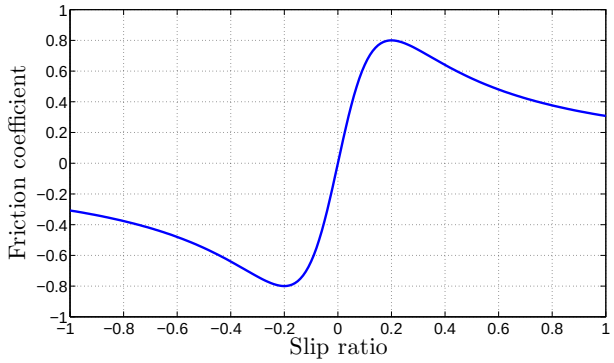


Figure 2.6: A typical relation between slip ratio and friction coefficient

In (2.5), μ_p is the peak friction coefficient that depends on the road surface and condition while κ_p is the slip ratio at which the maximum friction coefficient is obtained. In Figure 2.6, $\mu_p = 0.8$ and $\kappa_p = 0.2$. F_D is the longitudinal drag force acting on the quarter car and is modeled as a combination of aerodynamic drag proportional to the square of the speed, and a force due to the rear half car body, which we assume is proportional to the speed,

$$F_D = \frac{1}{2} C_{ad} v^2 + C_d v, \tag{2.7}$$

where C_{ad} is the aerodynamic drag coefficient (air density incorporated) assuming sea level density and C_d is a constant capturing the drag force due to the rear half car body. The variable v is the vehicle longitudinal speed.

In phases 2 and 3, we exploit two frames in the derivation of the equations of motion (EoMs). The first frame, F_A, is an inertial frame and the second frame, F_B, is attached to the wheel. The vectors i_A, j_A, k_A, i_B, j_B, and k_B are the basis vectors associated with Frames F_A and F_B. During phase 2, we have $F_A \xrightarrow[\hat{j}]{\theta} F_B$, which means F_B can be obtained by rotating F_A around eigenaxis $\hat{j}$ by an angle θ as in Figure 2.4 and we have $F_A \xrightarrow[\hat{j}]{-\theta} F_B$ during phase 3 as in Figure 2.5. Then EoMs for the three modes are derived using Newton's second law and are shown as follows.

1. EoM for mode 1 (phase 1 and 4):

$$\begin{aligned}
m_b \ddot{x}_5 &= -m_b g - k_1 x_1 - c_1 \dot{x}_1, \\
(m_b + m_r)\ddot{x}_4 &= k_3 x_3 + c_3 \dot{x}_3 - \frac{1}{2} C_{ad} \dot{x}_4^2 - C_d \dot{x}_4, \\
m_r(\ddot{x}_5 - \ddot{x}_1) &= -m_r g + k_1 x_1 + c_1 \dot{x}_1 - k_2 x_2 - c_2 \dot{x}_2, \\
m_t(\ddot{x}_5 - \ddot{x}_1 - \ddot{x}_2) &= -m_t g + k_2 x_2 + c_2 \dot{x}_2 - (k_4 a + c_4 \dot{a}), \\
m_t(\ddot{x}_3 + \ddot{x}_4) &= -k_3 x_3 - c_3 \dot{x}_3 + f_\mu, \\
I\ddot{\theta} &= \tau - f_\mu (R + a), \\
\dot{a} &= \dot{x}_5 - \dot{x}_1 - \dot{x}_2, \\
\dot{b} &= \dot{x}_3 + \dot{x}_4 - \dot{\theta}(R + a).
\end{aligned} \tag{2.8}$$

2. EoMs for mode 2 (phase 2):

$$
\begin{aligned}
m_b\ddot{x}_5 &= -m_b g - k_1 x_1 - c_1\dot{x}_1,\\
(m_b+m_r)\ddot{x}_4 &= k_3 x_3 + c_3\dot{x}_3 - \frac{1}{2}C_{ad}\dot{x}_4^2 - C_d\dot{x}_4,\\
m_r(\ddot{x}_5 - \ddot{x}_1) &= -m_r g + k_1 x_1 + c_1\dot{x}_1 - k_2 x_2 - c_2\dot{x}_2,\\
m_t(\ddot{x}_5 - \ddot{x}_1 - \ddot{x}_2) &= -m_t g + k_2 x_2 + c_2\dot{x}_2 - \cos\theta\cdot(k_4 a + c_4\dot{a}) - \sin\theta\cdot f_\mu,\\
m_t(\ddot{x}_3 + \ddot{x}_4) &= -k_3 x_3 - c_3\dot{x}_3 - \sin\theta\cdot(k_4 a + c_4\dot{a}) + \cos\theta\cdot f_\mu,\\
I\ddot{\theta} &= \tau - f_\mu(R+a),\\
\begin{bmatrix}\dot{a}\\ \dot{b}\end{bmatrix} &= \mathcal{R}_r^{-1}(\theta)\begin{bmatrix}-\dot{x}_1 - \dot{x}_2 + \dot{x}_5\\ \dot{x}_3 + \dot{x}_4\end{bmatrix} + \begin{bmatrix}\dot{\theta}b\\ -\dot{\theta}(R+a)\end{bmatrix},
\end{aligned}
\tag{2.9}
$$

where $\mathcal{R}_r(\theta) = \begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}$ is the planar rotation matrix.

3. EoMs for mode 3 (phase 3):

$$
\begin{aligned}
m_b\ddot{x}_5 &= -m_b g - k_1 x_1 - c_1\dot{x}_1,\\
(m_b+m_r)\ddot{x}_4 &= k_3 x_3 + c_3\dot{x}_3 - \frac{1}{2}C_{ad}\dot{x}_4^2 - C_d\dot{x}_4,\\
m_r(\ddot{x}_5 - \ddot{x}_1) &= -m_r g + k_1 x_1 + c_1\dot{x}_1 - k_2 x_2 - c_2\dot{x}_2,\\
m_t(\ddot{x}_5 - \ddot{x}_1 - \ddot{x}_2) &= -m_t g + k_2 x_2 + c_2\dot{x}_2 - \cos\theta\cdot(k_4 a + c_4\dot{a}) + \sin\theta\cdot f_\mu,\\
m_t(\ddot{x}_3 + \ddot{x}_4) &= -k_3 x_3 - c_3\dot{x}_3 + \sin\theta\cdot(k_4 a + c_4\dot{a}) + \cos\theta\cdot f_\mu,\\
I\ddot{\theta} &= \tau - f_\mu(R+a),\\
\begin{bmatrix}\dot{a}\\ \dot{b}\end{bmatrix} &= \mathcal{R}_r^{-1}(\theta)\begin{bmatrix}-\dot{x}_1 - \dot{x}_2 + \dot{x}_5\\ \dot{x}_3 + \dot{x}_4\end{bmatrix} + \begin{bmatrix}-\dot{\theta}b\\ \dot{\theta}(R+a)\end{bmatrix},
\end{aligned}
\tag{2.10}
$$

where I is inertia of the wheel and $\mathcal{R}_r(\theta) = \begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}$ is the planar rotation matrix.

The responses of hitting a pothole can then be described by a switched system for which the switching conditions are determined by the geometries described in (2.2)-(2.4). The switching diagram for hitting a small pothole is described in Figure 2.7. Before hitting the pothole, the quarter car is rolling on flat ground and the system is in mode 1. When the vehicle starts to contact the pothole, the system switches to mode 2 and the angle described in Figure 2.4 changes from 0 to φ_0. The system

switches to mode 3 when the wheel hits the rear edge of the pothole and the angle described in Figure 2.5 changes from θ_0 to 0. The vehicle then continues rolling on flat ground so it switches back to mode 1.

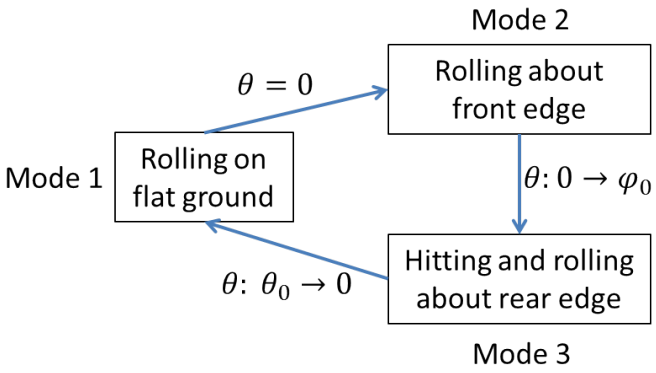


Figure 2.7: Switching diagram of hitting a small pothole

For simulation of the switched system, the *Events* option in MATLAB *ode45* is used to determine the switching conditions.

2.3 Model validation with FTire

The dynamics of hitting a pothole are modeled as a switched system described in Section 2.2. Before we exploit the developed model to implement pothole detection algorithms, model parameters are tuned and validated. Towards this end, we use FTire [38], a commercial software for tire-road interaction simulation with a *Flexible Ring tire* model. FTire can model tire responses to road anomalies such as potholes and speed bumps, providing benchmark to validate our model. The FTire simulation setup is illustrated in Figure 2.8. Tire type and pothole parameters can be specified in the FTire block set. The inputs of FTire are the three dimensional velocity of the rim center, wheel carrier orientation, driving torque and braking torque. The outputs of the FTire block include three dimensional tire forces, wheel angular velocity, tire deflection, *etc.* Since our model is two dimensional, to validate our model by comparing the simulation results, we feed longitudinal and vertical tire force components F_x, F_z into a suspension system with the same parameters as in the multi-phase model. Then the longitudinal and vertical velocity of the rim center v_x and v_z are fed back into FTire to close the loop.

Simulation results of the developed multi-phase model and FTire have been compared for various scenarios that include varying vehicle speed v, pothole diameter

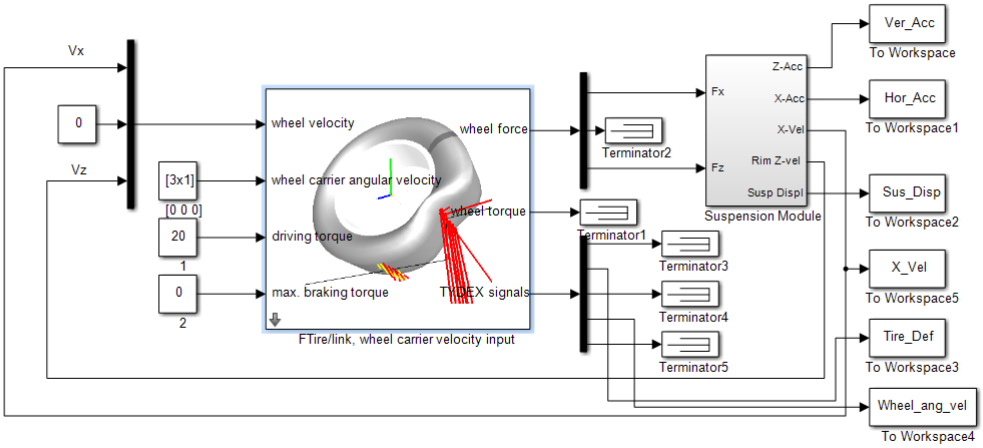


Figure 2.8: Simulink setup with FTire

l and front/rear edge height difference r. Simulation results of the two models are similar as shown in Figures 2.9 – 2.11. The wheel angular velocity and longitudinal

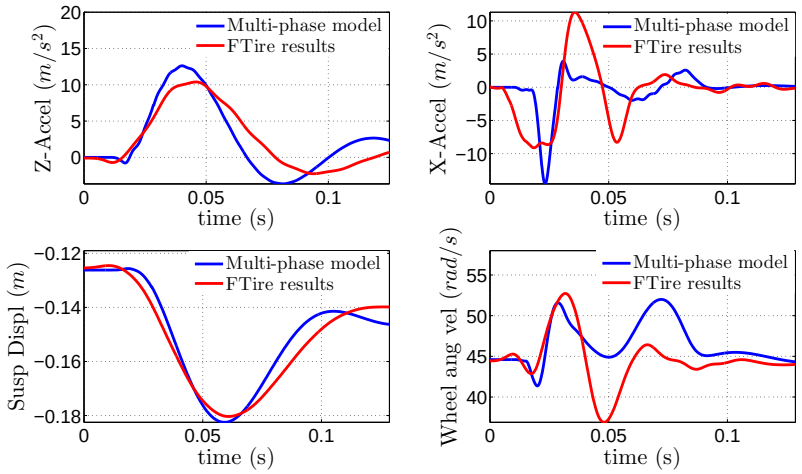


Figure 2.9: Responses of multi-phase model vs. FTire: $l = 0.25\,m$, $r = 0.05\,m$, $v = 30\,mph$

acceleration have more resonance in FTire than the multi-phase model response. The reason is that FTire, unlike our model, uses a Flexible Ring tire model, in which there is a rotational spring and damper between the rim and tire. Fortunately, the detection algorithm we develop below relies more on the trends than specific response, which are shown to be similar between the two models.

Remark 2.1. A physics-based multi-phase dynamic model is developed for the pur-

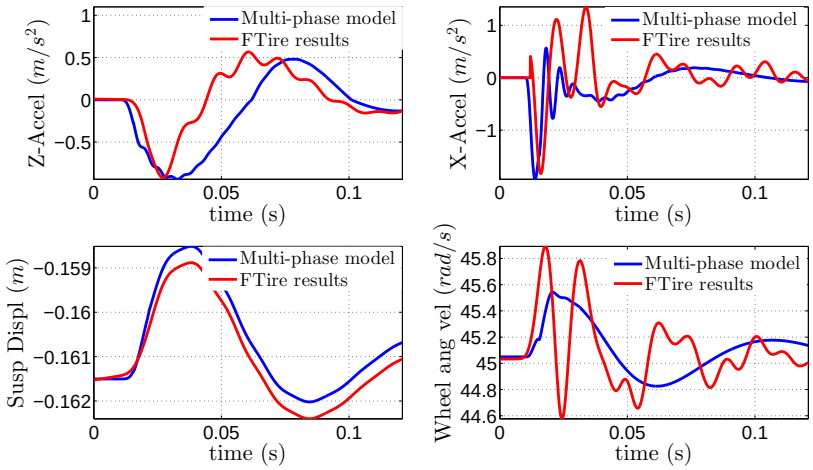


Figure 2.10: Responses of multi-phase model vs. FTire: $l = 0.15\,m$, $r = 0$, $v = 30\,mph$

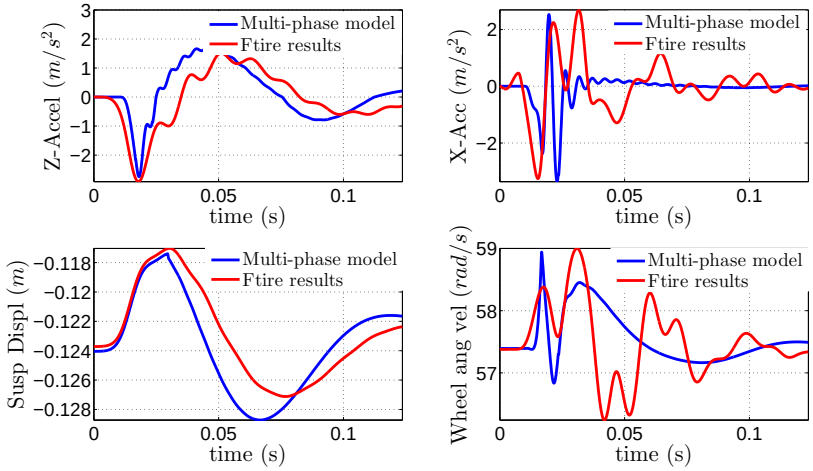


Figure 2.11: Responses of multi-phase model vs. FTire: $l = 0.25\,m$, $r = 0$, $v = 40\,mph$

pose of pothole detection. In this initial study, the multi-phase model is compared with FTire simulation package (as a benchmark) under three cases. Comprehensive comparisons should involve multiple runs (*e.g.*, 1000 runs) to obtain their differences statistically. Model parameters should also be further tuned to improve the model performance. More comprehensive model comparison and validation are considered as future work.

2.4 Model-based pothole detection

The multi-phase dynamic model developed in Section 2.2 and Section 2.3 is now exploited in a pothole detection algorithm.

2.4.1 Mode confidence

Based on the developed multi-phase model, hitting a pothole is modeled as a switched system with three sequential modes. The idea of the detection algorithm is to compare the inferred mode sequence with the expected mode sequence when hitting a pothole. By taking a sequence of measurements $Y_k = \{y_1, y_2, \cdots, y_k\}$ in an interval, the confidence of each mode is computed recursively based on Bayesian inference as follows,

$$\begin{aligned} p(m^j|Y_k) &= \frac{p(y_k|m^j, Y_{k-1})p(m^j|Y_{k-1})\cancel{p(Y_{k-1})}}{\sum_i p(y_k|m^i, Y_{k-1})p(m^i|Y_{k-1})\cancel{p(Y_{k-1})}} \\ &= \frac{p(y_k|m^j, Y_{k-1})p(m^j|Y_{k-1})}{\sum_i p(y_k|m^i, Y_{k-1})p(m^i|Y_{k-1})}, \end{aligned} \tag{2.11}$$

where m^j is the jth mode and $j = 1, 2, 3$. The mode with the highest confidence value is viewed as the true mode. The difficulty in applying (2.11) is in computing $p(y_k|m^j, Y_{k-1})$, the probability of a current measurement y_k given mode m^j and the previous observations Y_{k-1}. In this work, we apply the Unscented Kalman Filter (UKF) [39] to compute $p(y_k|m^j, Y_{k-1})$ due to its capacity for nonlinear estimation and relatively low computational complexity.

2.4.2 Estimation with Unscented Kalman Filter (UKF)

The computation of $p(y_k|m^j, Y_{k-1})$ using the UKF [39] is based on nonlinear model equations in each mode:

$$\dot{x} = f_i(x, \nu) + \mathbf{w}, \tag{2.12}$$

$$y = h_i(x, \nu) + \mathbf{v}, \tag{2.13}$$

where $x = [x_1,\ x_2,\ x_3,\ \theta,\ a,\ b,\ \dot{x}_1,\ \dot{x}_2,\ \dot{x}_3,\ \dot{x}_4,\ \dot{x}_5,\ \dot{\theta}]^{\mathrm{T}}$ is the state vector and $f_i(\cdot)$ is the system equation in mode i from (2.8)-(2.10); ν is the vector of the pothole parameters; h_i is the output function of mode i. $\mathbf{w}$ and $\mathbf{v}$ are the process noise and measurement noise, respectively. The UKF (see Figure 2.12) uses a deterministic sampling technique referred to as the unscented transform to pick a set of sigma points around the

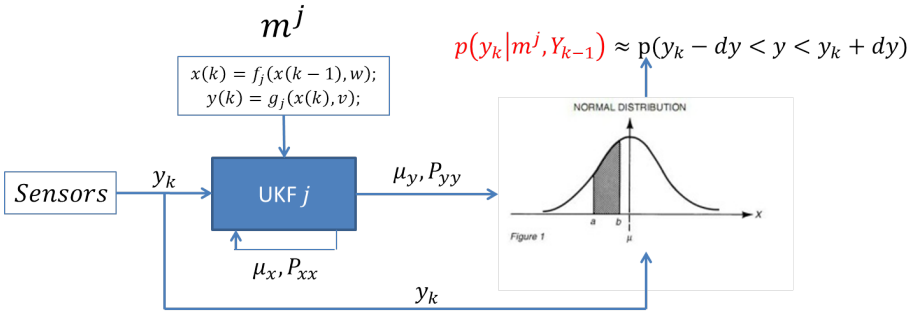


Figure 2.12: Estimate $p(y_k|m^j, Y_{k-1})$ with UKF

mean. These sigma points are then propagated through the system functions and the mean and covariance of the state distribution are estimated from these results. With measurements taken at each step, the UKF recursively computes state mean μ_x and covariance P_{xx} as well as output mean μ_y and covariance P_{yy}. Compared to its counterpart of extended Kalman filter [40], it avoids computations of Jacobian matrices that might cause numeric issues. Assuming that the output is Gaussian, the probability density function takes the following form,

$$F_y(y) = \frac{1}{\sqrt{(2\pi)^q \det P_{yy}}} \exp\Big(-\frac{1}{2}(y-\mu_y)^{\mathrm{T}} P_{yy}^{-1}(y-\mu_y)\Big), \tag{2.14}$$

where q is the dimension of measurement vector y. Then by specifying a small dy, $p(y_k|m^j, Y_{k-1})$ can be computed as

$$p(y_k|m^j, Y_{k-1}) \approx p(y_k - dy < y < y_k + dy) \approx 2F_y(y_k)dy. \tag{2.15}$$

2.4.3 Detection example

Detection relies on horizontal speed v_x, vertical acceleration a_z and wheel angular velocity ω, all of which are accessible measurements in modern vehicles from speedometer, z-axis accelerometer, and angular velocity measurement, respectively. Simulation results from FTire with added Gaussian noise are taken as fictitious measurements. Figure 2.13 shows the vertical acceleration simulation results from FTire with Gaussian noise. The duration of hitting the pothole is illustrated by the double arrow. The mode confidences are computed every 0.01 *sec* as in (2.11) and all initialized with 1/3 for the initial probability. The time interval is chosen to be 0.01 *sec* to represent a compromise between the fast dynamics and limited sensor bandwidth. The bandwidth of the embedded acceleration sensors typically ranges between 100 Hz and 380 Hz. The confidence score for every 0.01 s interval is illustrated in Fig-

ure 2.14. From the results, the second interval is dominated by mode 2 and the

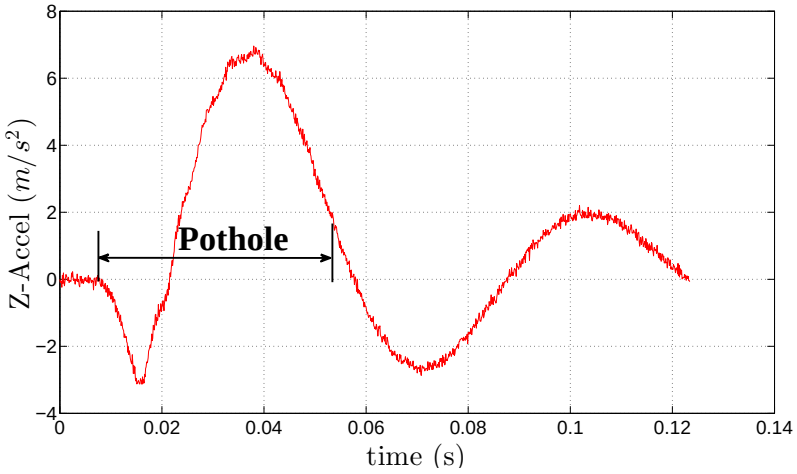


Figure 2.13: Fictitious vertical acceleration measurements from FTire

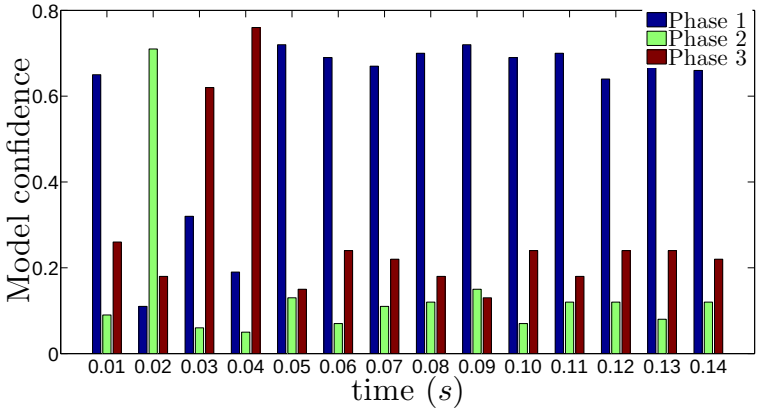


Figure 2.14: Mode confidence scores

following two intervals are dominated by mode 3. Since this sequence matches the formulated switching system in Figure 2.7, a pothole is successfully detected. Note that multiple tests should be performed to better evaluate the detection algorithm. However, due to FTire availability issues, we were only able to test a single case. More comprehensive evaluations should be performed in future work.

2.5 Summary and discussion

In this chapter, pothole characteristics were analysed and the dynamics of hitting a pothole by an automotive vehicle were broken down into phases. A multi-phase

dynamic model was built as a switched system with three modes to capture the responses of hitting a pothole. The developed model was validated by comparing results with FTire over varied scenarios. The results show that our multi-phase model was able to generate similar simulation results and trends. A model-based detection algorithm was developed that recursively computes the confidence of each mode with Bayesian estimation and the Unscented Kalman Filter.

A numerical simulation was presented to show the capability of the proposed detection algorithm. The confidence score of each mode was computed every 0.01 *sec* and each mode confidence computation interval should contain at least a few observations. As a result, the implementation requires sensors with high bandwidth (*e.g.,* $1K$ Hz) which may not be available in current vehicles, which has a bandwidth around 100 Hz. To resolve this problem, we develop a more practical detection algorithm in the next chapter.

CHAPTER 3

Simultaneous road profile estimation and anomaly detection

3.1 Introduction

Rough roads are undesirable for drivers due to the potential for ride discomfort, the need for decreased driving speed, and possible vehicle damage. Road anomalies such as potholes and speed bumps can make the ride even worse due to sudden accelerations which can lead to damage such as tire punctures and steering misalignment. If available, road profile and anomaly information can be used to improve route planning [31], enhance suspension control [28] and facilitate road maintenance. This information can also be seamlessly integrated with navigation systems for autonomous driving [41, 42].

Current road profile measures typically rely on sophisticated profilers [43] that may be expensive to acquire and operate [44]. Recently, inexpensive and easy-to-implement road profile estimation methods have been explored, as in [45] where a standalone vehicle accelerometer measures vehicle body acceleration in turn mapped to road power spectral density with a transfer function generated from a half-car model. Also, in [44], accelerometers in a smart phone were used to correlate vertical acceleration with road roughness. While these approaches are easy to implement, they are only able to classify road roughness level with no further road profile details.

Road anomaly detection has attracted attention in the past decade. Detection algorithms mainly fall into two categories: data-driven and model-based. Data-driven methods typically require extensive training data and are useful when models are not available. In [12], a pothole detector is developed with three external accelerometers and machine learning techniques are then exploited to train the system from collected data. Alternatively, model-based methods are able to deal with new and unforeseen

situations and are insensitive to unmeasured disturbances. However, they require an accurate physical model. In Chapter 2, a pothole detection algorithm was proposed by developing and exploiting a multi-phase dynamic model, but high-bandwidth sensors were required for successful identification.

In this chapter, we develop a systematic framework to simultaneously estimate road profile and detect road anomalies. A front half-car dynamic model is developed to represent the interactions between the vehicle and road disturbances. In the model, road velocity disturbances at the front two wheels are treated as inputs to the half car model and a multi-input observer developed in [46] is exploited to estimate the inputs. The input observer requires an estimate of system states. A jump diffusion process (JDP)-based estimator is developed to estimate these states.

As opposed to the Gaussian disturbance assumption in a Kalman filter, JDP-based estimation assumes the disturbance is a combination of Wiener process and Poisson process. This assumption can include rare but pronounced events such as potholes or bumps and it can also be used to represent wind gusts in other applications [47]. A JDP-based estimator was developed in [48] and we extend it to the vector disturbance case, as well as the case where the jump diffusion disturbance affects measurement channel data. We demonstrate that the JDP-based estimator outperforms a Kalman filter when jumps are present. The algorithm has been implemented in a Lincoln MKS test vehicle on which we demonstrate good estimation and detection performance.

This chapter is organized as follows. Section 3.2 develops the dynamic model and introduces the multi-input observer. Section 3.3 derives the JDP-based estimator and presents simulation results compared with a Kalman filter. Experimental results are presented in Section 3.4, followed by a summary and discussion in Section 3.5.

3.2 Background

3.2.1 Dynamic model

When vehicles travel on roads, they are perturbed by road disturbances. In suspension control applications, roadway vertical velocity disturbances are typically modeled as inputs to the suspension system [28, 37, 36]. If we can estimate these inputs, then the road velocity profile can be obtained and used for anomaly detection. We exploit the input observer from [46] to estimate road inputs.

To implement the input observer developed in [46], a dynamic model is required. We initially used the full-car model from Figure 3.1. The model has seven Degrees of Freedom (DoF) and fourteen states, a comprehensive model that can capture the

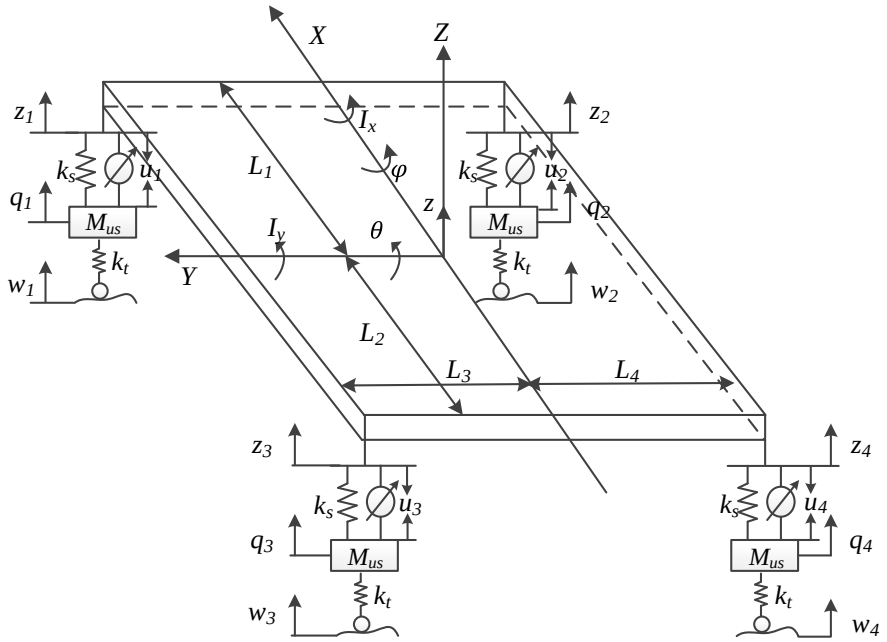


Figure 3.1: Seven DoF full-car model

disturbances at all four wheels. However, its use is impeded by the number of available sensors in our test vehicle, a 2012 Lincoln MKS, illustrated in Figure 3.3. In this vehicle, only four measurements (left front suspension deflection, left rear suspension deflection, vertical acceleration and roll rate) are available in the full-car model. Our simulation results reveal that some of the full-car model states can not be accurately estimated.

A reduced front half-car model, illustrated in Figure 3.2, is used as an alternative model for the design of the input observer. The front half car is modeled as a rigid body with mass m_b. I_x represents the moment of inertia about the longitudinal axis. z, z_1 and z_2 represent the vertical displacement of the center of gravity (CG), left body tip and right body tip, respectively, from equilibrium. The left and right tip-to-CG distances are denoted by L_1 and L_2, respectively. k_s and c_s, respectively, represent the spring stiffness and damping coefficient of the suspension system and we assume that the left and right side have the same parameters. We denote the roll angle by θ. q_1 and q_2 represent left and right suspension deflection from equilibrium, respectively. v_1 and v_2 are the road velocity inputs to the left and right wheel, respectively. Note that since the wheel is stiff, we approximate v_1 and v_2 as being directly applied to

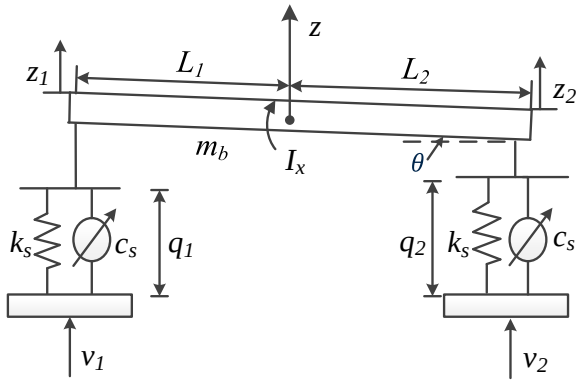


Figure 3.2: 2 DoF half-car model

Figure 3.3: Test vehicle and main available sensors

the left and right suspension, respectively.

Defining $x_1 = q_1$, $x_2 = q_2$, $x_3 = \dot{z}$ and $x_4 = \dot{\theta}$ as the states, we obtain the following equations of motion,

$$\begin{aligned}
\dot{x}_1 &= x_3 + L_1 x_4 - v_1, \\
\dot{x}_2 &= x_3 - L_2 x_4 - v_2, \\
\frac{1}{2} m_b \dot{x}_3 &= -k_s x_1 - c_s \dot{x}_1 - k_s x_2 - c_s \dot{x}_2, \\
\frac{1}{2} I_x \dot{x}_4 &= -L_1 k_s x_1 - L_1 c_s \dot{x}_1 + L_2 k_s x_2 + L_2 c_s \dot{x}_2.
\end{aligned} \tag{3.1}$$

The measurement vector y includes front left suspension deflection x_1, vertical accel-

eration $\ddot{z}$ and roll rate $\dot{\theta}$, *i.e.*,

$$y = [x_1 \quad \ddot{z} \quad \dot{\theta}]^{\mathrm{T}}. \tag{3.2}$$

Let $w = [v_1 \quad v_2]^{\mathrm{T}}$ represent the input vector. The equations of motion in (3.1) can be represented as:

$$\begin{aligned} \dot{x} &= Ax + \Gamma w, \\ y &= Cx + Dw + D_2\nu, \end{aligned} \tag{3.3}$$

where ν is measurement noise and A, Γ, C, D are constant matrices consistent with (3.1)-(3.2).

3.2.2 Input observer

To estimate the road input w in (3.3), we exploit a multi-input observer developed in [46]. For the half-car dynamics in (3.3), the model can be written as:

$$\dot{x}(t) = Ax(t) + \Gamma w(t). \tag{3.4}$$

The input observer has the following form,

$$\begin{aligned} \dot{\varepsilon}(t) &= -\gamma S\varepsilon(t) + \gamma SAx(t) + (\gamma S)^2 Kx(t), \\ \hat{w}(t) &= -\varepsilon(t) + \gamma SKx(t), \end{aligned} \tag{3.5}$$

where ε is the observer state, $\gamma > \frac{1}{2}$ is a scalar gain, $S = \frac{1}{2}\mathbf{I}_m + \frac{1}{2\gamma}P$ where P is a weighting matrix, m is the number of inputs, and $K = (\Gamma^{\mathrm{T}}\Gamma)^{-1}\Gamma^{\mathrm{T}}$ is the pseudo-inverse of Γ. Note that the column rank of Γ must be equal to m so that K is well-defined. Henceforth we assume that $P = \mathbf{I}_m$.

In [46], it is shown that, assuming $\|\dot{w}(t)\| \leq b_1$, the estimation error is bounded by

$$\|w(t) - \hat{w}(t)\|_P^2 \leq \|w(0) - \hat{w}(0)\|_P^2 e^{-\gamma t} + \frac{b_1^2}{2\gamma^2}. \tag{3.6}$$

Note that to implement the input observer in (3.5), we require a state estimate for x. To design a state estimator for the implementation of the input observer, we develop a JDP-based estimator below.

3.3 JDP-based estimation and comparison with the Kalman filter (KF)

3.3.1 JDP-based estimation

As discussed above, a state estimator is required to implement the input observer (3.3). Note that for the purpose of state estimation, road velocity inputs $w(t)$ are now treated as disturbances. The conventional treatment of the road velocity input as Gaussian noise is enhanced in this chapter with a JDP model that is able to account for rare but pronounced events such as potholes and bumps.

A JDP has the form $\eta + \sigma_\zeta \zeta$ where η is a vector Poisson (Jump) process, ζ is a standard vector Wiener (Diffusion) process and $\sigma_\zeta \sigma_\zeta^{\mathrm{T}} \geq 0$ is a covariance matrix. In a Kalman filter, it is assumed that $w(t) = \sigma_\zeta \dot{\zeta}$. The JDP-based estimator modifies this assumption so that

$$w(t) = \dot{\eta}(t) + \sigma_\zeta \dot{\zeta}(t), \tag{3.7}$$

where Poisson term η can model rare but pronounced events. In our problem, η represents road anomalies such as potholes or speed bumps.

For the half-car model (3.3) with the disturbance modeled in (3.7), the JDP-based estimator is assumed to have the following form,

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + F(\hat{y} - y) + (\Gamma + FD)\lambda\mu_\eta, \\ &= A\hat{x} + F(C\hat{x} - Cx - Dw - D_2\nu) + (\Gamma + FD)\lambda\mu_\eta, \end{aligned} \tag{3.8}$$

where F is the estimator gain to be determined, λ and μ_η are the rate parameter and mean jump size of η, respectively. Let $e = x - \hat{x}$ denote estimation error. Then,

$$\dot{e} = (A + FC)e + (\Gamma + FD)w + FD_2\nu - (\Gamma + FD)\lambda\mu_\eta. \tag{3.9}$$

Let $\nu = \sigma_\xi \dot{\xi}$ with ξ representing a standard vector Wiener process and $\sigma_\xi \sigma_\xi^{\mathrm{T}} \geq 0$ being the covariance matrix of ν. Defining $\bar{D} = [FD_2\sigma_\xi \quad (\Gamma + FD)\sigma_\zeta]$, we have,

$$de = \big[(A + FC)e - (\Gamma + FD)\lambda\mu_\eta\big]dt + \bar{D}\begin{bmatrix} d\xi \\ d\zeta \end{bmatrix} + (\Gamma + FD)d\eta. \tag{3.10}$$

Let $z = Se$ be a weighted estimation error to keep small. Select F to minimize

$$J = \lim_{t\to\infty} \frac{1}{t} E \int_0^t z^{\mathrm{T}}(\tau) z(\tau) d\tau. \tag{3.11}$$

Theorem 3.1. *Suppose the pair (C, A) is detectable, the pair (A, Γ) is stabilizable, $D_2 D_2^T > 0$ and $S^T S > 0$. Then the optimal gain F that minimizes (3.11) in the open set of all gains F for which $A + FC$ is asymptotically stable, is given by*

$$F = -\Gamma\bar{\Sigma} D^T V_2^{-1} - QC^T V_2^{-1}, \tag{3.12}$$

and Q is the unique positive semi-definite solution to

$$\begin{aligned}(A - \Gamma\bar{\Sigma} D^T V_2^{-1} C)Q + Q(A - \Gamma\bar{\Sigma} D^T V_2^{-1} C)^T \\ + V_1 - QC^T V_2^{-1} CQ = 0,\end{aligned} \tag{3.13}$$

where $V_1 = \Gamma\bar{\Sigma}\Gamma^T - \Gamma\bar{\Sigma} D^T V_2^{-1} D\bar{\Sigma}\Gamma^T$, $V_2 = D_2\sigma_\xi\sigma_\xi^T D_2^T + D\bar{\Sigma} D^T$ and $\bar{\Sigma} = \sigma_\zeta\sigma_\zeta^T + \lambda\mu_\eta\mu_\eta^T + \lambda\Sigma_\eta$ where Σ_η is the covariance of η.

Proof. The proof follows [48] and is represented for completeness since here a more general case is considered with vector input η and $D \neq 0$. Assuming that $\bar{A} = A + FC$ is asymptotically stable, and given that $S^{\mathrm{T}} S > 0$, we can find $M > 0$ such that,

$$M\bar{A} + \bar{A}^{\mathrm{T}} M = -S^{\mathrm{T}} S. \tag{3.14}$$

Let $V = \frac{1}{2} e^{\mathrm{T}} M e$. From (A.2), the infinitesimal of V, denoted by LV, is calculated as:

$$\begin{aligned} LV &= [\bar{A}e - (\Gamma + FD)\lambda\mu_\eta]^{\mathrm{T}} Me + \frac{1}{2} trace(\bar{D}^{\mathrm{T}} M\bar{D}) \\ &\quad + \int_{\mathbb{R}^m} \big[\frac{1}{2} s^{\mathrm{T}}(\Gamma + FD)^{\mathrm{T}} M(\Gamma + FD)s + e^{\mathrm{T}} M(\Gamma + FD)s\big]\lambda\phi(s)ds \\ &= \frac{1}{2} e^{\mathrm{T}}(\bar{A}^{\mathrm{T}} M + M\bar{A})e + \frac{1}{2} trace(\bar{D}^{\mathrm{T}} M\bar{D}) \\ &\quad + \frac{1}{2}\lambda \cdot trace[(\Gamma + FD)^{\mathrm{T}} M(\Gamma + FD)\Sigma_\eta] \\ &\quad + \frac{1}{2}\lambda \cdot \mu_\eta^{\mathrm{T}}(\Gamma + FD)^{\mathrm{T}} M(\Gamma + FD)\mu_\eta \\ &= -\frac{1}{2} e^{\mathrm{T}} S^{\mathrm{T}} Se + \frac{1}{2} trace(\bar{D}^{\mathrm{T}} M\bar{D}) + \frac{1}{2}\tilde{D}^{\mathrm{T}} M\tilde{D} + \frac{1}{2} trace[\hat{D}^{\mathrm{T}} M\hat{D}], \end{aligned} \tag{3.15}$$

where $\tilde{D} = \sqrt{\lambda}(\Gamma + FD)\mu_\eta$ and $\hat{D} = \sqrt{\lambda}(\Gamma + FD)N$ with $NN^{\mathrm{T}} = \Sigma_\eta$ being a lower triangular matrix from Cholesky decomposition.

This expression, along with $S^{\mathrm{T}}S > 0$, $M > 0$, implies that $E[V(t)]$ is bounded. Then, using Dynkin's formula, we have

$$\begin{aligned}\frac{1}{t}E[V(t)] - \frac{1}{t}E[V(0)] = -\frac{1}{2}\frac{1}{t}E\int_0^t z^{\mathrm{T}}(\tau)z(\tau)d\tau + \frac{1}{2}\tilde{D}^{\mathrm{T}}M\tilde{D} \\ + \frac{1}{2}trace(\bar{D}^{\mathrm{T}}M\bar{D}) + \frac{1}{2}trace(\hat{D}^{\mathrm{T}}M\hat{D}).\end{aligned} \tag{3.16}$$

Consequently,

$$\lim_{t\to\infty}\frac{1}{t}E\int_0^t z^{\mathrm{T}}(\tau)z(\tau)d\tau = \tilde{D}^{\mathrm{T}}M\tilde{D} + trace(\bar{D}^{\mathrm{T}}M\bar{D}) + trace(\hat{D}^{\mathrm{T}}M\hat{D}). \tag{3.17}$$

Next apply the method of indeterminate Lagrangian multipliers. There exist $\lambda_0 \in \{0, 1\}$ and a matrix Q, $(\lambda_0, Q) \neq 0$ which together with the optimal values of F and M, yield the first-order optimality conditions for function J from (3.11):

$$\begin{aligned}J &= \lambda_0\big[trace(\bar{D}\bar{D}^{\mathrm{T}}M) + trace(\tilde{D}\tilde{D}^{\mathrm{T}}M) \\ &\quad + trace(\hat{D}\hat{D}^{\mathrm{T}}M)\big] + trace\big[(M\bar{A} + \bar{A}^{\mathrm{T}}M + S^{\mathrm{T}}S)Q\big] \\ &= trace\big[(FD_2\sigma_\xi\sigma_\xi^{\mathrm{T}}D_2^{\mathrm{T}}F^{\mathrm{T}} + (\Gamma + FD)\sigma_\zeta\sigma_\zeta^{\mathrm{T}}(\Gamma + FD)^{\mathrm{T}})M\big] \\ &\quad + trace\big[\lambda\cdot(\Gamma + FD)\mu_\eta\mu_\eta^{\mathrm{T}}(\Gamma + FD)^{\mathrm{T}}M\big] \\ &\quad + trace\big[\lambda\cdot(\Gamma + FD)\Sigma_\eta^{\mathrm{T}}(\Gamma + FD)^{\mathrm{T}}M\big] \\ &\quad + trace\big[(M\bar{A} + \bar{A}^{\mathrm{T}}M + S^{\mathrm{T}}S)Q\big] \\ &= trace(FD_2\sigma_\xi\sigma_\xi^{\mathrm{T}}D_2^{\mathrm{T}}F^{\mathrm{T}}M) \\ &\quad + trace[(\Gamma + FD)\bar{\Sigma}(\Gamma + FD)^{\mathrm{T}}M] \\ &\quad + trace\big[(M\bar{A} + \bar{A}^{\mathrm{T}}M + S^{\mathrm{T}}S)Q\big],\end{aligned} \tag{3.18}$$

where $\bar{\Sigma} = \sigma_\zeta\sigma_\zeta^{\mathrm{T}} + \lambda\mu_\eta\mu_\eta^{\mathrm{T}} + \lambda\Sigma_\eta$. Therefore,

$$\frac{\partial J}{\partial F} = 2MFD_2\sigma_\xi\sigma_\xi^{\mathrm{T}}D_2^{\mathrm{T}} + 2M\Gamma\bar{\Sigma}D^{\mathrm{T}} + 2MFD\bar{\Sigma}D + 2MQC^{\mathrm{T}}. \tag{3.19}$$

Setting $\frac{\partial J}{\partial F} = 0$ implies that,

$$F = -QC^{\mathrm{T}}V_2^{-1} - \Gamma\bar{\Sigma}D^{\mathrm{T}}V_2^{-1}, \tag{3.20}$$

where $V_2 = D_2\sigma_\xi\sigma_\xi^{\mathrm{T}}D_2^{\mathrm{T}} + D\bar{\Sigma}D^{\mathrm{T}}$.

Also,

$$\begin{aligned}\frac{\partial J}{\partial M} &= FD_2\sigma_\xi\sigma_\xi^{\mathrm{T}}D_2^{\mathrm{T}}F^{\mathrm{T}} + \Gamma\bar{\Sigma}\Gamma^{\mathrm{T}} + 2FD\bar{\Sigma}\Gamma^{\mathrm{T}} \\ &\quad + FD\bar{\Sigma}D^{\mathrm{T}}F^{\mathrm{T}} + QA^{\mathrm{T}} + QC^{\mathrm{T}}F^{\mathrm{T}} + AQ + FCQ \\ &= FV_2F^{\mathrm{T}} + \Gamma\bar{\Sigma}\Gamma^{\mathrm{T}} + 2FD\bar{\Sigma}\Gamma^{\mathrm{T}} + 2QC^{\mathrm{T}}F^{\mathrm{T}} + QA^{\mathrm{T}} + AQ.\end{aligned} \tag{3.21}$$

Setting $\frac{\partial J}{\partial M} = 0$ and from (3.21), it follows that,

$$AQ + QA^{\mathrm{T}} + \Gamma\bar{\Sigma}\Gamma^{\mathrm{T}} - FV_2^{-1}F^{\mathrm{T}} = 0. \tag{3.22}$$

(3.22) along with (3.20) imply (3.13). □

Remark 3.2. The JDP-based estimator (3.8), (3.12), (3.13) is similar to the steady-state Kalman filter but an additional term $(\Gamma + FD)\lambda\mu_\eta$ is added. The additional term can potentially increase the dynamic response of the filter and thus is able to capture abrupt changes such as road anomalies. The algebraic Riccati equation in (3.13) has more terms that can encode additional disturbance information.

3.3.2 Simulation results

In this subsection, we simulate and compare state estimation and resulting input estimation performance of a Kalman filter and the JDP-based estimator. Both estimators are tuned manually to provide the best performance. Kalman filter parameters were set to $W = 10 \cdot \mathbf{I}_4$ and $V = 10^{-4} \cdot \mathbf{I}_3$, where W is the process noise covariance and V is the measurement noise covariance. The tuned parameters of the JDP estimator used in the simulation and experimental evaluations are listed in Table 3.1.

Table 3.1: JDP estimator parameters

μ_η	Σ_η	σ_ζ	σ_ξ	λ
$[1, 1]$	$5 \cdot \mathbf{I}_2$	$10 \cdot \mathbf{I}_2$	$0.01 \cdot \mathbf{I}_2$	0.01

We first present a performance comparison by simulating the response of hitting a pothole. A pothole can be modeled as an input for which a jump occurs at one wheel while the other one is flat. Figure 3.4 shows an example, where a left pothole at 2s and a right pothole at 4s are encountered. With the specified inputs in Figure 3.4,

the state estimation errors for x_2 and x_3 are illustrated in Figure 3.5. Note that x_1 and x_4 are directly measured.

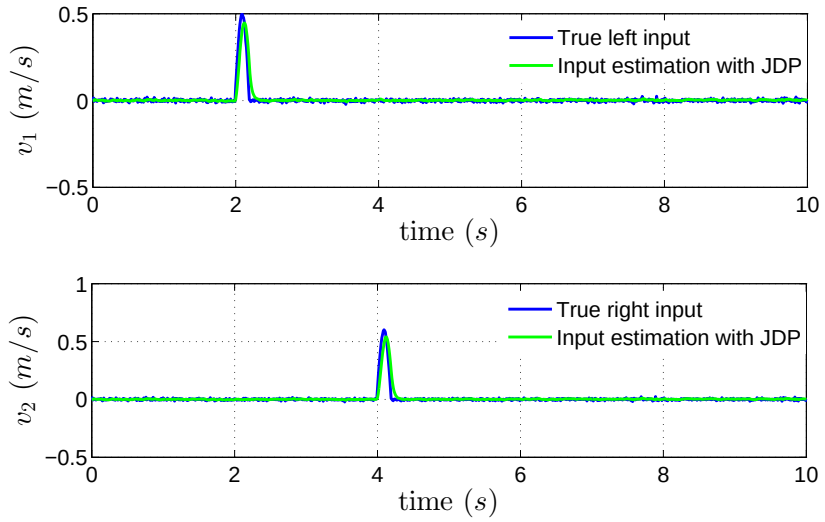


Figure 3.4: Input estimation with the JDP estimator

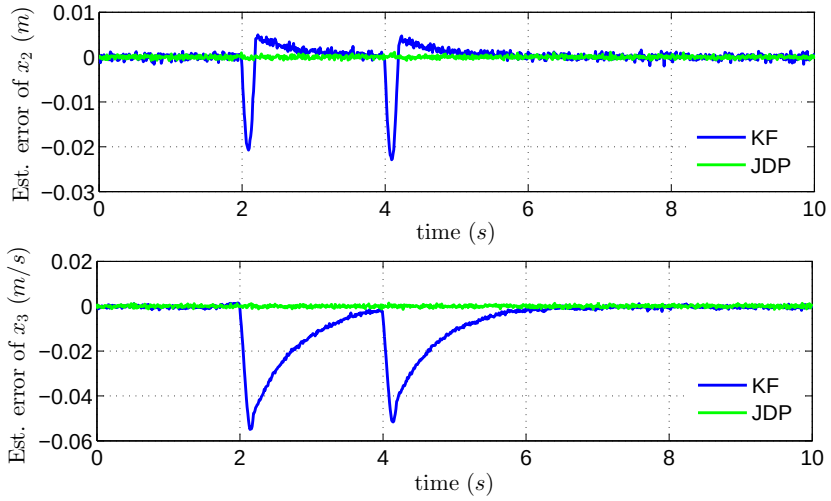


Figure 3.5: State estimation comparison with KF and JDP estimator

The Kalman filter works poorly when jumps occur, *i.e.*, around 2s and 4s. As a result, the input estimation for the jumps works poorly as can be seen in Figure 3.6. On the other hand, the JDP-based estimator works better than the Kalman filter as can be seen in Figure 3.5. The input estimation tracks the true inputs well with the JDP-based estimator, as shown in Figure 3.4.

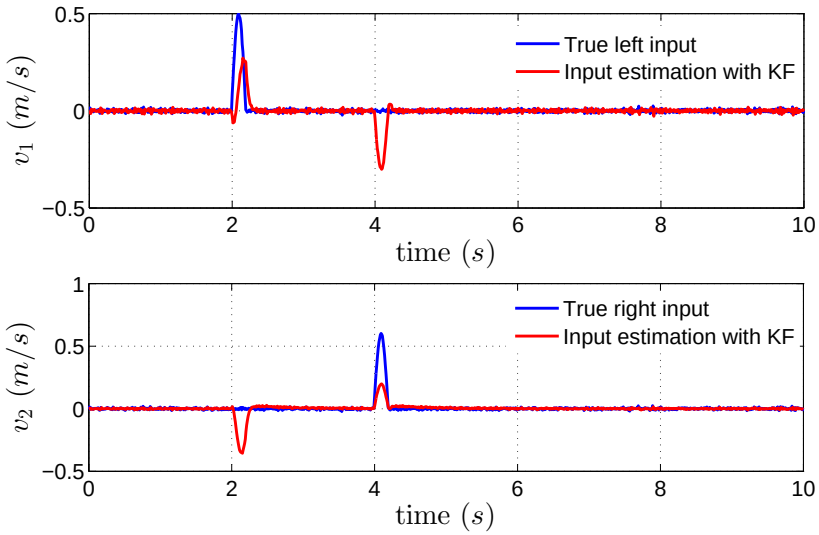


Figure 3.6: Input estimation with the KF

We next present a performance comparison in the case where the vehicle hits a speed bump. A speed bump can be modeled as an input where both wheels have jumps with similar magnitudes. See Figure 3.7 as an example where a speed bump is encountered at 2s. State estimation errors are illustrated in Figure 3.8. The JDP-based estimator tracks the true states well while the Kalman filter does not work well during and immediately after the jumps occur.

Consequently, as shown in Figure 3.7, the inputs with a JDP-based state estimator are tracked well, while the jumps are poorly tracked with the Kalman filter, as shown in Figure 3.9.

3.4 Anomaly detection algorithm and experimental results

3.4.1 Road anomaly detection algorithm

Section 3.3 showed that the JDP-based state estimator (3.8) coupled with the input observer (3.5) can be exploited to estimate road velocity inputs. With road velocity input estimation, Algorithm 3.1 is developed to detect and label road anomalies.

In the algorithm, $\hat{v}_1$ and $\hat{v}_2$ are, respectively, the maximum magnitudes of the estimated left and right road velocity inputs over the past time window. $\bar{v}$ is the average car speed over the time window; t_{p2p} denotes the peak-to-peak time duration from the last time window to the previous window; L is the car length; $\mathcal{L}^-$ is the label variable to indicate whether an anomaly was reported for the last time window.

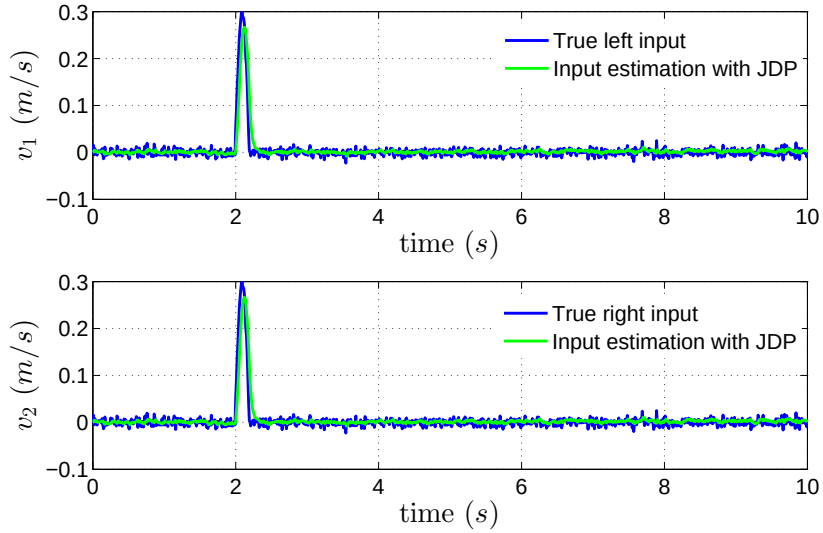


Figure 3.7: Input estimation with the JDP estimator

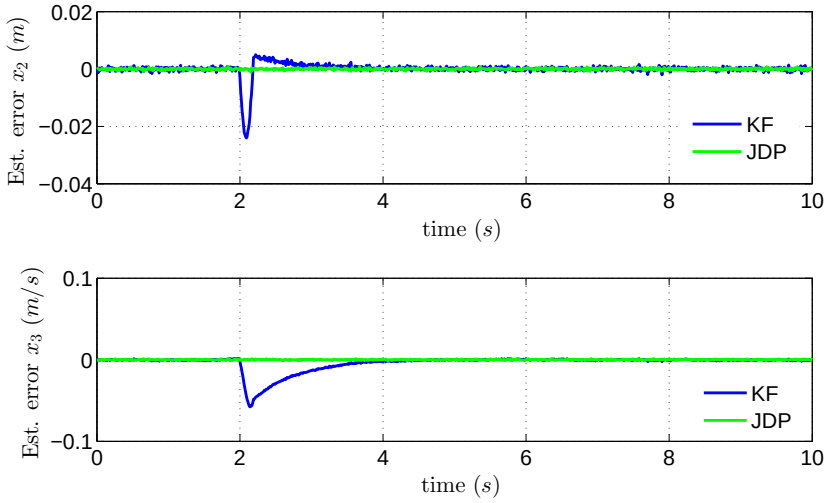


Figure 3.8: State estimation error comparison with KF and JDP estimator

Value $\mathcal{L}^- = 1$ indicates an anomaly was reported in that time window while $\mathcal{L}^- = 1$ indicates no anomaly. th_1 is a vehicle speed-dependent threshold for large anomaly response; $th_2 < th_1$ is a vehicle speed-dependent threshold for small anomalies and th_{perc} is a threshold for the left and right input difference. The reason that th_1 and th_2 are speed-dependent is higher vehicle speed, based on experimental data, leads to larger response for same road anomalies. In the real-time implementation, lookup tables are implemented for th_1 and th_2 as a function of average speed.

Since we use a front half-car model, repeated patterns can be found when the rear

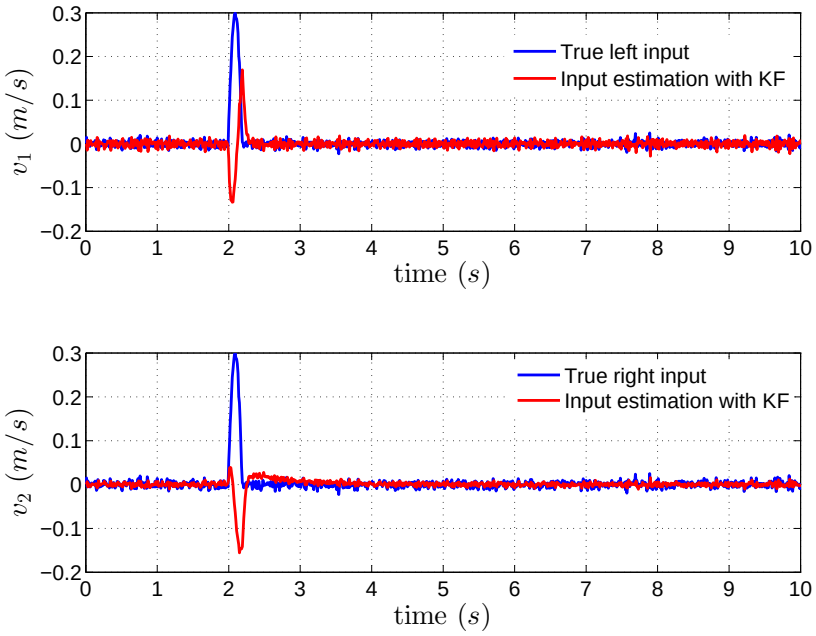


Figure 3.9: Input estimation with the KF

car hits the same anomaly. An experimental example is illustrated in Figure 3.13 where the repeated patterns can be found around 3s and 5.3s. The first *if* statement is used to differentiate the repeated pattern from a new anomaly. This is because the rear half of the car also hits the anomaly causing secondary jumps. We compare the peak-to-peak time duration t_{p2p} with car length L divided by the average speed during window $\bar{v}$. If $t_{p2p} < L/\bar{v}$, then it is not classified as a new anomaly. Otherwise, the subsequent steps are executed.

The anomaly detection algorithm is based on the fact that the pothole is an event where jump occurs on only one side while speed bumps or road joints are events where both sides have jumps. The two thresholds th_1 and th_2 are magnitude thresholds used to determine whether the response is large or small.

3.4.2 Experimental results

We have shown above that the road profile estimator works well in simulations. We also tested the profile estimator in a 2012 Lincoln MKS test vehicle, pictured in Figure 3.3. However, there is no "true" road velocity for comparison. As a result, we validate road input estimation performance by running our test vehicle on road segments with different road surface types to see whether the estimator can reflect

roughness levels. Three different surfaces are illustrated in Figure 3.10 where the roughness increases from left to right. The test vehicle was driven at approximately 30 km/h on the road segments and left input estimations $\hat{v}_1$ are illustrated in Figure 3.11.

(a) Surface 1 (b) Surface 2 (c) Surface 3

Figure 3.10: Three road surface types; roughness increases from left to right

Our anomaly detection algorithm validation procedure is straightforward. We ran our detection algorithm in real-time on dSPACE and an interface was used to indicate real-time detections as illustrated in Figure 3.12. The interface can indicate whether

Algorithm 3.1 Road Anomaly Detection Algorithm

Inputs: $\hat{v}_1$, $\hat{v}_2$, $\bar{v}$, t_{p2p}, $\mathcal{L}^-$

Outputs: Anomaly indication and anomaly label (left/right/large/small pothole, speed bump)

if $\mathcal{L}^- = 1$ and $t_{p2p} < L/\bar{v}$ **then**
    Report *no* anomaly.
else
    if $\hat{v}_1 > th_1(\bar{v})$ or $\hat{v}_2 > th_1(\bar{v})$ **then**
        if $\frac{|\hat{v}_1 - \hat{v}_2|}{\min(\hat{v}_1, \hat{v}_2)} < th_{perc}$ **then**
            speed bump detected.
        else if $\hat{v}_1 > \hat{v}_2$ **then**
            left large pothole detected.
        else
            right large pothole detected.
    else if $\hat{v}_1 > th_2(\bar{v})$ or $\hat{v}_2 > th_2(\bar{v})$ **then**
        if $\frac{|\hat{v}_1 - \hat{v}_2|}{\min(\hat{v}_1, \hat{v}_2)} < th_{perc}$ **then**
            road joints detected.
        else if $\hat{v}_1 > \hat{v}_2$ **then**
            left small pothole detected.
        else
            right small pothole detected.
    else
        report *no* anomaly.

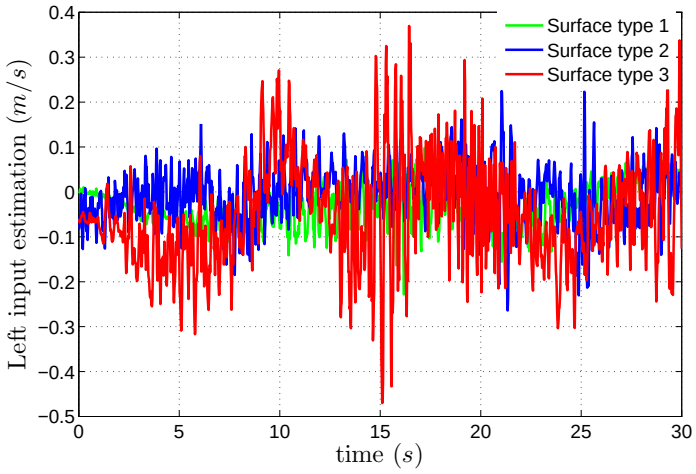


Figure 3.11: Input estimation on the three road surfaces

an anomaly has been detected (state light), the anomaly type, GPS coordinates, left and right inputs, and time. Figure 3.13 shows estimated road inputs in a vehicle test

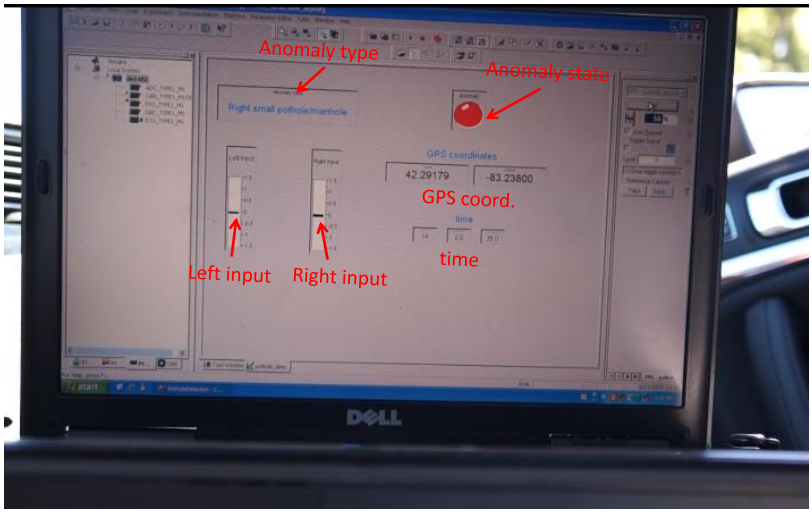


Figure 3.12: Real-time dSPACE ControlDesk interface

when hitting a small pothole followed by a large pothole at the right side. Jumps are detected approximately 2s and 4.2s. Using anomaly detection Algorithm 3.1, a right small pothole and a right large pothole were successfully detected. Note that repeated patterns can be found approximately 3s and 5.3s. This is because the rear half of the car also hits the pothole causing secondary jumps. To differentiate the repeated pattern from a new pothole, we compare the peak-to-peak time duration t_{p2p} with the car length L divided by the minimum speed during window v_{min}. If $t_{p2p} < L/v_{min}$, then it is not classified as a new anomaly. Otherwise, a new anomaly

is detected.

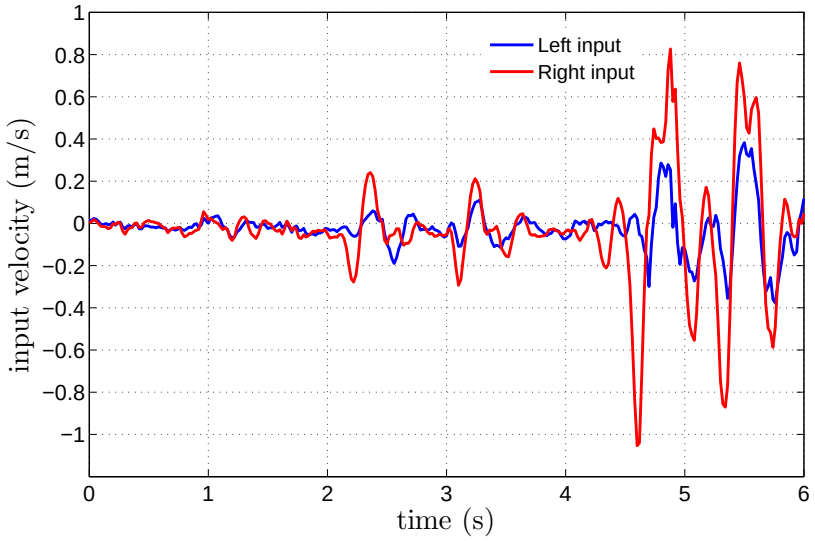


Figure 3.13: Input estimation when driving over a small pothole followed by a large pothole

Note that we compared the right and right side inputs to determine the anomaly types. If the vehicle happens to hit a right pothole and left pothole simultaneously with similar magnitudes, current algorithm may treat it as a speed bump.

As we drove our vehicle around Dearborn, Michigan, anomalies were detected and recorded. One of the trips is shown in Figure 3.14. During our test drives, the algorithm did not reveal any false positives and missed detections were rare. The missed detections were mainly due to smooth manholes and road joints, which are not true anomalies given our classifications. Anomaly labeling accuracy was 145 out of 168 correctly labeled reports. Note that some of the labeling errors were due to events such as uneven speed bumps.

We note that our test vehicle can be used as a probe vehicle for creating a detailed anomaly map. This map can be stored on a cloud and shared with drivers to better plan their routes. Anomaly map data is also useful for road agencies to efficiently maintain the roads.

3.5 Summary and discussion

This chapter proposed and validated a method of simultaneous road profile estimation and anomaly detection. A multi-input observer was used to estimate road velocity inputs. A jump diffusion process (JDP)-based estimator was developed and

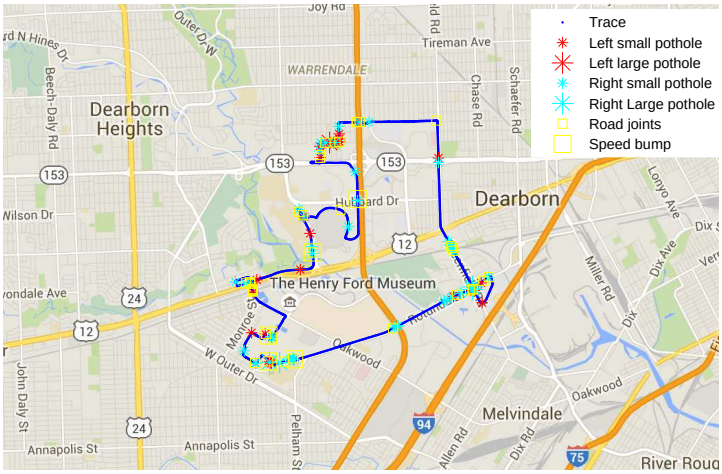


Figure 3.14: Detected anomalies in Dearborn with our test vehicle

shown to have better estimation performance than a Kalman filter when jumps are present. The JDP-based estimator forms the central element of an anomaly detection algorithm. Anomaly detection was implemented in real-time on a test vehicle and promising experimental results were obtained. The anomaly detection procedure can be applied to create a crowd-sourced anomaly map which can assist drivers in route planning and help road agencies maintain the roads.

CHAPTER 4

Cloud-enabled anomaly reports clustering

4.1 Introduction

Mobile sensing and data sharing offer new opportunities to advance intelligent transportation systems. Modern vehicles are embedded with sophisticated sensors and control units that can be exploited to characterize road and environmental information in real time. References [10, 11, 12] provide examples of traffic density estimation, road friction coefficient estimation and pothole detection, respectively. Sensed information can be sent to a server, *e.g.,* the cloud, to be further processed, crowd-sourced, then shared with other vehicles and road agencies.

Road anomalies such as potholes and bumps are annoying events that can cause ride discomfort and vehicle damage. If available, anomaly maps can be used to enhance route planning [31], improve suspension control [28] and notify road maintenance. Anomaly detection algorithms have been developed in previous work. For example, a pothole detector with three external accelerometers was developed using machine learning techniques [12]. In [26] and Chapter 2 of this thesis, we developed a road anomaly detection algorithm based on a half-car model by exploiting a multi-input observer. Promising detection performance was demonstrated in a test vehicle using standard sensors.

Anomaly detectors can be integrated in a Vehicle-to-Cloud-to-Vehicle (V2C2V) framework as illustrated in Figure 4.1. A fleet of vehicles equipped with anomaly detectors are deployed as mobile sensors. Anomaly locations, *e.g.*, from Global Positioning System (GPS) coordinates, are sent to the cloud, where a clustering module is implemented to process raw anomaly reports. Clusters with high credibility score are stored in a cloud database where their locations can later be broadcast to other vehicles and road agencies. Clusters with low credibility score are stored in a buffer and not shared. In this chapter, we develop a clustering algorithm that can process raw

reports and retrieve useful anomaly information. The desired clustering algorithm has the following properties:

- *No assumptions on number of clusters.* The number of road anomalies can not be known in advance and is continuously evolving. New anomalies can develop and old anomalies can disappear once repaired. The algorithm hence should not assume a constant number of clusters [49, 50].

- *Ability to handle outliers.* False alarms can sometimes occur. The clustering algorithm should be able to discriminate outliers and not broadcast outlier information to vehicles and road agencies.

- *Consideration of anomaly evolution.* Road anomalies are evolving, that is, new potholes may occur and old potholes may be fixed. The clustering algorithm must be able to deal with change in aggregated reports.

- *Localization of isolated anomalies and information compression for stretched anomaly segments.* The algorithm should also be able to accurately localize isolated anomalies and compress summary information from a segment with intensively distributed anomalies.

- *Memory and computation efficiency.* The clustering algorithm needs to process large-scale data streams as efficiently as possible. Cluster information should be stored in a compact data structure and updated with minimal computational overhead.

In general, clustering algorithms partition data into groups with underlying patterns. They are widely applied in the fields of image processing [51], data mining [52], and diagnostics and prognostics [53]. Many clustering algorithms are designed to deal with static data [49, 54, 55], that is, cases where all data are available in advance. These algorithms are not applicable to processing anomaly reports since the reports are dynamic and time-evolving. Recently, clustering algorithms have been developed to deal with evolving data streams. The CluStream algorithm [50] exploits micro-clusters to summarize information for a set of data points. The micro-clusters are updated online with new stream inputs and a weighted k-means algorithm is applied offline on the micro-clusters to obtain the final clusters. While good accuracy can be obtained, the algorithm assumes a constant number of clusters so it cannot be used in our problem. In [56], a streaming k-means clustering algorithm is developed with a divide-and-conquer strategy. It optimizes a k-means objective function and

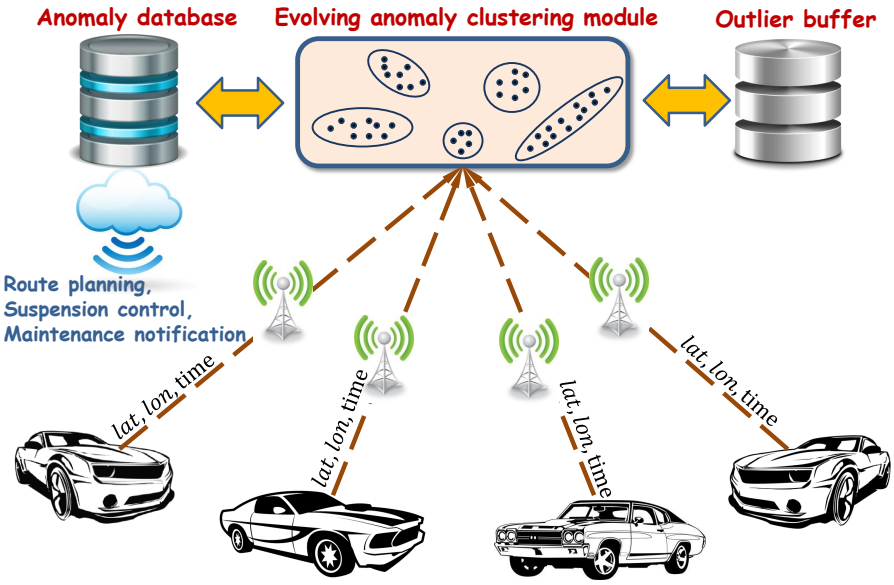


Figure 4.1: Vehicle-to-Cloud-to-Vehicle anomaly detection and information sharing

can generate more than k clusters. However, the obtained clusters are hypercircles which cannot be used to compress information for stretched anomaly segments. Also, outliers and the evolving nature of anomalies cannot be well handled.

An extended Gustafson-Kessel algorithm is developed in [57], where Mahalanobis distance is exploited to measure the similarity between clusters and new data points. The cluster center and covariance matrix are updated recursively with new data inputs. Updated clusters are hyperellipsoids with arbitrary orientation. The algorithm is applicable to real-time pattern recognition and information compression. However, it is not able to deal with outliers and cannot capture the road anomalies that change over time.

In this chapter, we develop a novel clustering framework that satisfies all specified requirements. Similar to [57], we exploit Mahalanobis distance as a similarity metric. Two cluster types, the outlier cluster and the main cluster, are defined based on their computed credibility values. A cluster feature vector is defined by a weight, center, covariance matrix inverse, creation time and a label. Clusters are updated in a single-pass setting; A Woodbury matrix inverse lemma [58] is exploited to simplify the covariance matrix update and avoid possible singularity issues in numerical computations. Clusters are pruned based on their weights and creation time to deal with outliers as well as anomaly changes over time. Memory and computations is

are light and simulation results demonstrate the efficiency of the proposed clustering algorithm.

The chapter is organized as follows. Section 4.2 presents background on Mahalanobis distance and its relation to the χ^2 distribution. Section 4.3 is devoted to the discussion of cluster feature definition and the clustering algorithm. Simulation results are described in Section 4.4, followed by a summary and discussion in Section 4.5.

4.2 Mahalanobis distance and χ^2 distribution

The Mahalanobis distance measures the similarity between a point and a cluster of points [59]. It generalizes a notion of number of standard deviations between a point and the mean of the cluster for multi-dimensional data. The distance grows as the point moves away from the mean along each principal component axis. As a result, the distance is unitless and scale-invariant, and accounts for the distribution and correlations of the cluster data. Let $x \in \mathbb{R}^n$ be a data point. Let μ and Σ be the mean and covariance matrix of a cluster of points denoted by $\mathcal{C}$, respectively. The Mahalanobis distance between x and $\mathcal{C}$, $\mathcal{D}(x, \mathcal{C})$, is defined as:

$$\mathcal{D}(x, \mathcal{C}) = \sqrt{(x-\mu)^{\mathrm{T}} \Sigma^{-1} (x-\mu)}. \tag{4.1}$$

Note that Mahalanobis distance in (4.1) includes the Euclidean distance between x and μ as a special case with Σ being the identity matrix.

Suppose the data points are normally distributed around the cluster center μ with covariance Σ, *i.e.*, $X \sim \mathcal{N}_n(\mu, \Sigma)$, where $\mathcal{N}_n$ represents the multivariate normal distribution of dimension n. Define

$$Z = \Sigma^{-\frac{1}{2}}(X - \mu) = [Z_1,\, Z_2, \cdots,\, Z_n]^{\mathrm{T}}.$$

It is straightforward to show that $Z \sim \mathcal{N}_n(\mathbf{0}_n, \mathbf{I}_n)$, where $\mathbf{0}_n$ and $\mathbf{I}_n$ represent the zero vector of dimension n and identity matrix of dimension n, respectively. As a result, the Mahalanobis distance in (4.1) between X and $\mathcal{C}$ is:

$$\mathcal{D}^2(X, \mathcal{C}) = Z^{\mathrm{T}} Z = Z_1^2 + Z_2^2 + \cdots + Z_n^2, \tag{4.2}$$

which means that $\mathcal{D}^2(X, \mathcal{C})$ is chi-square distributed with degrees of freedom n, *i.e.*, $\mathcal{D}^2(X, \mathcal{C}) \sim \chi_n^2$. The chi-square value is often associated with a p-value, which is

defined as the probability of obtaining a result equal to or "more extreme" than what is observed. The chi-square distribution is frequently used in statistical hypothesis testing. The value $(1-p)$ is known as the confidence interval (CI) that represents the probability of $\mathcal{D}^2(X,\mathcal{C}) < \chi^2_n(p)$. The cross references of the p-value, CI and the sigma values (σ, standard deviation) for $n=1$ and χ^2_n values for some low dimensions are given in Table 4.1.

Table 4.1: Cross-reference table of p-value, CI and sigma band for $n=1$, and χ^2 values for $n=1,2,\cdot,6$

σ band	1σ	2σ	3σ	4σ
CI (%)	68.3%	95.45%	99.73%	99.99%
p-value	0.317	0.0455	0.0027	0.000006
$\chi^2_1(p)$	1	4	9	16
$\chi^2_2(p)$	2.3	6.18	11.83	19.33
$\chi^2_3(p)$	3.53	8.02	14.16	22.06
$\chi^2_4(p)$	4.72	9.72	16.25	24.50
$\chi^2_5(p)$	5.89	11.31	18.21	26.77
$\chi^2_6(p)$	7.04	12.85	20.06	28.91

4.2.1 Math preliminaries

In this subsection, we introduce the following lemma that will be used in subsequent developments.

Lemma 4.1. *([58]): Let A, B, C, D be matrices of appropriate dimensions. Suppose matrices A, C, and $C^{-1}+DA^{-1}B$ are nonsingular, then*

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1}. \tag{4.3}$$

Lemma 4.1 is often referred to as the *Woodbury matrix inversion lemma.*

4.3 Anomaly Report Stream Clustering Algorithm (ARSCA)

In this section, we develop a new clustering algorithm ARSCA to process road anomaly report stream that satisfies all the desired properties specified in Section I. We first introduce the cluster features, followed by the developed clustering algorithm ARSCA.

4.3.1 Cluster features

The main goal of our clustering algorithm is to obtain anomaly information by processing aggregated anomaly reports from vehicles. To achieve this goal, we represent each cluster $\mathcal{C}_i$, $i = 1, 2, \cdots, c$ with a tuple,

$$\mathcal{C}_i = (w_i, v_i, \Sigma_i^{-1}, t_i^0, \mathcal{L}_i), \tag{4.4}$$

where $w_i = \sum_{k=1}^{M_i} f(t - t_{ik})$ is the weight of cluster $\mathcal{C}_i$ with M_i being the number of anomaly reports in the cluster. t and t_{ik} denote the current time instant and the time instant that report k, x_{ik}, was added to cluster i, respectively. Time stamps t and t_{ik} are represented in a common time unit, for example, in days. $f(\tau) = \alpha^{-\lambda\tau}$ is a decaying function with α and λ being two positive parameters. The weight w_i reflects the credibility score of the cluster where high w_i represents high credibility. Note that the weight of a newly received report is one and the weight decays as a function of the past time period.

The cluster center v_i is defined as a weighted mean:

$$v_i = \frac{\sum_{k=1}^{M_i} f(t - t_{ik}) x_{ik}}{\sum_{k=1}^{M_i} f(t - t_{ik})}, \tag{4.5}$$

and Σ_i is the weighted covariance matrix defined as

$$\Sigma_i = \frac{\sum_{k=1}^{M_i} f(t - t_{ik})(x_{ik} - v_i)(x_{ik} - v_i)^{\mathrm{T}}}{\sum_{k=1}^{M_i} f(t - t_{ik})}. \tag{4.6}$$

We track the inverse of Σ_i, instead of Σ_i, for the convenience of recursive computation as detailed in Section 4.3.3. The time stamp $t_i^0 > 0$ represents the time when cluster $\mathcal{C}_i$ was first created.

The variable $\mathcal{L}_i \in \{m, o\}$ in (4.4) serves as a label indicating the cluster type. We specify two cluster types, main clusters (m-clusters, $\mathcal{L}_i = m$) and outlier clusters (o-clusters, $\mathcal{L}_i = o$). M-clusters are the clusters with high credibility that are believed to represent true anomalies. Credibility of clusters are reflected in the cluster weight w_i, where high w_i implies high credibility. On the other hand, o-clusters represent outliers due to false alarms or those representing true anomalies that do not yet have high aggregated weights.

We note that the m-clusters and o-clusters can be interchanged as weights change. This allows new anomalies to become m-clusters and removed (repaired) anomalies to

become outliers. Two thresholds h_m and h_o, $h_o > h_m > 0$, are introduced to capture the interchange capability. For an m-cluster, if few reports are gained and the cluster weight decays such that $w_i < h_m$, then cluster i will be relabeled an o-cluster. For an o-cluster, if the cluster weight, with aggregated reports, increases such that $w_i > h_o$ then the cluster is relabeled an m-cluster. The constraint $h_m < h_o \cdot \alpha^{-\lambda}$ avoids clusters repeatedly switching labels. Thresholds h_m and h_o can be set as a function of annual average daily traffic (AADT) for each road segment.

On the other hand, if an o-cluster fails to become an m-cluster after a certain period of time T_o, it means that the o-cluster corresponds to false alarms and should be deleted from the outlier buffer. We check all the o-clusters periodically, *e.g.,* at the end of each day. If $t - t_i^0 > T_o$, then delete $\mathcal{C}_i$ from the outlier buffer, where t is current time and t_i^0 is cluster creation time.

Obtained clusters are hyperellipsoids with orientation determined by the principal axes of their covariance matrices Σ_i. The cluster representation can localize true location for isolated events and can compress information for a stretch of anomaly events with arbitrary orientation as illustrated in Figure 4.2. A stretch of three anomalies is included in Cluster 1 with an orientation aligned with the road. Moreover, it can accurately indicate the anomaly location for an isolated anomaly as in Cluster 2.

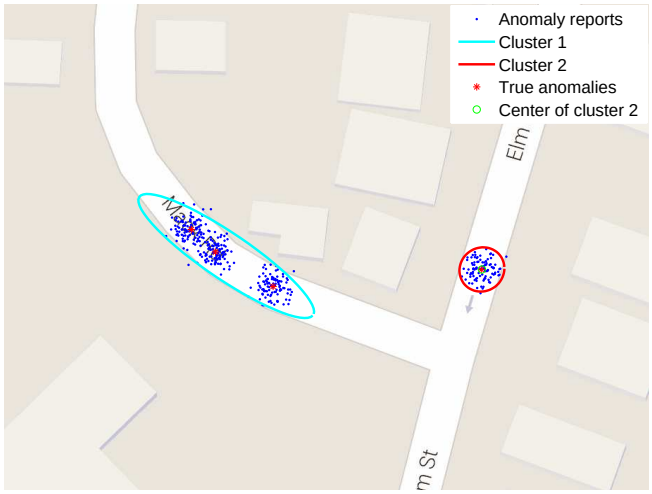


Figure 4.2: Clusters in forms of ellipsoids

M-cluster information is stored in a cloud database and can be shared between vehicles for route planning [31], suspension control [28], or other purposes. The information can also inform road agencies for maintenance. O-clusters are stored in a buffer that is not shared with other users.

4.3.2 Cluster maintenance algorithm

In this chapter, we develop Algorithm 4.1 to process road anomaly report streams. When a new anomaly report arrives, there are two possible scenarios. First, if the newly arrived report is "close" to some existing clusters, then the data should be merged into the "closest" one. On the other hand, if there is no existing cluster or the data is not close to any of the existing clusters, a new cluster should be created and centered at the new reported location. The "closeness" or similarity of the newly reported location and existing clusters is measured by the Mahalonobis distance discussed in Section 4.2. These scenarios (or conditions) are characterized by the *if* statements at Steps 4 and 9 in Algorithm 4.1.

Note that since the Mahalanobis distance is unitless and scale-invariant, we are able to directly process GPS coordinates without transforming it to state plane coordinates in Euclidean space.

Let $\mathcal{C}$ and $\mathcal{C}^+$ define the set of old cluster features and updated cluster features, respectively. Let X define the sequence of anomaly reports. Suppose there are c existing clusters when a new report $x_{\text{new}} = (lon,\, lat)$ arrives. Then based on (4.1), the squared Mahalanobis distance between data x and cluster $\mathcal{C}_i = (w_i,\, v_i,\, \Sigma_i^{-1},\, t_i^0,\, \mathcal{L}_i)$, $i = 1, 2, \cdots, c$ is calculated as

$$\mathcal{D}^2(x, \mathcal{C}_i) = (x - v_i)^{\mathrm{T}} \Sigma_i^{-1} (x - v_i). \tag{4.7}$$

The cluster, i^*, with the minimum distance can be obtained as

$$i^* = \arg \min_{i=1,\cdots,c} \mathcal{D}^2(x, \mathcal{C}_i). \tag{4.8}$$

Finding the "closest" cluster i^* using (4.7) and (4.8) is illustrated by Steps 7 and 8 in Algorithm 4.1.

As discussed in Section 4.2, suppose x is normally distributed around v_{i^*} with covariance matrix Σ_{i^*}. Then $\mathcal{D}^2(x, \mathcal{C}_{i^*}) \sim \chi_n^2$, where $n = 2$ is the dimension of x. We thus define a boundary parameter $\chi_2^2(p)$ to determine whether the data is close enough to the cluster and can be included in the cluster. The bound can be different between urban and rural areas due to GPS characteristics.

Consequently, if the squared Mahalanobis distance to the closest cluster is within the bound $\mathcal{D}^2(x, \mathcal{C}_{i^*}) \leq \chi_2^2(p)$ then we merge x into cluster i^*, which is captured by Step 9 in Algorithm 4.1. The weighted mean v_{i^*}, weighted covariance matrix Σ_{i^*}, and

Algorithm 4.1 ARSCA

Constant Parameters: p, α, λ, h_o, h_m, T_o, γ

Inputs: $\mathcal{C}$, X

Outputs: $\mathcal{C}^+$

top:

do

  Read next report $x_k \in X$

  if no cluster exists, **then**

    Initialize the first cluster $\mathcal{C}_1$:

$$w_1 = 1,\ v_1 = x_1,\ \Sigma_1^{-1} = \gamma \mathbf{I}_2,\ \mathcal{L}_1 = o,\ t_1^0 = t.$$

  else

    Calculate the Mahalanobis distances to all existing clusters:

$$\mathcal{D}^2(x_k, \mathcal{C}_i) = (x_k - v_i)\Sigma_i^{-1}(x_k - v_i)^{\mathrm{T}}, \quad i = 1, \cdots, c.$$

    Find the closest cluster as:

$$i^* = \arg \min_{i=1,\cdots,c} \mathcal{D}^2(x_k, \mathcal{C}_i).$$

    if $\mathcal{D}^2(x_k, \mathcal{C}_{i^*}) \leq \chi_2^2(p)$, **then**

      Update the covariance matrix inverse of $\mathcal{C}_{i^*}$ using (4.13) and (4.14).

      Update the center of $\mathcal{C}_{i^*}$ using (4.10).

      Update the weight of $\mathcal{C}_{i^*}$ using (4.9).

      if $\mathcal{L}_{i^*} = o$ and $w_{i^*} > h_o$, **then**

        Set $\mathcal{L}_{i^*} = m$ and update the cloud database with C_{i^*}.

    else

      Create a new cluster and increment $c = c + 1$.

      Initialize the new cluster $\mathcal{C}_c$ as:

$$w_c = 1,\ v_c = x_k,\ \Sigma_c^{-1} = \gamma \mathbf{I}_2,\ t_c^0 = t,\ \mathcal{L}_c = o.$$

while More than m minutes before the end of day t

Within m minutes to the end of day t:

Update the cluster weights:

$$w_i^+ = w_i \cdot \alpha^{-\lambda}, \quad i = 1, 2, \cdots, c.$$

Check the m-clusters:

if $\mathcal{C}_i$ is an m-cluster and $w_i < h_m$ after the update, **then**

  Set $\mathcal{L}_i = o$; delete it from the cloud database; send it to the outlier buffer.

Check the o-clusters:

if $\mathcal{C}_i$ is an o-cluster and $t - t_i^0 = T_o$, **then**

  delete $\mathcal{C}_i$ from the outlier buffer.

increment the day count: $t = t + 1$.

go to *top*.

cluster weight w_{i^*} are, respectively, updated as

$$w_{i^*}^{+} = w_{i^*} + 1, \tag{4.9}$$

$$v_{i^*}^{+} = \frac{w_{i^*} v_{i^*} + x}{w_{i^*} + 1}, \tag{4.10}$$

$$\Sigma_{i^*}^{+} = \frac{w_{i^*} \Sigma_{i^*} + (x - v_{i^*})(x - v_{i^*})^{\mathrm{T}}}{w_{i^*} + 1}, \tag{4.11}$$

where the superscript "$+$" denotes "update". The weight and mean updates are given by Steps 11 and 12 in Algorithm 4.1, respectively. Note that the inverse covariance update at Step 10 in Algorithm 4.1 does not use (4.11). Instead, we exploit the Woodbury Inverse Lemma to avoid potential numerical computation issues as will be discussed in the next subsection.

Suppose cluster i^* is an o-cluster and with increased weight, the updated weight w_{i^*} becomes greater than threshold h_o. In this case we relabel cluster i^* as an m-cluster and its information is stored on a cloud database and shared with vehicles and agencies. This relabeling procedure is presented by Steps 13 and 14 in Algorithm 4.1.

On the other hand, if $\mathcal{D}^2(x, \mathcal{C}_{i^*}) > \chi_2^2(p)$, that is, the data is outside all cluster boundaries, we increment $c = c + 1$ and assign a new cluster $\mathcal{C}_c$ to x,

$$\mathcal{C}_c = (1, x, \gamma \mathbf{I}_n, o, t), \tag{4.12}$$

where $\gamma > 0$ is a scalar that initializes the covariance matrix inverse and t is the current time (in days). This initialization is illustrated by Steps 15-17 in Algorithm 4.1.

At the end of each time period, the weight of each cluster is decayed by multiplying by $\alpha^{-\lambda}$. We check the weights of all m-clusters: if any m-cluster has a weight less than h_m, then it is removed from the database and sent to o-clusters in the outlier buffer. O-cluster weights are also updated. If the cluster creation day of an o-cluster is less than the current day minus T_o, a time duration threshold to delete o-clusters if they fail to become m-clusters, then the cluster is removed from storage. These relabeling and pruning procedures are characterized by Steps 20-26.

Note that anomaly reports are processed in a single pass; that is, they are all processed exactly once. Anomaly information is summarized in cluster features without storing separate reports. This reduces memory and computation resources requirements in comparison to algorithms that store individual reports.

4.3.3 Recursive computation of matrix inverse

Expression (4.11) provides a simple way to update the covariance matrix for clusters. However, after each update, the inverse of covariance matrix must be computed to estimate Mahalanobis distance from next arrived point as in (4.7). This recursive computation of matrix inversion may cause singularity issues due to numerical ill-conditioning. As an alternative, we exploit Lemma 4.1, the *Woodbury matrix inversion lemma*, to resolve this issue.

Let $A = \frac{w_i}{w_i+1}\Sigma$, $B = x - v_i$, $C = \frac{1}{w_i+1}$, and $D = (x - v_i)^{\mathrm{T}}$. From (4.3) and (4.11), it follows that

$$\begin{aligned}(\Sigma_i^+)^{-1} &= \big(\frac{w_i}{w_i+1}\Sigma + (x-v_i)\frac{1}{w_i+1}(x-v_i)^{\mathrm{T}}\big)^{-1} \\ &= \frac{w_i+1}{w_i}\Sigma_i^{-1} - \frac{w_i+1}{w_i}\Sigma_i^{-1}(x-v_i)\Big[(w_i+1) \\ &\quad + (x-v_i)^{\mathrm{T}}\frac{w_i}{w_i+1}\Sigma_i^{-1}(x-v_i)\Big]^{-1}\cdot \\ &\quad (x-v_i)^{\mathrm{T}}\frac{w_i}{w_i+1}\Sigma_i^{-1} \\ &= (1+\frac{1}{w_i})\Sigma_i^{-1}\big(\mathbf{I}_n - K_i\Sigma_i^{-1}\big),\end{aligned} \tag{4.13}$$

where

$$K_i = (x-v_i)\Big[w_i + (x-v_i)^{\mathrm{T}}\Sigma_i^{-1}(x-v_i)\Big]^{-1}(x-v_i)^{\mathrm{T}}. \tag{4.14}$$

Note that calculation of K_i in (4.14) requires only a scalar inversion. Recursive matrix inversion computations with (4.13) replace numerical matrix inversion with a simple algebraic calculation. This technique greatly simplifies the calculation and resolves possible singularity issues. The update of the covariance matrix inverse using (4.13) and (4.14) is illustrated by Step 10 in Algorithm 4.1.

4.3.4 Parameter selection discussion

The following parameters must be specified to implement the proposed clustering algorithm.

- p-value. p-value is required to obtain a chi-square value bound $\chi_2^2(p)$ based on Table 4.1. This bound controls the distance over which reports can be merged to an existing cluster. This parameter depends on GPS accuracy that may vary from urban to rural areas.

- α and λ in the decay function. This pair of positive parameters is used in the

decay function to control dependence on old data. Large α and λ represent less dependence on old data. Note that maintenance information from road agencies can be incorporated to change decay rate if we know anomalies in certain areas have been repaired.

- Thresholds $h_o > h_m > 0$. These two thresholds define the criteria for switching between m-cluster and o-cluster. Thresholds vary with the AADT of the road segments. The higher the AADT, the higher h_o and h_m should be.

- Time unit and pruning period T_o. Since the life cycle of road anomalies is typically at least a few days, it is reasonable to use "day" as the time unit. Time duration parameter T_o controls how long we keep an o-cluster. If an o-cluster exists more than T_o days and does not change to an m-cluster, we remove it from memory.

- Inverse covariance matrix initialization parameter γ. The parameter $\gamma > 0$ is used to initialize the inverse covariance matrix in a new cluster. Simulation results show that $\gamma = 10^8$ works well.

- Time parameter m. The time parameter m depends on how long it needs to update the cluster weights. For example, $m = 3$ means that the last 3 minutes of each day are used to update the cluster weights. New reports are not processed during that time.

4.4 Simulation demonstration

In this section, we present a simulation to demonstrate the efficacy of the proposed clustering algorithm. We simulate the algorithm over a period of 15 days on the roads around North Campus of the University of Michigan as illustrated in Figure 4.3. There are 10 true anomalies and 4 false alarms in the simulation.

From day 1 to day 5, anomalies 1-9 are present. On day 6, anomalies 6-8 are fixed and anomaly 10 is developed. The total number of reports is uniformly distributed between 100 and 150 each day. The reports are randomly generated around the true anomalies with a covariance matrix corresponding to a $5 \cdot \mathbf{I}_2\,(m^2)$ covariance in the state plane coordinates. Two of the outliers are generated on day 3 and two are generated on day 5.

Parameters in Table 4.2 are used in the Matlab simulation. The chosen p-value corresponds to 3σ bands. Decay parameters α and λ, thresholds h_o and h_m, and

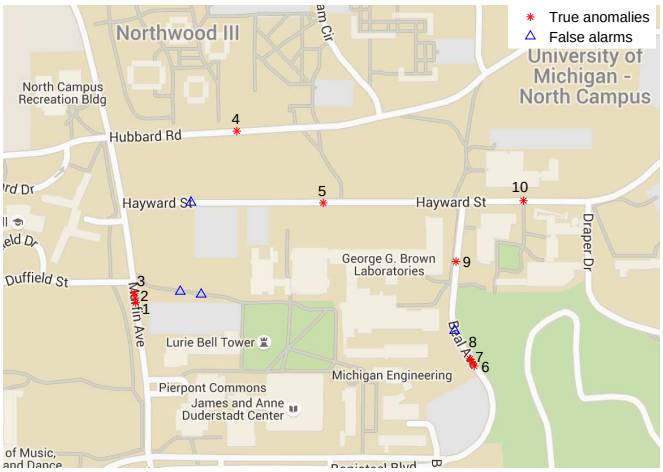


Figure 4.3: Road anomalies and false alarms

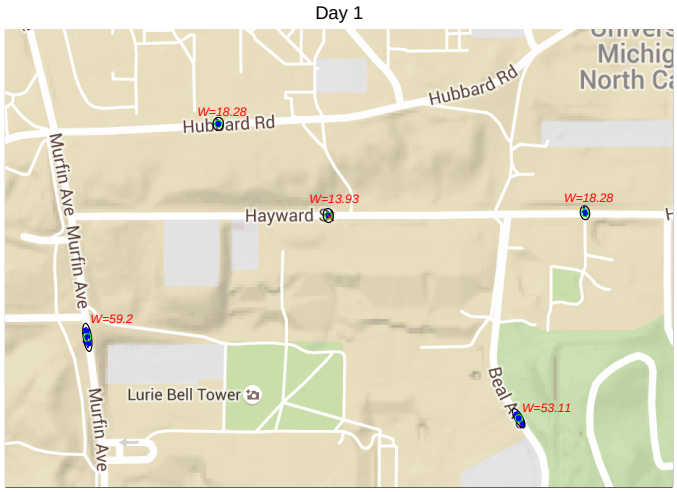


Figure 4.4: Snapshot at the end of day 1

pruning period T are chosen based on statistics specified in the simulation setup.

Table 4.2: Parameters for simulation

p	α	λ	h_o	h_m	T_o	γ
0.0027	2	0.2	80	50	5	10^8

Note that anomalies 1, 2, 3 and anomalies 6, 7, 8 are groups that can be included in one cluster each for the benefits of memory efficiency. A snapshot at the end of day 1 is illustrated in Figure 4.4. With aggregated reports, the weight of each cluster increases. However, since none of the clusters have a weight more than $h_o = 80$, all clusters are labeled as o-clusters with black outlines.

The clustering algorithm can compress information about densely clustered anoma-

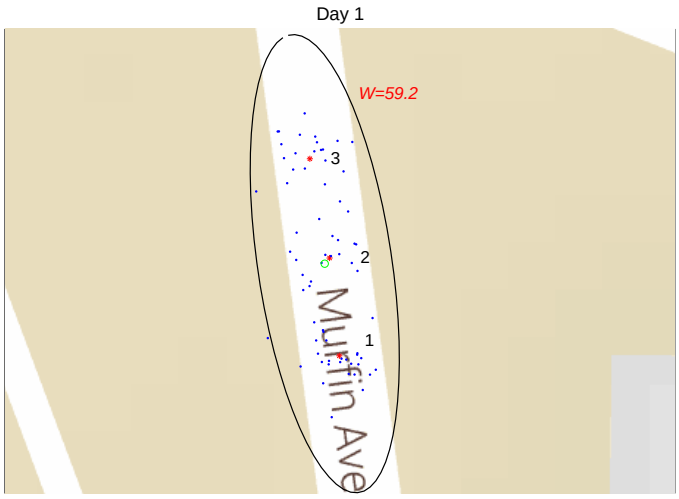


Figure 4.5: Information compression for anomalies 1, 2 and 3

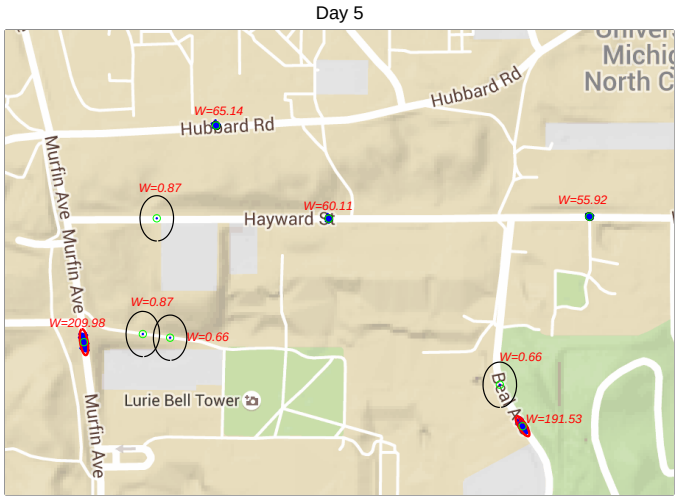


Figure 4.6: Snapshot at the end of day 5

lies as illustrated in Figure 4.5. Note that the maximum band can be compressed is controlled by the chi-square number $\chi^2_2(p)$. With a larger p, the algorithm is able to discriminate "closer" anomalies.

A snapshot at the end of day 5 is illustrated in Figure 4.6. With aggregated reports, the clusters formed by anomalies 1-3 and anomalies 6-8 have greater weights than h_o thus become m-clusters represented in red ellipsoids. At the same time, four outliers have been formed.

Based on the simulation setup from day 6, anomalies 6-8 are repaired and anomaly 10 is developed. A snapshot at the end of day 10 is shown in Figure 4.7. Since the outliers fail to become m-clusters in $T = 5$ days, the outliers are removed from the

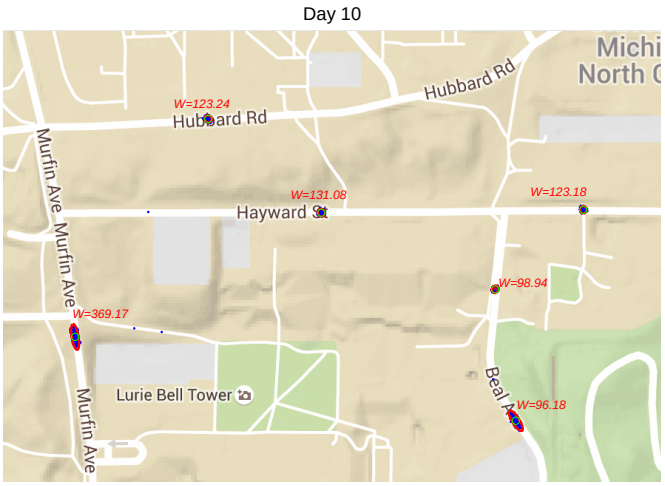


Figure 4.7: Snapshot at the end of day 10

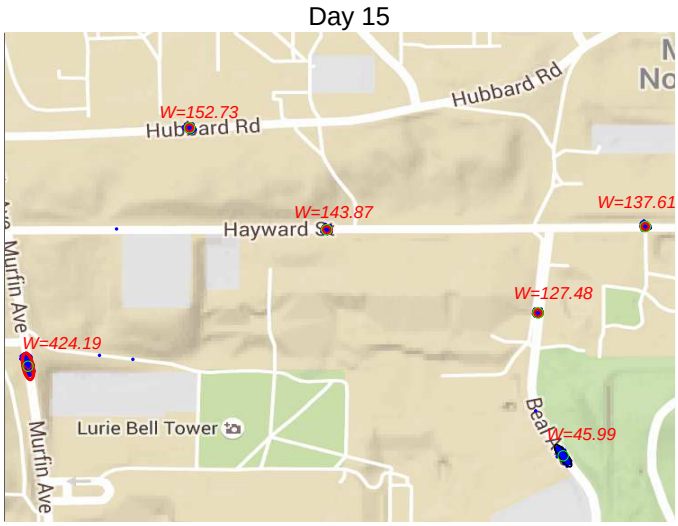


Figure 4.8: Snapshot at the end of day 15

outlier buffer. Also, since anomalies 6-8 are fixed and no new reports are aggregated, the weight decreases due to the forgetting function.

Finally, a snapshot at the end of day 15 is shown in Figure 4.8. The weight of the cluster formed by anomalies 6-8 continues decreasing based on the forgetting function. As a result, at the end of day 15, the weight becomes less than $h_m = 50$. The cluster is relabeled as an o-cluster and is moved to the buffer that is note shared with interested parties.

The simulation shows that the algorithm is able to handle outliers and can successfully capture the evolution of anomalies. The computation and storage requirements are also light.

4.5 Summary and discussion

In this chapter, the clustering algorithm ARSCA was developed to process anomaly reports that can be integrated in a Vehicle-to-Cloud-to-Vehicle framework. Cluster information was summarized in a feature vector that was recursively updated with new reports. The Mahalanobis distance was exploited to measure the similarity between a reported location and existing clusters. The obtained clusters were hyperellipsoids with arbitrary orientations. The Woodbury matrix inverse lemma was employed to facilitate the recursive computation of the covariance matrix inverse. The ARSCA algorithm can reject outliers and capture the evolving nature of road anomalies with a combined o-cluster and m-cluster strategy. The proposed algorithm can localize isolated anomalies and compress information for stretched anomaly segments with light memory and computation requirements.

Parameter selection was briefly discussed in this study. Future work will include a more comprehensive study on parameter selection based on GPS characteristics and traffic density information. Real-world data should be collected and machine learning techniques can be applied to find optimal parameters. Implementation with a V2C2V architecture will also be considered in our future work.

Part II

Cloud-aided automotive applications

CHAPTER 5

Cloud-aided semi-active suspension control

5.1 Introduction

Cloud computing can enable significant advancement in automotive control, diagnostics and prognostics capabilities. While traditional vehicle Electronic Control Units (ECUs) have limited computational characteristics, cloud computing will facilitate the implementation of more computationally-intensive predictive, optimal and cooperative driving strategies in a Vehicle-to-Cloud-to-Vehicle (V2C2V) control framework [4, 5, 16]. A general-purpose V2C2V architecture is illustrated in Figure 1.2.

In Chapter 3, a framework of simultaneous road profile estimation and anomaly detection was developed. This framework can be used to crowdsource road profile and anomaly information using onboard sensors. This information can then be stored in a cloud database and shared with other vehicles. In this chapter, we exploit the crowdsourced road profile and anomaly information for next-generation vehicle suspension control.

Suspension control is well researched, see *e.g.*, [36, 35, 60] and references therein, but not from the perspective of a V2C2V implementation. The benefits of such an implementation include the ability to plan and optimally respond based on lookahead road profile and pothole information that can be accessed in a distributed cloud

database. A simplified (yet representative) treatment of the problem is adopted in this chapter, based on a linear quarter-car model with the suspension set to one of a finite number of damping modes. Road input is modeled as a combination of a known deterministic component and an unknown stochastic component, where the deterministic part is obtained from vehicle crowdsourcing as discussed in Chapter 3 and the stochastic component models random mismatch between the true profile and estimated one. A simplified model is also developed for characterizing vehicle response to traversing a pothole. At the time scale of the suspension dynamics, the response to a pothole is modeled as an instantaneous change in wheel velocity, and we link this velocity change to the physical characteristics of the pothole and the vehicle.

We show that the problem of selecting the optimal damping mode reduces to integrating Riccati-type differential equations backward in time over the prescribed look-ahead horizon. The computations depend on vehicle parameters including vehicle speed as well as known deterministic road profile complemented with stochastic and pothole information. A V2C2V implementation is defined in which cloud resources are used to store vehicle and road profile data. A cloud-based computation pool is used to determine the optimal suspension mode, and vehicle-to-cloud communications are employed to exchange the information.

Figure 5.1 illustrates the envisioned cloud-based vehicle software agent that has access to stored vehicle parameters (M_{us}, M_s, k_{us}, k_s, R, $c_{s,i}$) defined in Section 5.2. This software agent receives vehicle state estimate $\hat{x}$, its error covariance matrix Σ, vehicle longitudinal velocity v_{car}, wheel speed ω, and GPS coordinates and time stamp t_{v2c}. It sends road profile information $\dot{r}_o$ to the vehicle-based state estimator along with the computed optimal suspension mode (i^*), and time stamp (t_{c2v}). The data packets to transmit between cloud and vehicle are small, so while a link must be available for real-time use, communication bandwidth need not be extensive.

This chapter is organized as follows. In Section 5.2 we describe the suspension dynamic model as well as the road input model. In Section 5.3 we define the optimal suspension mode selection problem and present its solution . The proposed architecture for V2C2V suspension control system is discussed in Section 5.4, along with the discussion of additional issues, such as the communication delay and suspension force saturation, that need to be considered in a full implementation. Simulation results are reported in Section 5.5, and Section 5.6 concludes this chapter.

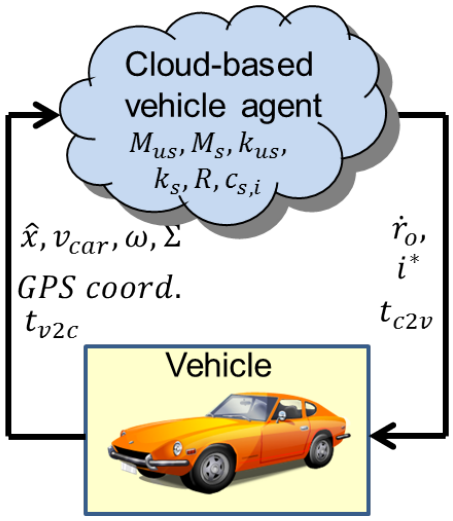


Figure 5.1: V2C2V Suspension Control System.

5.2 Suspension dynamic model and road input characterization

This section presents models of suspension dynamics, road profile, and pothole impact.

5.2.1 Dynamic model of semi-active suspension systems

Quarter-car models are often used for suspension control design [36, 35, 61, 62] because they are simple yet capture many important characteristics of the full-car model. A quarter-car model, with two degrees of freedom (DoF) as in Figure 5.2, is used. The car body is represented by sprung mass M_s, while the tire and axles are represented by unsprung mass M_{us}. The spring and shock absorber with adjustable damping ratio constitute the suspension system, connecting sprung (body) and unsprung (wheel assembly) masses. The tire is modeled as a spring with stiffness k_{us} and its damping ratio is assumed to be negligible in the suspension formulation. From Figure 5.2, we have the following equations of motion:

$$\begin{aligned} \dot{x}_1 &= x_2 - w - \dot{r}_o, \\ M_{us}\dot{x}_2 &= -k_{us}x_1 + k_s x_3 + c_s(x_4 - x_2), \\ \dot{x}_3 &= x_4 - x_2, \\ M_s\dot{x}_4 &= -k_s x_3 - c_s(x_4 - x_2), \end{aligned} \tag{5.1}$$

where x_1 is tire deflection from equilibrium; x_2 is unsprung mass velocity; x_3 is suspension deflection from equilibrium; x_4 is sprung mass velocity; $\dot{r}_o$ represents the deterministic vertical *velocity* disturbance due to the known road profile; w represents the zero-mean white noise stochastic road disturbance; c_s is the adjustable damping ratio of the suspension damper; and k_s and k_{us} are suspension and tire stiffness, respectively. Define $x = [x_1 \quad x_2 \quad x_3 \quad x_4]^T$. The suspension system model can then

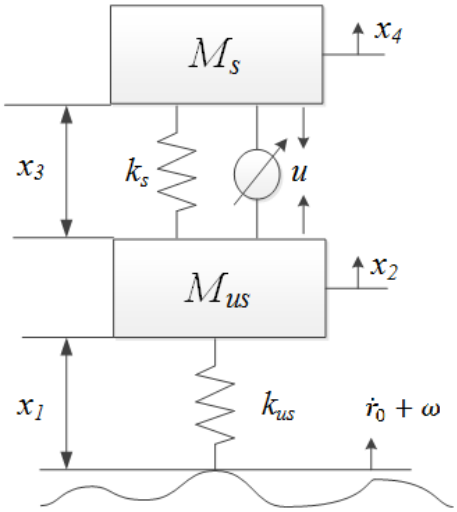


Figure 5.2: Semi-active suspension dynamics.

be written as:

$$\dot{x} = Ax + B\dot{r}_o + Bw, \tag{5.2}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_{us}}{M_{us}} & -\frac{c_s}{M_{us}} & \frac{k_s}{M_{us}} & \frac{c_s}{M_{us}} \\ 0 & -1 & 0 & 1 \\ 0 & \frac{c_s}{M_s} & -\frac{k_s}{M_s} & -\frac{c_s}{M_s} \end{bmatrix}, \quad B = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5.3}$$

The semi-active suspension system enables damping level to be varied, for instance, by changing the viscosity of a magneto-rheologic (MR) fluid [63], or by changing the orifice of a current-controlled valve in Telescopic-Hydraulic dampers [64]. A typical vehicle implementation features several suspension modes (*e.g.*, sport, comfort, *etc.*) selectable by a higher-level supervisory control scheme or manually by the driver. Each of these modes may employ a different dependence of the damping force versus the relative velocity or different feedback controller gains optimized for different performance metrics. In this chapter, we will treat a simplified yet representative case, where m individual suspension modes are defined that correspond to

different choices of suspension damping coefficient c_s. Treatment of suspension force saturation, beyond brief comments in Section 5.4, is relegated to future publications.

5.2.2 Road input modeling

Road profile information can be exploited for suspension mode optimization. Accurate road profile information is available from vendors who collect data using static or inertial profilers. Inertial profilers, composed of a vehicle mounted accelerometer, a laser height sensor and a distance measuring instrument, have been widely used for accurate road profile collection [65]. These sensors operate at approximately 16 kHz and can take about 15 readings per inch of vehicle travel at 60 mph. Those readings are filtered and can be stored on the cloud for future use. However, profilers are generally expensive to acquire and operate.

Chapter 3 develops a method to obtain the road profile by crowdsourcing the information from numerous vehicles. Collected road profile and anomaly information can be stored in cloud databases to keep the dynamic road information up-to-date.

To accommodate uncertainties in road profile collection, in (5.2) the road profile is modeled by a combination of deterministic ($\dot{r}_o$) and stochastic (w) components. This formulation permits modeling of road profile as purely stochastic (*i.e.*, $\dot{r}_o = 0$) if no other data is available or purely deterministic (*i.e.*, $w = 0$) if the road profile can be assumed known.

As is conventional, w is assumed to be white noise of unit intensity. Consequently, the true intensity of the noise scales matrix B, and since it is linearly dependent on vehicle speed, v_{car} [66], we can re-state the model in (5.2) as

$$\dot{x} = Ax + B\dot{r}_o + Bw, \tag{5.4}$$

where A is given in (5.3), and

$$B = B_0 \cdot \frac{v_{car}}{v_{car,0}}, \tag{5.5}$$

where B_0 is the B matrix of (5.3) scaled by the noise intensity at the vehicle velocity $v_{car,0}$. v_{car} is the actual velocity. Consequently, the computation of B in (5.5) involves obtaining B_0 and $v_{car,0}$ for the given road segment, and employing scaling based on actual vehicle speed v_{car}.

5.2.3 Pothole response modeling

Potholes are one of the main road hazards that can degrade ride comfort, cause tire damage or even accidents. Road input is often modeled as either deterministic (*e.g.*, given by sinusoids [64, 67, 68]) or stochastic [36, 37] without an explicit representation for hitting a pothole. In this chapter, we explicitly consider the response of hitting a pothole. In Chapter 2, a multi-phase dynamic model is developed to characterize the response of hitting a pothole. For the purpose of semi-active suspension control, this model is simplified in this chapter. At a time scale of vehicle suspension dynamics, hitting a pothole is modeled as a reset of the state

$$
\begin{aligned}
& x(t_p^+) = Reset[t_p, x(t_p^-)] : \\
& x(t_p^-) = \begin{bmatrix} x_1(t_p^-) \\ x_2(t_p^-) \\ x_3(t_p^-) \\ x_4(t_p^-) \end{bmatrix} \rightarrow x(t_p^+) = \begin{bmatrix} x_1(t_p^-) \\ x_2(t_p^-) + \Delta \\ x_3(t_p^-) \\ x_4(t_p^-) \end{bmatrix},
\end{aligned} \tag{5.6}
$$

where t_p denotes the time of hitting the pothole, t_p^- is immediately before, t_p^+ is immediately after, and Δ depends on vehicle physical parameters, wheel rotational speed and pothole parameters.

A model for wheel velocity change Δ is now described. Similar to Chapter 2, the geometry at the instant of hitting the rear edge of a pothole is shown in Figure 5.3, where l denotes the diameter of the pothole; d is the vertical difference between the front (left) and rear (right) edges of the pothole; R represents the effective wheel radius; r represents the maximum vertical wheel dropping. The case considered here occurs when the height of the rear edge is higher than that of the front edge. Similar analysis can be performed if the rear edge is lower than the front edge.

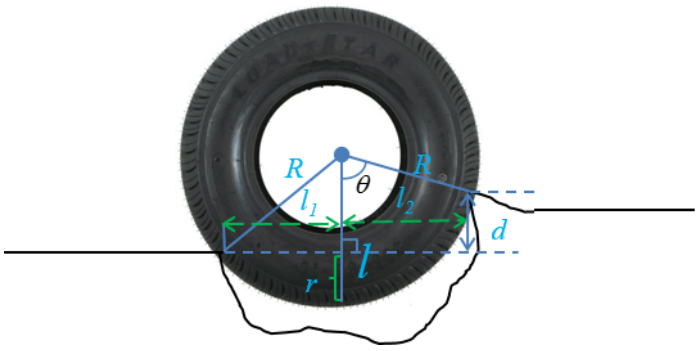


Figure 5.3: Pothole analysis.

The following equations are then satisfied,

$$\begin{aligned} l &= l_1 + l_2, \\ d &= \sqrt{R^2 - l_1^2} - \sqrt{R^2 - l_2^2}. \end{aligned} \tag{5.7}$$

From (5.7), l_1, l_2 can be determined if l, d and R are known. Then,

$$r = R - \sqrt{R^2 - l_1^2}. \tag{5.8}$$

Note that if the vertical difference between the front and rear edges d is 0, then

$$l_1 = l_2 = \frac{l}{2}.$$

To obtain the change in wheel velocity and impact force, we first find the wheel longitudinal and vertical velocities relative to the ground when hitting the rear edge. The kinematics are illustrated in Figure 5.4. The magnitudes of longitudinal and vertical relative velocities, v_x and v_y, respectively, satisfy

$$\begin{aligned} v_x =& v_{car} - \omega R \cos\theta = v_{car} - \omega R \, \frac{R - r - d}{R} \\ =& v_{car} - \omega(R - r - d), \end{aligned} \tag{5.9a}$$

$$\begin{aligned} v_y =& \sqrt{2rg} + \omega R \sin\theta = \sqrt{2rg} + \omega R \, \frac{l_2}{R} \\ =& \sqrt{2rg} + \omega l_2, \end{aligned} \tag{5.9b}$$

where v_{car} is the car center of mass velocity magnitude and ω represents the angular velocity magnitude of the wheel. The distances l_2 and r are determined from (5.7) and (5.8). Note that both v_{car} and ω can be obtained from onboard sensors. According to the model previously defined in (5.6),

$$\Delta = \sqrt{2rg} + \omega l_2. \tag{5.10}$$

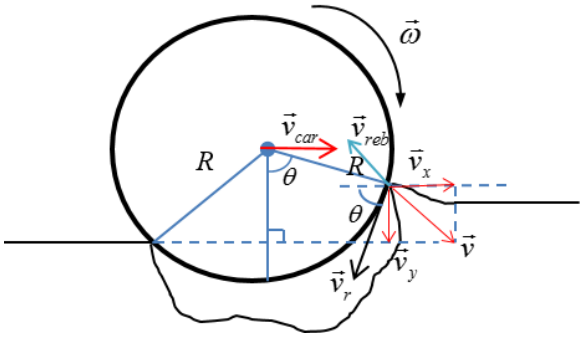


Figure 5.4: Kinematics when hitting the rear edge.

5.3 Performance prediction and mode selection

Consider the linear system model of suspension dynamics (5.4) and m discrete control modes representing different suspension damping levels corresponding to different values of damping coefficient c_s in (5.1). The suspension performance can be measured by the following quadratic cost function,

$$J = \frac{1}{2} \int_{t_0}^{t_0+T} q_1 x_1^2 + q_2 x_3^2 + \dot{x}_4^2 \, \mathrm{d}t,$$

where q_1, q_2 are weights and t_0 and T are the initial time and the look-ahead horizon, respectively. The above cost function trades off several competing objectives: passenger comfort (reflected in $\dot{x}_4$), small desired suspension stroke due to packaging constraints (reflected in x_3), and small tire deflection to ensure good vehicle handling (reflected in x_1). Note that horizon T may vary according to road conditions (*e.g.*, smooth roads vs uneven roads).

Let

$$y = \dot{x}_4 = Cx = \begin{bmatrix} 0 & \frac{c_s}{M_s} & -\frac{k_s}{M_s} & -\frac{c_s}{M_s} \end{bmatrix} x.$$

Then

$$J = \frac{1}{2} \int_{t_0}^{t_0+T} x(t)^{\mathrm{T}} Q x(t) \, \mathrm{d}t, \tag{5.11}$$

where

$$Q = Q_1 + Q_2 \cdot C^{\mathrm{T}} C,$$

and

$$Q_1 = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad Q_2 = 1.$$

Consider first the situation when the vehicle travels over road segment without a pothole. If control mode i is engaged, the corresponding cost functional over a finite time interval $[t_0, t_0 + T]$, is given by

$$J_i = \frac{1}{2} \int_{t_0}^{t_0+T} x^{\mathrm{T}}(t,i) Q x(t,i) dt, \tag{5.12}$$

with $x(t,i)$ denoting vehicle trajectory in mode i. A stochastic differential equation (5.1) can be written as

$$dx = A_i x dt + B r_o dt + B d\zeta, \tag{5.13}$$

where ζ is the zero mean and unit intensity (standard) Wiener process and $w =$"$\frac{d\zeta}{dt}$" in (5.2).

Since the system has a stochastic input, each realization of this stochastic input will lead to a specific value of the cost function. A single cost function value can be defined by taking the expected value,

$$J_i = E\left[\frac{1}{2} \int_{t_0}^{t_0+T} x^{\mathrm{T}}(t,i) Q x(t,i) dt\right]. \tag{5.14}$$

For (5.13), the infinitesimal generator for a given function, $V_i(t,x)$, defined on a time interval $t_{ini} \leq t \leq t_{fin}$ has the following form,

$$L_i V_i = \frac{\partial V_i}{\partial x}(A_i x + B \dot{r}_o(t)) + \frac{\partial V_i}{\partial t} + \frac{1}{2} trace(B^{\mathrm{T}} V_{i,xx} B), \tag{5.15}$$

with $V_{i,xx}$ denoting the Hessian matrix of second derivatives. If we can find a function V_i such that

$$L_i V_i + \frac{1}{2} x^{\mathrm{T}} Q x(t) = 0, \tag{5.16}$$

then applying Dynkin's formula [69] to both sides we obtain

$$E[V_i(t_{ini}, x(t_{ini})] - E[V_i(t_{fin}, x(t_{fin})] = E\Big[\int_{t_{ini}}^{t_{fin}} \frac{1}{2} x^{\mathrm{T}}(t,i) Q x(t,i) dt\Big]. \tag{5.17}$$

If the function V_i is known, the cost in (5.14) can be easily evaluated. Assuming that for $t_{ini} \leq t \leq t_{fin}$,

$$V_i(t, x, t_{fin}) = \frac{1}{2} x^{\mathrm{T}} P_i(t) x - x^{\mathrm{T}} g_i(t) + \phi_i(t), \tag{5.18}$$

and substituting this expression into (5.16) we obtain that $P_i(t)$, $g_i(t)$ and $\phi_i(t)$ satisfy the following ordinary differential equations,

$$\dot{P}_i(t) + A_i^{\mathrm{T}} P_i(t) + P_i(t) A_i + trace(B^{\mathrm{T}} P_i(t) B) \cdot I = -Q, \tag{5.19}$$

$$-\dot{g}_i(t) + P_i(t) B \dot{r}_o(t) - A_i^{\mathrm{T}} g_i(t) = 0, \tag{5.20}$$

$$\dot{\phi}_i(t) - \dot{r}_o^{\mathrm{T}}(t) B^{\mathrm{T}} g_i(t) = 0, \tag{5.21}$$

where $t_{ini} \leq t \leq t_{fin}$. To preserve the meaning of V_i being the cost-to-go, we impose additional terminal conditions,

$$P_i(t_{fin}) = 0,\ g(t_{fin}) = 0,\ \phi(t_{fin}) = 0. \tag{5.22}$$

The system (5.19)-(5.22) must be solved backward in time. After solving these equations, the expected cost function can be evaluated using (5.17).

Suppose next there is a single pothole over the look-ahead time interval $[t_0, t_0 + T]$ that the vehicle hits at a time instant t_p, $t_0 \leq t_p \leq t_0 + T$. Note that the pothole is at a specific spatial location and once it comes in range at time t_0, the time to hit it, t_p, depends on vehicle speed. Then the value of the cost (5.12) can be evaluated as

$$\begin{aligned} \tilde{J}_i &= E\Big[V_i(t_0, x(t_0), t_p)\Big] \\ &+ E\Big[V_i(t_p, Reset[t_p, x(t_p^-, i)], t_0 + T)\Big], \end{aligned} \tag{5.23}$$

with the reset map defined by (5.6) and (5.10), and $x(t_p^-, i)$ denoting the prediction of $x(t_p^-)$ in mode i. The on-board mode selection logic is based on choosing the mode

i^* that minimizes the predicted cost value, i.e.,

$$i^* \in argmin_{i=1,\cdots,m}\{\tilde{J}_i\}. \tag{5.24}$$

5.4 Towards V2C2V implementation

The V2C2V semi-active suspension control structure is presented in Figure 5.1. On the cloud, all pertinent data and parameters can be accessed and used to solve control equations (5.19), (5.20), (5.21). The state estimate $\hat{x}(t_s)$, the state estimation error covariance matrix, $\Sigma(t_s)$, vehicle velocity $v_{car}(t_s)$, and wheel speed $\omega(t_s)$ at time instant $t_s < t_p$ are communicated from the vehicle to the cloud. The current suspension mode, i^-, is assumed to apply over $[t_s, t_0)$ and the selection of the mode is optimized for time interval $(t_0, t_0 + T]$. On the cloud, the vehicle speed and wheel speed are assumed to be constant over the time interval $[t_s, t_0 + T]$, allowing the prediction of t_p.

Since system model (5.4) is linear, the distribution of $\bar{x}(t_0)$, *i.e.*, the prediction of $x(t_0)$, is Gaussian with the mean and the covariance matrix that can be computed by integrating the following differential equations,

$$\dot{\mu} = A_{i^-}\mu + B\dot{r}_o,\ \mu(t_s) = \hat{x}(t_s), \tag{5.25}$$

and

$$\dot{\Sigma} = A_{i^-}\Sigma + \Sigma A_{i^-}^{\mathrm{T}} + BB^{\mathrm{T}}, \tag{5.26}$$

with

$$\Sigma(t_s) = E[(x(t_s) - \hat{x}(t_s))(x(t_s) - \hat{x}(t_s))^{\mathrm{T}}].$$

Then, $\bar{x}(t_0) \sim N(\mu(t_0), \Sigma(t_0))$. Similarly, the prediction of $x(t_p^-)$, $\bar{x}(t_p^-)$, satisfies $\bar{x}(t_p^-) \sim N(\mu(t_p), \Sigma(t_p))$, where

$$\dot{\mu} = A_i\mu + B\dot{r}_o, \tag{5.27}$$

and

$$\dot{\Sigma} = A_i\Sigma + \Sigma A_i^{\mathrm{T}} + BB^{\mathrm{T}}, \tag{5.28}$$

with $\mu(t_0)$ and $\Sigma(t_0)$ as the initial conditions for (5.27)-(5.28). Finally, the prediction

of $x(t_p^+)$, $\bar{x}(t_p^+)$, satisfies

$$\bar{x}(t_p^+) \sim N\left(\mu(t_p) + \begin{bmatrix} 0 \\ \Delta \\ 0 \\ 0 \end{bmatrix}, \Sigma(t_p)\right).$$

The following proposition can be exploited in computing the value of the quadratic cost function (5.18) and in evaluating (5.23):
Proposition: Suppose $x \sim N(\mu, \Sigma)$ and V_i is given by (5.17). Then,

$$\begin{aligned} E[V_i(t, x, t_{fin})] &= \frac{1}{2} trace\big(P_i(t)\Sigma\big) \\ &+ \frac{1}{2}\mu^{\mathrm{T}} P_i \mu + \mu^{\mathrm{T}} g_i(t) + \phi_i(t). \end{aligned} \tag{5.29}$$

Proof. The proof follows by the properties of the expectation and algebraic manipulations,

$$\begin{aligned} E[V_i(t, x, t_{fin})] &= E\big[\frac{1}{2} x^{\mathrm{T}} P_i x - x^{\mathrm{T}} g_i(t) + \phi_i(t)\big] \\ &= E[\frac{1}{2} x^{\mathrm{T}} P_i(t) x] - \mu^{\mathrm{T}} g_i(t) + \phi_i(t). \end{aligned} \tag{5.30}$$

Now we calculate the expectation as follows,

$$\begin{aligned} E[\frac{1}{2} x^{\mathrm{T}} P_i(t) x] &= tr(E[\frac{1}{2} x^{\mathrm{T}} P_i(t) x]) = E[tr(\frac{1}{2} x^{\mathrm{T}} P_i(t) x)] \\ &= E[tr((\frac{1}{2} P_i(t) x x^{\mathrm{T}}))] = \frac{1}{2} tr(P_i(t) E[x x^{\mathrm{T}}]) \\ &= \frac{1}{2} tr(P_i(t)(\Sigma + \mu\mu^{\mathrm{T}})) = \frac{1}{2} tr(P_i(t)\Sigma) + \frac{1}{2}\mu^{\mathrm{T}} P_i(t)\mu. \end{aligned} \tag{5.31}$$

As a result, (5.30) and (5.31) lead to (5.29). □

Once the best suspension mode i^* is selected based on (5.24), it is transmitted to the vehicle. To solve the optimal mode selection problem, the most expensive part of the computation is to solve differential equations (5.19) - (5.22) and (5.25) - (5.28). For our specific suspension control problem, we apply the Runge-Kutta (RK4) method [70] with an error order of $O(h^4)$, where h is the integration step-size. In Equations (5.19) - (5.22), at a specific time stamp t, $P_i(t)$ is a four by four matrix; $g_i(t)$ is a four by one vector and $\phi_i(t)$ is a scalar. Thus, the three differential equations have order nine. In addition, means and covariances of the state are propagated as in (5.25) -

(5.28), which are ODEs of order ten. Suppose in a prescribed control horizon T, we have N discretization steps. By applying RK4, there are five computations for every step size. Since we have m modes, the computation steps would be $19 \times 5 \times m \times N$, which has complexity $O(mN)$. Suppose we choose a $T = 30$-sec horizon, a 0.01-second step size and $m = 5$ modes, there would be at least $1,425,000$ computation steps in 30 sec.

This complexity is not high and may be performed on-board if the information on road profile and potholes was available. The cloud-based implementation offers, however, significant opportunities to refine mode evaluation to account for different scenarios.

In particular, not every vehicle may hit a pothole (*e.g.*, the vehicle may be in a different lane or pass the pothole between the wheels). If probability of a vehicle hitting a pothole, p, is known, expression (5.23) can be changed to account for average performance,

$$\begin{aligned} \tilde{J}_i &= (1-p)E\Big[V_i(t_0, x(t_0), t_0+T)\Big] \\ &+ pE\Big[V_i(t_0, x(t_0), t_p)\Big] \\ &+ pE\Big[V_i(t_p, Reset[t_p, x(t_p^-, i)], t_0+T)\Big]. \end{aligned} \tag{5.32}$$

The complexity of this expression increases when vehicles might hit multiple potholes within the look-ahead horizon. Similarly, to ensure robustness of mode selection with communication latencies and GPS location errors, multiple scenarios with respect to these latencies can be considered so that the best average mode can be selected.

Finally, the definition of modes corresponding to damping coefficient values c_s in (5.1) is convenient as it keeps the model linear and enables evaluation of the cost (5.12) by integrating Riccati-like equations. If the modes are defined differently or suspension force saturation is taken into account, the evaluation of (5.12) may need to be performed by averaging results over different road profile scenarios. In the trivial case when $w = 0$, road profile is assumed to be fully known and there are no communication latencies, a single simulation suffices. Otherwise, multiple Monte Carlo-style simulations may be used to evaluate mode performance over different scenarios thus characterizing the response of the nonlinear system to stochastic inputs. An alternative approach exploits stochastic linearization techniques and is pursued in the conference paper [71].

5.5 Simulation results

The optimal suspension control problem has been analyzed in the previous section. By making use of stored deterministic profile data as a preview and solving three differential equations, costs of different modes can be predicted so that the best mode with minimal cost is chosen. In this section, we simulate the model and optimal control computations on a desktop computer. For simulations, a road segment over the future $T = 10$ sec horizon is modeled as follows,

$$\dot{r}_o(t) = \begin{cases} -0.2 \cdot \cos 2\pi t & 0.25s \leq t \leq 0.75s, \\ 0.05 \cdot \sin 2\pi t & 6.25s \leq t \leq 6.75s, \\ 0 & \text{otherwise.} \end{cases}$$

A unit intensity zero mean white noise input scaled by 0.01 is also included. See Figure 5.5 for the specified road profile. The simulation parameters are specified in Table 5.1.

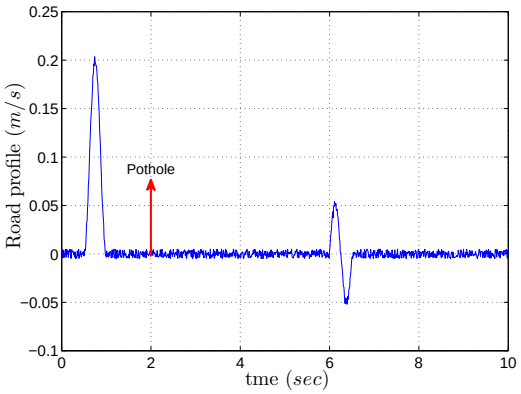


Figure 5.5: Road disturbances $(\dot{r}_0 + w)$.

Table 5.1: Simulation parameters

M_s	M_{us}	k_s	k_{us}	R	v_{car}	$v_{car,0}$
290 kg	60 kg	16800 N/m	19000 N/m	0.3 m	15 m/s	20 m/s

Five damping modes are modeled as specified in Table 5.2. The cost function weights are chosen as $q_1 = 10^6$ and $q_2 = 10^3$, which means the design is more handling-oriented.

For the initial condition $\hat{x}(0) = [0\ 0\ 0\ 0]^{\mathrm{T}}$, by integrating Riccati-like equations (5.19), (5.20) and (5.21) together with the propagated state means and covariances

Table 5.2: Damping Coefficients C_s (N· s/m)

Mode #1	Mode #2	Mode #3	Mode #4	Mode #5
200	400	800	900	1000 (max)

with (5.27) and (5.28), the cost of each mode is evaluated per (5.17) and (5.29) as:

$$J_1 = 6.7074, \quad J_2 = 4.8023, \quad J_3 = 4.1453, \quad J_4 = 2.8347, \quad J_5 = 2.6332.$$

In this case, the cloud will command the suspension system to select Mode 5 for the next 10 second horizon, for the given $\dot{r}_o(t)$ profile, as it has the lowest cost.

Now we consider the case where in addition to the previous road profile, there is also a pothole at time $t = 2s$ depicted as a red upward arrow in Figure 5.5. The pothole has a diameter of 0.3 m and the height difference between the front and rear edge is 0.05 m. As a result, there is a state reset as in (5.6). The cost of each mode is evaluated with (5.23) by integrating the Riccati-like equations (5.19) - (5.22), together with the propagated state means and covariances with (5.25) - (5.28). The costs are

$$J_1 = 7.9077, \quad J_2 = 5.7099, \quad J_3 = 5.2434, \quad J_4 = 3.6337, \quad J_5 = 3.6669.$$

In this case, Mode 4 is selected as it has the lowest cost.

5.6 Summary and discussion

A novel Vehicle-Cloud-Vehicle (V2C2V) implementation of a semi-active suspension control system was discussed. Cloud storage enabled detailed road profile data to be exploited to optimize suspension control decisions. In the proposed V2C2V implementation, pothole and road profile data were stored and accessed on the cloud. We showed that in the simplified form, the optimal semi-active suspension mode selection problem reduced to solving three differential equations and then evaluating the predicted cost over a prescribed horizon. Thus the optimal suspension mode for each road segment can be computed on the cloud and downloaded to the vehicle.

CHAPTER 6

Cloud-aided safety-based route planning

6.1 Introduction

According to a recent report [72] from the United States (US) National Highway Traffic Safety Administration, 33,561 people lost their lives on US roadways during 2012. The estimated property damage caused by auto accidents exceeded 200 billion dollars. To increase vehicle safety, various improvements in vehicle design and control are being pursued. For example, Adaptive Cruise Control systems [73, 74] are being implemented to automatically adjust vehicle speed to maintain a safe following distance. Lane Keeping Assist systems [75, 76] can alert the driver when the system detects that the vehicle is about to deviate from a traffic lane. In addition, the AAA Foundation for Traffic Safety has implemented the US Road Assessment Program to rank road risk into five levels, helping drivers plan their travel routes and assisting road agencies in improving the roads [77]. In this chapter, we propose road risk management through risk-aware route planning, with the goal of determining routes with low risk of accident as well as fast travel time. While modern vehicle navigation systems are able to generate optimal routes in terms of travel time, distance or fuel economy, in this chapter, we augment such cost terms with a safety-based risk metric.

Due to road geometries, distractors, traffic density, and other factors, some roads are more prone to accidents than others. To assess relative risk level and in turn implement safety-based route planning, a quantitative risk metric is required. Road risk index (RRI), which is modeled as a function of risk exposure, crash rate and severity, is accepted as a quantitative metric to reflect the relative crash risk as a function of road segment [78, 79, 80]. If available, RRI can also be used in safety-based route planning. However, crash rate records classified by route segment are not commonly available. As a result, some objective crash rate prediction models have been developed to predict crash rate based on road geometry. In [78], Miaou and Lum

studied the relation between highway crash rates and road geometries with Ordinary Least Squares (OLS) and Poisson Regression (PR) models. A Quantile Regression (QR) model was developed in [80] and demonstrated to have better prediction performance compared with the OLS and PR models proposed in [78]. In this chapter, a hybrid Artificial Neural Network (ANN) is developed and demonstrated to have better performance than the previously proposed models.

To develop the crash prediction model, an informative database with $30,682$ road segments and $144,821$ crashes from the Highway Safety Information System (HSIS) is processed. While many advanced modeling techniques such as polynomial regression, support vector regression and ANN are available, the ANN has been demonstrated to effectively model complex relationships and has been successfully applied in a variety of applications such as handwriting recognition [81], vehicle fuel economy modeling [82] and traffic prediction modeling [10]. From a practical implementation standpoint, ANNs have an important advantage in that they are familiar to automotive engineers, and represent proven technology that has been used in production vehicles [48]. Automotive engineers thus already understand ways to make the ANN-based solutions robust to real world variability and noise factors.

In this study, a hybrid ANN model is developed for road crash rate prediction. The inputs of this ANN include road geometries (segment length, curvature, grade *etc.*) and traffic density (annual average daily traffic). The outputs are the predicted crash rates categorized in three severity levels (fatality, injury and property damage only). Furthermore, the raw data is partitioned into three clusters with a fuzzy C-means clustering algorithm and three separate neural networks from data in each cluster are trained. This hybrid ANN model accurately predicts RRI as a function of road geometries and traffic information.

While this static hybrid ANN model can accurately predict RRI given road geometries and historical traffic information, real-time factors such as weather, time of day or day of the week, which can affect the risk level, need to be also considered. We therefore augment the prediction model with a dynamic layer to include such influences by employing multiplicative correction factors; these factors have not been considered in previous crash prediction models [78, 79, 80].

Safety-based route planning considers both travel time and risk in selecting the best route. The aim is to find a route with a minimum composite cost, which, in this chapter, is based on the travel time and RRI. Cost function weights reflect driver preferences. This problem can be specified as a multi-objective optimization problem that reduces to a Mixed Integer Programming (MIP) problem. A realistic route is

planned from Scioto Downs Inc, Ohio to Delaware, Ohio to demonstrate this route planning functionality. Several examples are presented to illustrate how the "safe" route can be different from the fastest route and how real-time factors can influence the resulting routes.

The proposed safety-based route planning functionality is envisioned within the framework of a V2C2V-equipped vehicle fleet. In our proposed V2C2V route planning architecture, safety is factored into planner decisions as illustrated in Figure 6.1. As

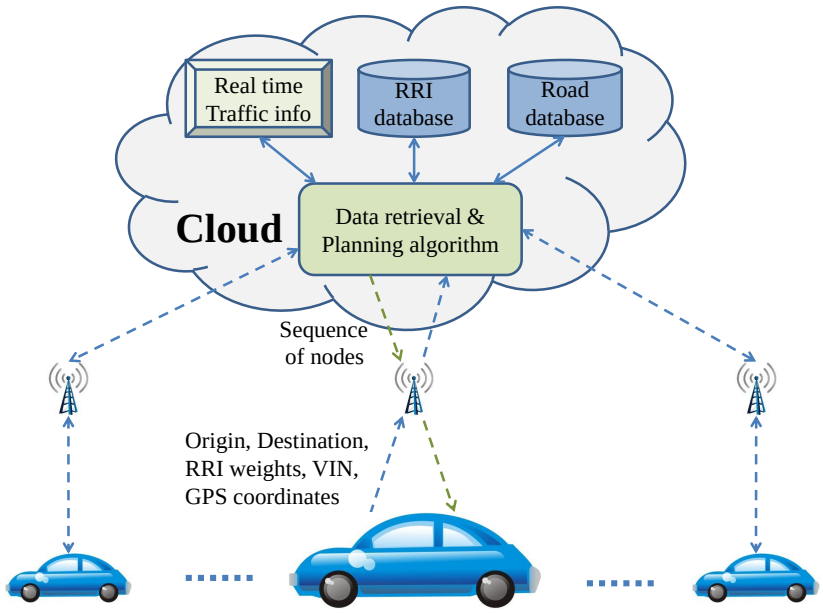


Figure 6.1: Architecture of cloud-aided safety-based route planning

in other V2C2V applications, vehicles communicate with the cloud through a wireless channel. The user initiates planning by providing the origin (current position by default), the destination, preferences that inform RRI weights and Vehicle Identification Number (VIN). If the driver fails to follow the planned route, the vehicle keeps sending its GPS coordinates to the cloud so that the planner can replan and update the routes accordingly. The cloud hosts our RRI model (see Section 6.2) and the planning algorithm. Real-time traffic and weather information can be obtained from a variety of sources. For example, INRIX XD Traffic delivers detailed traffic speeds every 800 feet (250 meters) across 4 million miles of roads in 37 countries [83]. While INRIX and other traffic data sources can ultimately be integrated within our route planner, incorporation of this data is beyond the scope of this study. We therefore generate representative data for our simulation studies.

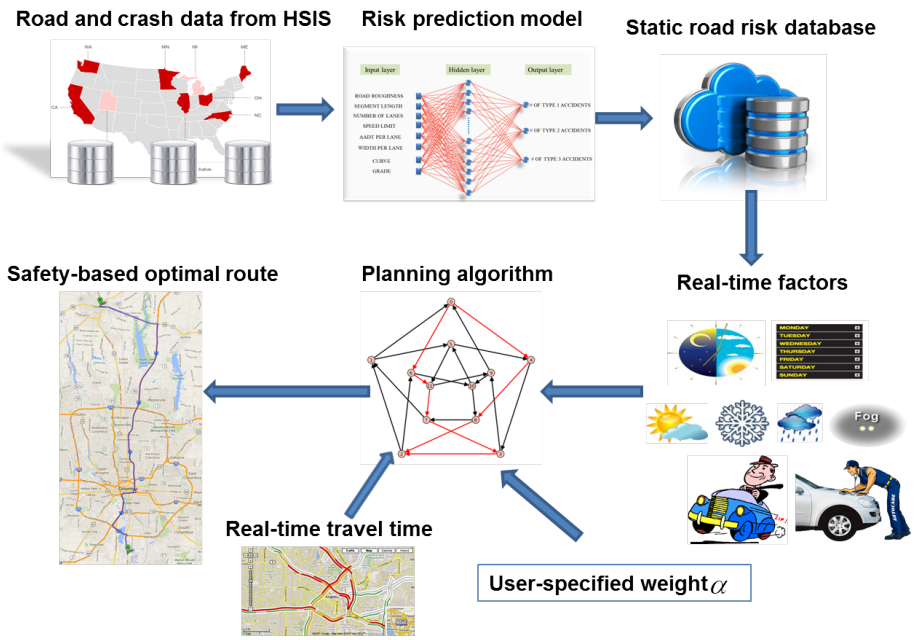


Figure 6.2: Safety-based route planning overview

An overview of our work is given in Figure 6.2. A road and accident database from the Highway Safety Information System (HSIS) is first processed to extract road characteristics. With a hybrid neural network, the data is then translated to a road risk database which is subsequently combined with real-time factors to provide safety-based route planning. The real-time factors include time of day, day of the week, real-time weather and traffic. Then a safety-based route planning is realized using a Mixed Integer Programming (MIP) algorithm with the real-time RRI, traffic and a user-specified weighting factor.

The contributions of this chapter include the following. First, by processing an informative road and accident database from HSIS, a hybrid ANN model is developed to efficiently predict crash rate. The hybrid ANN is demonstrated to have better performance than existing models presented in previous publications [78, 79, 80]. In addition, real-time factors such as weather, time of day and day of the week which have not been considered in previous models are incorporated as correction factors to a static RRI. An original framework for safety-based route planning is presented to provide optimal routes balancing safety and the traditional metrics such as travel time. Last but not least, real-world case studies are presented to demonstrate the applicability of the proposed route planning framework. While comprehensive time-based route planners and a number of road risk models have been studied previously, this work integrates risk and time together in a holistic route planning framework.

The chapter is organized as follows. Section 6.2 develops a neural network-based risk prediction model and compares it with QR and OLS models to quantify performance improvement. Model sensitivity to changes in inputs are analyzed and RRI dependence on various factors is established in Section 6.3. Real-time factors affecting road risk and their modeling are discussed in Section 6.4. The safety-based route planning problem is formulated using a Mixed Integer Programming problem in Section 6.5. Real-world case studies are presented to illustrate the safety-based route planning in Section 6.6, and Section 6.7 concludes the chapter.

6.2 Road risk modeling

6.2.1 Road risk index

A natural way to model the risk level of a road segment is to associate it with a crash rate (*e.g.*, number of accidents per year). However, this method has two deficiencies: it cannot be used for roads with no historical crash data, and it is not able to predict dependence on dynamic factors such as traffic density and road geometries.

In the past two decades, a different approach based on a Road Risk Index (RRI) has emerged. De Leur and Sayed have studied a driver-based subjective assessment of existing road risks [79]. Objective statistical crash prediction models have also been developed. Miaou and Lum proposed a Poisson Regression model [78] and Wu presented a Quantile Regression method [80]. These statistical risk models relate crash counts to influencing factors such as road geometries and traffic information and can be used to predict crash rate, generate road risk indices, and suggest improved road designs.

In [79, 80], RRI is modeled as a function of exposure, crash rate and severity. Exposure represents the amount of activity in which a crash may occur; Vehicle Miles Traveled (VMT) is the most common measure of exposure. Crash rate is the number of crashes per unit of exposure. It is proportional to the probability of a crash. Crash severity reflects the consequence of crashes in terms of injuries and property damage. Crash severity is officially classified in three categories: fatality, injury, and property damage only [72]. Since we develop road risk index for individual drivers, we only consider crash rate and severity in defining RRI so that

$$RRI(i) = \mathcal{F}\Big(\sum_{j=1,2,3} n_{ij} S_j / AADT_i\Big), \tag{6.1}$$

where i is road segment number; $RRI(i)$ is the risk index of road segment i; $j = 1, 2, 3$, represents the severity level of fatality, injury and property damage only, respectively; n_{ij} is the predicted number of accidents of level j to occur over road segment i in a certain period of time, *e.g.*, during one year; S_j is the cost of an accident of type j; and $\mathcal{F}(\cdot)$ is a function that scales the cost of most homogenous segments to an index between 0 and 100. According to a report from the Bureau of Transportation Statistics (BTS) [80], the average costs of fatality, injury, and property damage accidents are $S_1 = \$4,113,956$, $S_2 = \$144,291$, and $S_3 = \$6,783$, respectively. The $AADT_i$ in (6.1) represents the annual average daily traffic on road segment i. In this study a linear function $\mathcal{F}(x) = x/10$ is used and (6.1) becomes

$$RRI(i) = \frac{1}{10} \sum_{j=1,2,3} n_{ij} S_j / AADT_i. \tag{6.2}$$

We develop a data-driven model to predict the number of accidents n_{ij} as a function of road segment geometry, traffic conditions and weather conditions.

6.2.2 HSIS database

Road and crash data from the Highway Safety Information System (HSIS) [84] are utilized to develop a crash rate prediction model. The HSIS has a multi-state database that contains crash, roadway inventory, and traffic volume data for a select group of States (Washington, California, Minnesota, Illinois, Ohio, Maine and North Carolina). Crash data and road information for Ohio from 2006 are used in this study to develop a risk prediction model. The Ohio data provided by HSIS include the following:

- Accident (accident, vehicle and occupant),
- Roadway inventory file, denoted *Roadlog*,
- State supplemental inventory, containing curve and grade.

Accidents are recorded case-by-case. Separate files containing vehicle and occupant information can be linked to accident data for specific cases using the accident case number. The accident data can also be linked to a *Roadlog* file using three common variables: county, route number, and milepost. Unlike an accident file, each record in the *Roadlog* file contains information on a *homogeneous* section of the roadway (*i.e.*, a stretch of road that is consistent in terms of certain characteristics), with each new section being defined by a new beginning reference point. Each record in

the *Roadlog* file contains current characteristics of the road system including surface type and width, shoulder and median information, lane information, etc.

Data on 30,682 *homogeneous* road segments and 144,822 accidents for the year 2006 in Ohio were processed and eight features were extracted as inputs to the model. These features include:

- *Pavement Roughness*, defined as the International Roughness Index (IRI). Generally, a road segment is smooth if its IRI is under 100 and rough if its IRI is above 180. IRI varies from 30 to 562.
- *Speed limit*, the officially marked speed limit of the road segment in mile per hour (mph). Speed limit varies from 20 to 65 mph.
- *Segment length*, the length of a homogenous road segment in miles. It varies from 0.01 to 16.97 miles.
- *Number of lanes*, representing the total number of lanes over both travel directions. The range is from 1 to 11.
- *Annual average daily traffic per lane (AADT)*, reflecting average traffic density. AADT per lane varies from 27.5 to 56,980.
- *Width per lane*, varying from 7 to 36 feet.
- *Curvature*, the *maximum* degree of curvature in that road segment. An n-degree curve turns in the forward direction by n degrees over 100 ft. Curvature ranges from 0 to 270 degrees.
- *Grade*, representing the largest slope over the road segment in percent, which is the tangent of the angle of inclination times 100. Grade ranges from 0 to 20 percent.

To build an objective model relating database features to accident rate, the above eight features are used as the explanatory variables and accidents of each type in Ohio in 2006 are the response variables. The statistics of these variables are listed in Table 6.1. We next explore models to capture relationships between explanatory and response variables.

Table 6.1: Explanatory and response variable statistics

Variables	Mean	Std.Dev	Min	Max
Response variable				
Number of fatality crashes	0.2	0.14	0	2
Number of injury crashes	12.3	6.94	0	72
Number of property-damaged-only crashes	34.7	17.81	0	183
Explanatory variables				
Pavement roughness (IRI)	126.9	60.77	30	562
Speed limit (mph)	47.03	11.36	20	65
Segment length ($miles$)	0.62	1.07	0.01	16.97
Number of lanes	2.82	1.27	1	11
AADT per lane	3790.4	3929	27.5	56980
Width per lane (ft)	12.37	2.75	7	36
Grade (percent)	1.70	3.33	0	20
Curve ($deg/100ft$)	5.65	17.81	0	270

6.2.3 Crash rate prediction

In this subsection, we develop and compare Ordinary Least Squares, Quantile Regression and neural network models to predict the crash rates based on explanatory variables in Table 6.1. Let $\{x_i, y_{ij}\}$ represent the observations for road segment i, $i = 1, 2, \cdots, 30682$. x_i is the vector of explanatory variables as in Table 6.1, and y_{ij} is the number of level j accidents occurring on road segment i, where $j = 1, 2, 3$ represents fatality, injury and property damage only, respectively. We note that the difference between y_{ij} and n_{ij} in (6.1) is that y_{ij} represents database statistics while n_{ij} is the prediction with a certain model.

6.2.3.1 Ordinary least squares (OLS)

OLS is a linear regression model linking the explanatory variable x_i and response variable y_{ij},

$$y_{ij} = x_i^{\mathrm{T}} \beta_j + \epsilon_{ij}, \tag{6.3}$$

where $\beta_j \in \mathbb{R}^8$ is the vector of identified parameters and ϵ_{ij} represents fitting errors. Note that since there are three separate response variables denoting each one of severity levels, we have three sets of linear models for $j = 1, 2, 3$. For each linear model, β_j minimizes the sum of fitting error squared and is given by

$$\hat{\beta}_j = (X^{\mathrm{T}} X)^{-1} X^{\mathrm{T}} Y_j, \tag{6.4}$$

where $Y_j \in \mathbb{R}^{30682}$, and $X \in \mathbb{R}^{30682\times 8}$ is a matrix of regressors. With $\hat{\beta}_j$ identified, the number of accidents of level j can be predicted by

$$n_{ij} = x_i^{\mathrm{T}} \hat{\beta}_j. \tag{6.5}$$

6.2.3.2 Quantile regression (QR)

In [80], Wu developed a crash rate prediction model with QR. QR provides a more comprehensive representation of effects of the explanatory variables on the response variable. QR models the relation between a set of explanatory variables and specific percentiles (or quantiles) of the response variable. It specifies changes in the quantiles of the response. A QR model generates an ensemble of models for each accident type j, $j = 1, 2, 3$, corresponding to specified quantiles. For each severity level j and a quantile τ, the linear model with parameter vector $\beta_{j\tau}$ can be obtained by solving

$$\min_{\beta_{j\tau}\in\mathbb{R}^8} \quad \sum_{i=1}^{30682} \rho_\tau(y_{ij} - x_i^{\mathrm{T}} \beta_{j\tau}), \tag{6.6}$$

where $\rho_\tau(\cdot)$ is the tilted absolute value function as in Figure 6.3. The resulting

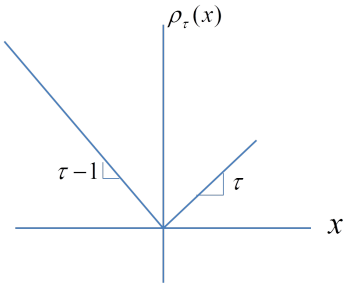


Figure 6.3: The function $\rho_\tau(x)$ in 6.6

minimization problem can be solved by linear programming methods [85].

With the models of corresponding quantiles, predictions can be made by dividing quantiles into intervals and associating each interval with a quantile in the middle of the interval. Let x_i be the vector of explanatory variables of road segment i in Table 6.1 and τ_k be the middle quantile of the kth interval, $k = 1, 2, \cdots, N$. With $\hat{\beta}_{j\tau}$ identified, the number of accidents of level j can be predicted by

$$n_{ij} = \sum_{k=1}^{N} p_k \cdot x_i^{\mathrm{T}} \hat{\beta}_{j\tau_k}, \tag{6.7}$$

where N is the number of divided intervals, and p_k is the probability of the occurrence of crash rate falling into the kth quantile interval.

6.2.3.3 Neural network model

A neural network model mapping explanatory variables in Table 6.1 to the number of accidents of each type is illustrated in Figure 6.4. The network uses a single layer with 20 neurons in the hidden layer. The Matlab^{TM} neural network toolbox was used for network training and testing.

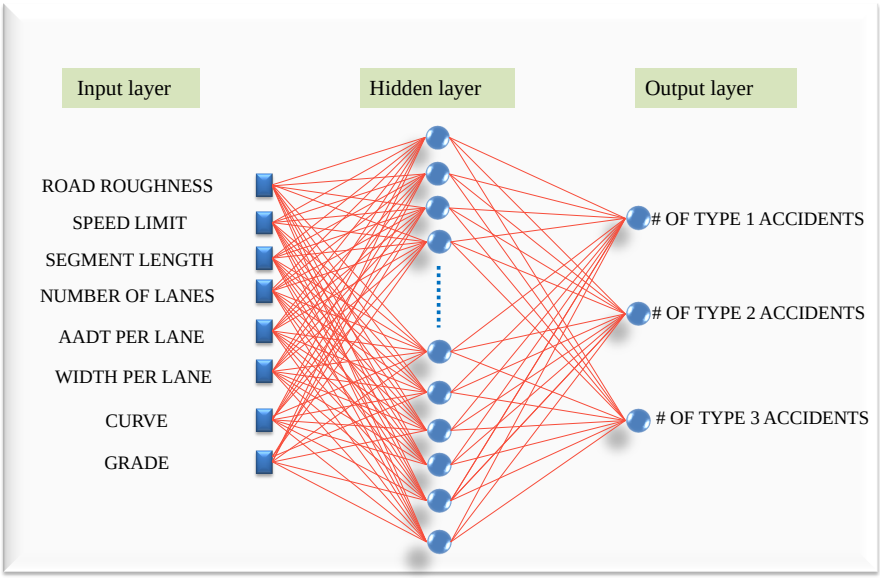


Figure 6.4: Neural network for crash rate prediction

6.2.3.4 Comparison

We next compare the performances of the aforementioned models. A dataset with 30,862 road segments was randomly partitioned such that 70 percent of the samples were used to train the models and 30 percent were reserved for cross-validation. All models used the same training and testing data. Root mean square error (RMSE) is selected as the performance criterion, where

$$RMSE = \sqrt{\sum_{j=1,2,3} \sum_{i=1}^{30682} (n_{ij} - y_{ij})^2}. \tag{6.8}$$

In (6.8), n_{ij} is the predicted number of level j accidents for a road segment i, and y_{ij} is the observation from data. The performance comparison is illustrated in Ta-

Table 6.2: Model performance comparison

Model	RMSE (acc/yr)
Ordinary Least Squares	5.17
Quantile Regression	3.51
Neural Network	3.35

ble 6.2, where *acc/yr* is number of accidents per year which is the unit of RMSE. The comparison shows that the proposed neural network model has the best RMSE. To further improve the model performance, a hybrid neural network architecture is considered next.

6.2.3.5 Hybrid neural network model

A hybrid neural network scheme is shown in Figure 6.5. Input data were first partitioned into three groups with a fuzzy C-means clustering algorithm [86]. Then for each cluster a separate neural network was developed. The RMSE of the three networks are 3.12, 3.24 and 2.96, respectively, which is an improvement over the conventional neural network. Based on the number of accidents listed in Table 6.1, the predicted error is about five percent. Each of these three networks has three outputs corresponding to the three crash ratings. To compute the output of the hybrid neural network, we first compute the Euclidean norms (distance) between the input and cluster centers. Then we apply the neural network corresponding to the minimum-distance cluster. As a result, accident rates of each type can be predicted and the risk index can be generated using (6.1). Since the best RMSE results are obtained with the hybrid neural network, it is chosen as the model for the analysis and optimization to follow.

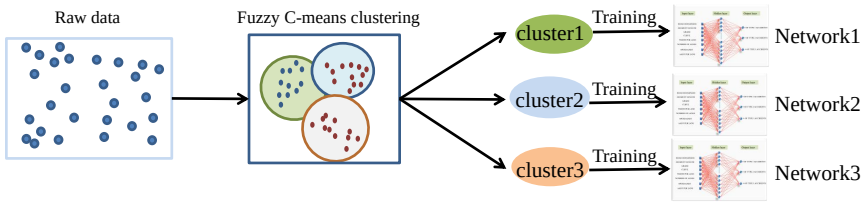


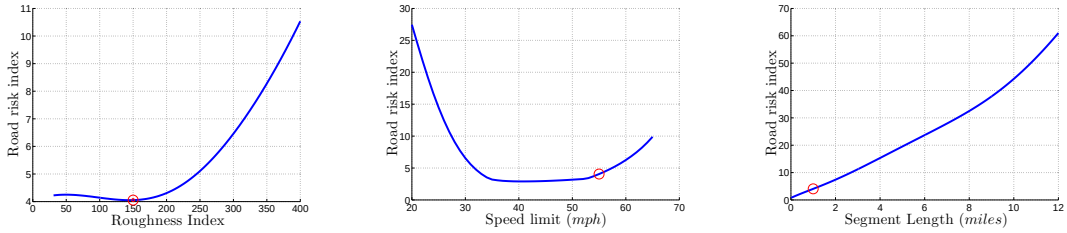
Figure 6.5: A hybrid neural network model

Note that the developed model assumes the underlying data set is stationary.

This assumption can be relaxed with the V2C2V implementation, where the cloud can constantly adapt road changes and update the model.

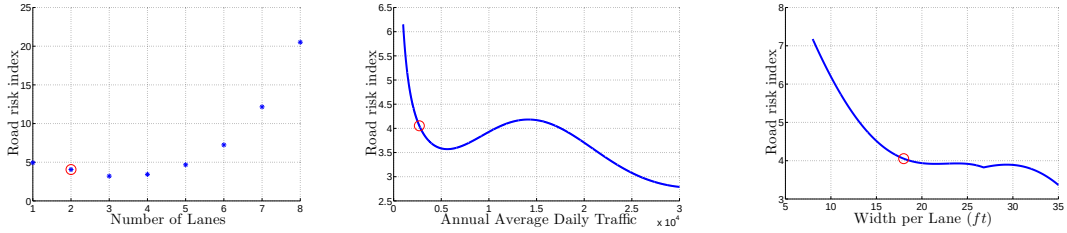
6.3 Model sensitivity analysis

In this subsection, we use the hybrid neural network model to analyze the sensitivity of RRIs to changes in the inputs around nominal values. Sensitivity analysis can reveal interesting trends and, provided the trends are reasonable, help build confidence in the model. The nominal road segment corresponds to 150 IRI, 55 mph speed limit, 1 mile length, 2 lanes, 2700 AADT per lane, an 18 feet lane width, 13.5 degrees of curvature, 3 percent grade and no-adverse weather. By varying each variable one at a time, we obtain the results in Figure 6.6(a)- Figure 6.8(b), where red circles represent the nominal values.



(a) RRI dependence on road roughness (b) RRI dependence on speed limit (c) RRI dependence on road segment length

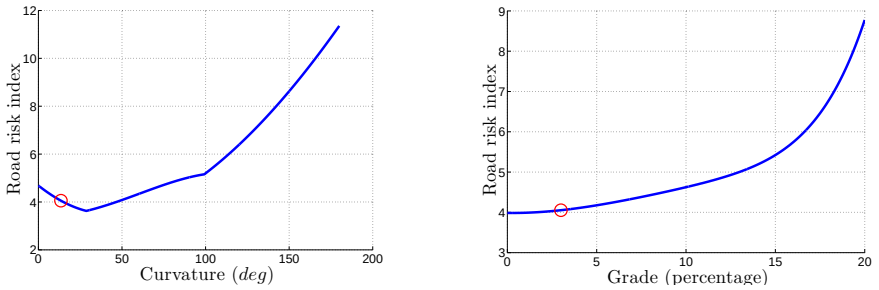
Figure 6.6: RRI dependence on roughness, speed limit and segment length



(a) RRI dependence on number of lanes (b) RRI dependence on AADT per lane (c) RRI dependence on width per lane

Figure 6.7: RRI dependence on number of lanes, AADT and lane width

Figure 6.6(a) shows that RRI varies negligibly when IRI is under 200 and it increases abruptly with an approximate slope of 3/100 RRI per IRI when IRI is above 200. Figure 6.6(b) shows that segments with a speed limit between 35 and 55 mph have the lowest risk index. Figure 6.6(c) shows that the risk index increases



(a) RRI dependence on curvature (b) RRI dependence on grade

Figure 6.8: RRI dependence on curvature and grade

almost linearly as road segment length increases with an approximate slope of 4.5 RRI per mile. Figure 6.7(a) indicates that segments with more lanes tend to carry higher risk. This result assumes that the AADT per lane is fixed at a nominal value. More lanes thus correspond to more traffic and higher risk. Figure 6.7(b) shows that as AADT per lane increases, the risk tends to first decrease and then increase between 8,000 and 15,000. This is not surprising given that RRI in (6.10) is proportional to the probability of having an accident. As a result, more traffic may lead to more total accidents but lower probability of an accident for any particular vehicle. Figure 6.7(c) indicates that the wider the lane, the safer the road segment. Figure 6.8(a) shows risk first decreases and then increases with curvature and Figure 6.8(b) shows that higher slope leads to higher risk.

Note that the model has been developed for homogeneous road segments, in which the number of lanes, lane width, speed limit *etc.*, do not change. The homogeneous road segments are atomic, *i.e.*, any route is composed of these segments. However, for the purpose of route planning, it is more convenient and yields a more tractable search space to represent road segments as edges between intersections, where each edge may consist of multiple homogeneous segments. The RRI of an edge between intersections can be obtained by summing RRIs over homogeneous segments. Note that for those edges that include only a part of a homogeneous segment, according to Figure 6.6(c), we can add the RRI according to the proportion of the homogeneous segment included. An illustrative example is shown in Figure 6.9. The ith edge with vertices V_i and V_{i+1} consists of homogenous road segments s_{m+2}, s_{m+3}, s_{m+4} and part of road segments s_{m+1} and s_{m+5}, then the RRI of the edge i is computed as

$$\begin{aligned} RRI(i) = 1/2 \cdot RRI(s_{m+1}) + RRI(s_{m+2}) + RRI(s_{m+3}) \\ + RRI(s_{m+4}) + 2/3 \cdot RRI(s_{m+2}). \end{aligned} \tag{6.9}$$

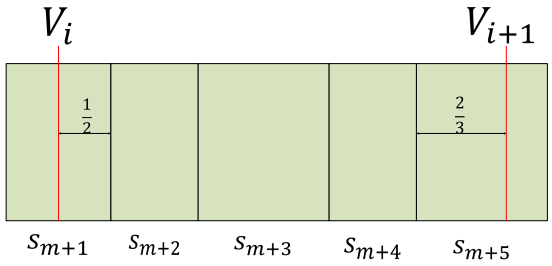


Figure 6.9: Edge RRI composition of multiple homogeneous road segments

Remark 6.1. RRIs of homogenous road segments are summed to obtain the overall RRI of a road edge between intersections as in (6.9). Note that the summation is valid if we assume that the events of having an accident over the road segments are mutually independent. Since RRI is a linear function of probabilities of accidents as in (6.1), it is straightforward to check that the summation of RRIs directly follows from the definition.

6.4 Real-time factors influencing road risk

Besides road geometries and historical traffic density, real-time factors can have a significant impact on road risk. For example, a snowy road segment is more risky than a normal road segment and driving at midnight on the weekend can be more dangerous because of greater likelihood of impaired and fatigued drivers. Unfortunately, these factors were not available in the HSIS database and could not be treated as ANN inputs in the same way as in the previous section. In this section, we describe how such factors are incorporated into the model.

6.4.1 Weather

Weather can have a significant impact on traffic safety. Adverse weather conditions such as snow, fog and rain can worsen the driving environment. In the HSIS crash database, accidents are recorded with the weather conditions present when the accident happened. To see how those adverse weather conditions affect crash risk, we investigate the average number of accidents happening in each recorded weather condition. By processing raw accident data, we determine the average number of daily accidents in Ohio in 2006 versus weather conditions, as shown in Table 6.3. The average number of daily accidents over all weather conditions was 396.77. We apply a

Table 6.3: Average daily accidents under different weather conditions

Weather conditions	Not adverse	Snow	Rain	Fog
Average daily accidents	367.89	452.68	415.32	409.58

correction factor as the ratio of number of accidents at specific weather conditions to total number of daily accidents. For instance, if we obtain a RRI of 4.5 for a segment from the neural network model and it is currently snowy or predicted to be snowy, then the RRI is corrected to

$$4.5 \times \frac{452.68}{396.77} = 5.13.$$

6.4.2 Time of day, day of the week

According to a report from the National Highway Traffic Safety Administration (NHTSA) [87], the later hours of the weekend and late afternoon to evening on the weekdays tend to be the riskiest periods for driving. The HSIS accident database includes hour of the day and day of the week of each accident. The distribution of number of accidents versus day of the week is illustrated in Figure 6.10. As shown in Figure 6.10, the number of accidents occurring during a week is evenly distributed except that Fridays and Sundays are a little above and below average, respectively. Figure 6.11 illustrates the distribution of weighted number of accidents over time of day for weekdays and weekends. The weighted number of accidents is defined as the number of accidents divided by the AADT distribution over a given one hour time interval, as shown in Figure 6.12. Each bin in the Figure 6.12 histogram corresponds to a one-hour period. For example, the bin centered at 0.5 describes the period from 0:00 am to 0:59 am. These figures reveal similar conclusions to these presented in [87]. For instance, accidents are most likely in late nights during the weekend. This fact can be explained by a considerable number of drivers in the late night of the weekend being fatigued or impaired. Based on Figure 6.11, a correction factor can be defined as the ratio of the weighted number of accidents to the average. For instance, travel at 1:30 am on Saturday will have a correction factor of $\frac{580}{304} = 1.91$, where 580 is the weighted number of accidents in the 1-2 am period in the weekend and 304 is the average weighted number of hourly accidents over a week.

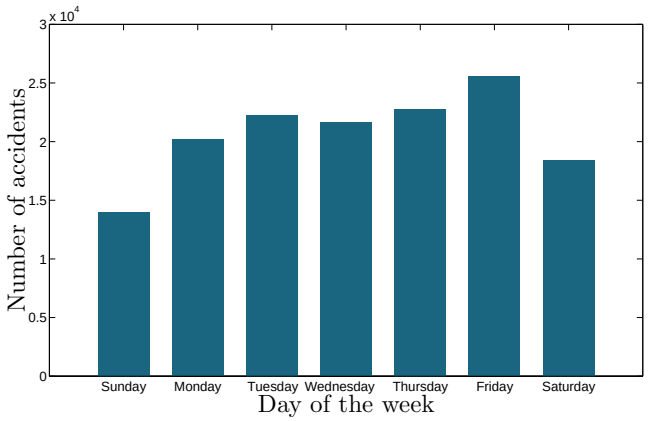


Figure 6.10: Accident distribution over day of the week

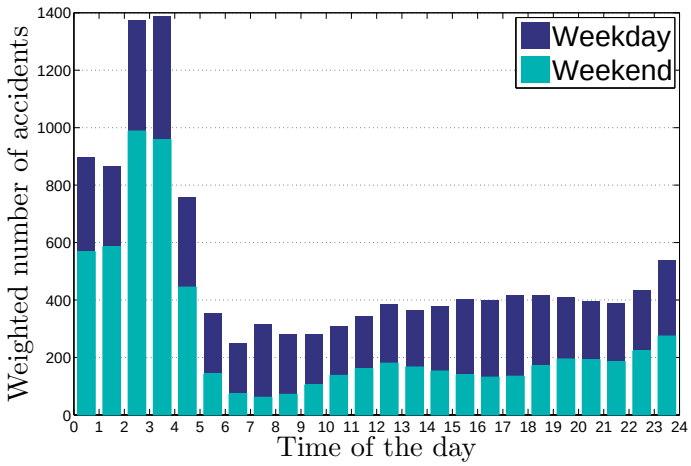


Figure 6.11: Accident distribution over time of day

6.4.3 Driving style and vehicle conditions

Driver characteristics and vehicle conditions are also important factors influencing safety. Aggressive drivers are more prone to accidents than non-aggressive drivers. Older or improperly maintained vehicles may also result in more accidents. Onboard driving style identification [88, 89] and vehicle condition monitoring [90, 91] are becoming available. In this study, we assume compounding correction factors of 1.2 and 1.1 for an aggressive driver and a poorly maintained vehicle, respectively.

6.5 Safety based route planning

Route planning is a network flow problem [92]. A road network can be modeled as a directed graph as shown in Figure 6.14(a). Intersections and road segments are

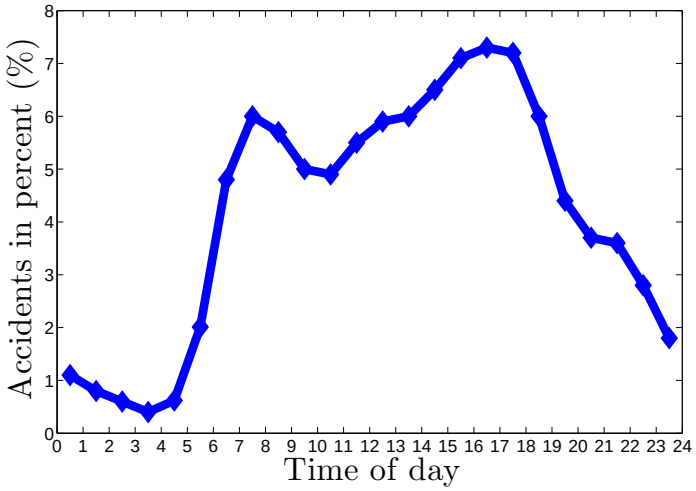


Figure 6.12: AADT distribution over time of day (1996-2006) [2]

abstracted as nodes and edges, respectively, in this directed graph. A route planner generates an optimal route in the transportation network based on specified cost functions. Traditional planners mainly consider time, distance or fuel economy. Route planners have been applied to a variety of applications. For example, evacuation planning has been studied for victim evacuation [93, 94, 95]. Our study focuses on traditional route planning but with time and our new road risk index cost metrics.

A variety of searching/optimization algorithms can be exploited to solve the road-based route planning problem, e.g., Dijkstra's algorithm [96], A^* algorithm [97] and the Genetic Algorithm [98]. In this chapter, we apply a Mixed Integer Programming (MIP) approach as a prototype. This choice is motivated by the existence of commercial MIP solvers such as Cplex [99] or Gurobi [100] that have been deployed in cloud-based applications. The MIP based approaches have also been previously used for route planning applications as in [101].

6.5.1 Problem Formulation

Safety-based route planning is a multi-objective graph traversal problem. The goal is to find an optimal route that minimizes a weighted sum of cumulative travel time and RRIs. The problem can be defined as follows.

Problem 6.1. For a directed graph $G = \{V, E\}$, where V is the set of vertices and E is the set of edges, we assign each edge $e_{i,j}$ a pair $(t_{i,j}, r_{i,j})$, where $i, j \in V$ are two adjacent vertices; $e_{i,j} \in E$ represents an edge from i to j; the pair $(t_{i,j}, r_{i,j})$ represents expected real-time travel time red from the cloud and RRI of $e_{i,j}$, respectively. Let

s and d be the start and destination vertices, respectively. We denote by $\mathcal{P}$ the set of all paths from s to d, where a path P is a sequence of vertices from s to d, i.e., $\{s, v_1, v_2, \ldots, v_n, d\}$. The problem is to find an optimal route that minimizes the following cost function,

$$\min_{P \in \mathcal{P}} t_{s,v_1} + t_{v_1,v_2} + \cdots + t_{v_n,d} + \alpha \times (r_{s,v_1} + r_{v_1,v_2} + \cdots + r_{v_n,d}), \tag{6.10}$$

where α is the weight on cumulative RRI reflecting the relative driver weighting of route safety and travel time.

6.5.2 Optimal route planning by Mixed Integer Programming

In this chapter we employ Mixed Integer Programming (MIP) [102, 103] to solve Problem 6.1. For each edge $e_{i,j}$, we assign a binary decision variable $x_{i,j} \in \{0, 1\}$ which determines if the edge is travelled, and we redefine the problem as follows,

$$J = \frac{1}{2} \sum_{e_{i,j} \in E} x_{i,j} t_{i,j} + \alpha x_{i,j} r_{i,j} \rightarrow \min_{x_{i,j}} \tag{6.11}$$

subject to:

$$\sum\nolimits_{e_{s,v} \in E} x_{s,v} = 1, \tag{6.12}$$

$$\sum\nolimits_{e_{s,v} \in E} x_{s,v} = 1, \tag{6.13}$$

$$\sum\nolimits_{e_{i,j} \in E} x_{i,j} = \sum_{e_{j,k} \in E} x_{j,k}, \forall j \in V, j \neq s \,\&\, j \neq d. \tag{6.14}$$

Constraints (6.12) and (6.13) imply that there is only one edge in the path from the start node and only one to the destination. Constraint (6.14) dictates that each vertex in-between has the same number of incoming and outgoing edges.

We use Cplex [99] to solve the above problem. Note that since route optimization can be performed on the cloud, powerful solvers such as Cplex can be made readily available and can reasonably be expected to quickly generate optimal long-distance routes.

6.6 Route planning case study

In this section, we consider a real world route planning case study. As illustrated in the Google Maps snapshot in Figure 6.13(a), our goal is to plan a route from Scioto

Downs Inc, Ohio to Delaware, Ohio. To plan the route, we first abstract the road network into the graph as shown in Figure 6.14(a). Nodes represent intersections of main roads included in the database. For example, node 2 represents the intersection of Route 23 and Interstate 270. The goal is to find a path from node 1 to node 29 with a minimum cost specified in (6.10). For each edge, we define a pair of metrics $(t_{i,j}, r_{i,j})$, which are the travelling time and dynamic RRI, respectively. The expected travel time is measured using Google Maps at 04:30 PM, 10/16/2014 (Thursday) EST. The RRIs are generated using the model we developed in Section 6.2 and dynamic factors in Section 6.4. To accomplish this, we link map road segments to the corresponding homogenous road segments in the HSIS database according to the county, road number and milepost. We will next illustrate the optimal route under the following scenarios to identify the specific road segment characteristics (but not RRIs which are computed from the model). We assume that the driver's driving style and the vehicle conditions are both nominal.

6.6.1 Time optimal route

When $\alpha = 0$, the user does not care about road risk and desires a time-optimal route. As expected, the Cplex results match the Google Maps results shown in Figure 6.13(a). The optimal time route in terms of Figure 6.14(a) nodes is 1-2-3-9-14-18-24-27-29. The expected traveling time is 42 minutes and the total risk index is 161.89. The final cost is $J = 42 + 161.89 \times 0 = 42$.

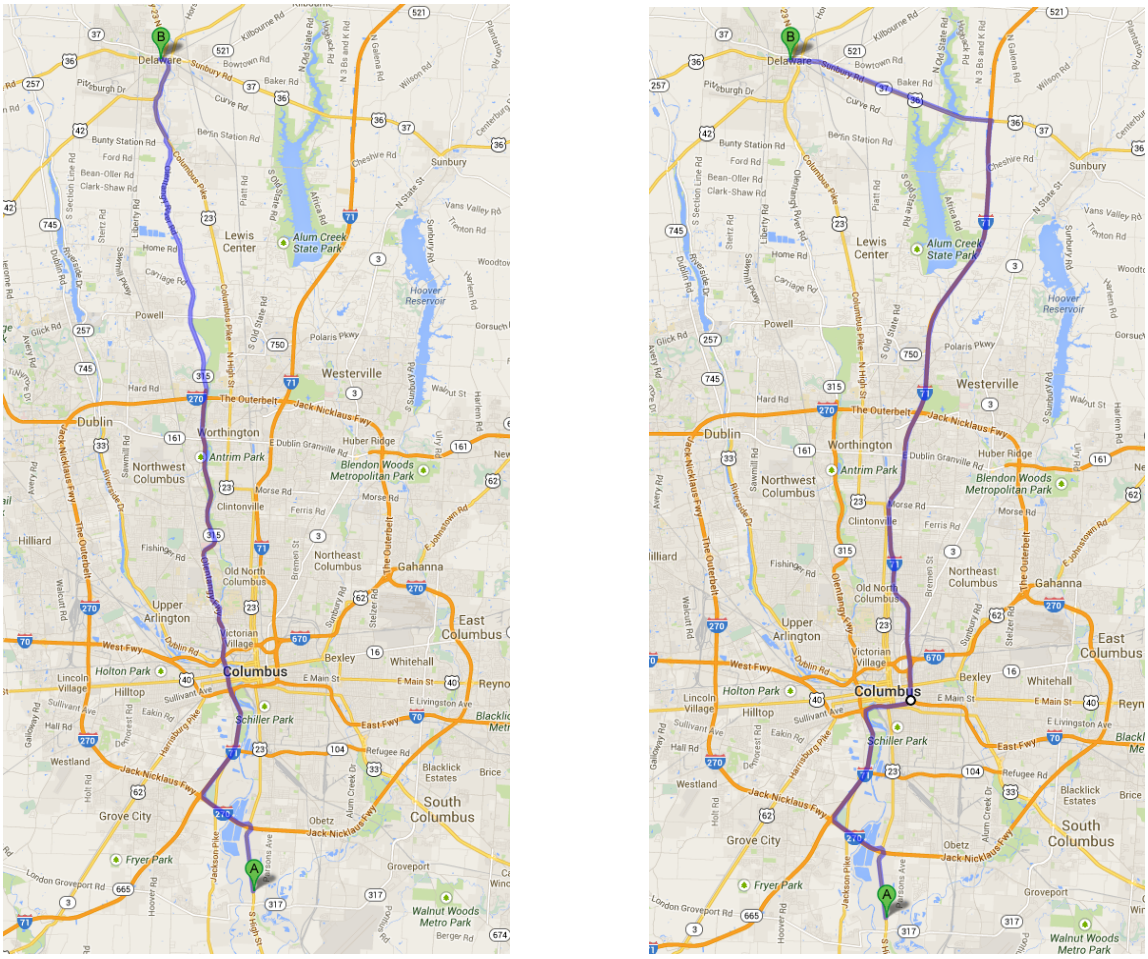
6.6.2 $\alpha = 0.2$

When $\alpha = 0.2$, the optimal route is 1-2-3-9-14-15-20-26-28-29, as shown in Figure 6.13(b). The expected travelling time is 44 minutes and the total risk index is 103.57. The final cost $J = 44 + 0.2 \times 103.57 = 64.71$. This second route has 36% less risk than the first route but requires 2 additional minutes of travel time.

The above example pair shows that a route can indeed change if safety is taken into account. We now show that real-time factors such as weather and time of day can also lead to different optimal routes.

6.6.3 Route changes with time of day, day of the week

Suppose the time is now 1:00 am on Saturday, as discussed in Section 6.4.2, all the RRI will be updated by applying a correction factor of $\frac{585}{304} = 1.92$. Again assume weighting factor $\alpha = 0.2$. As a result of the day and time, RRI is greatly increased for



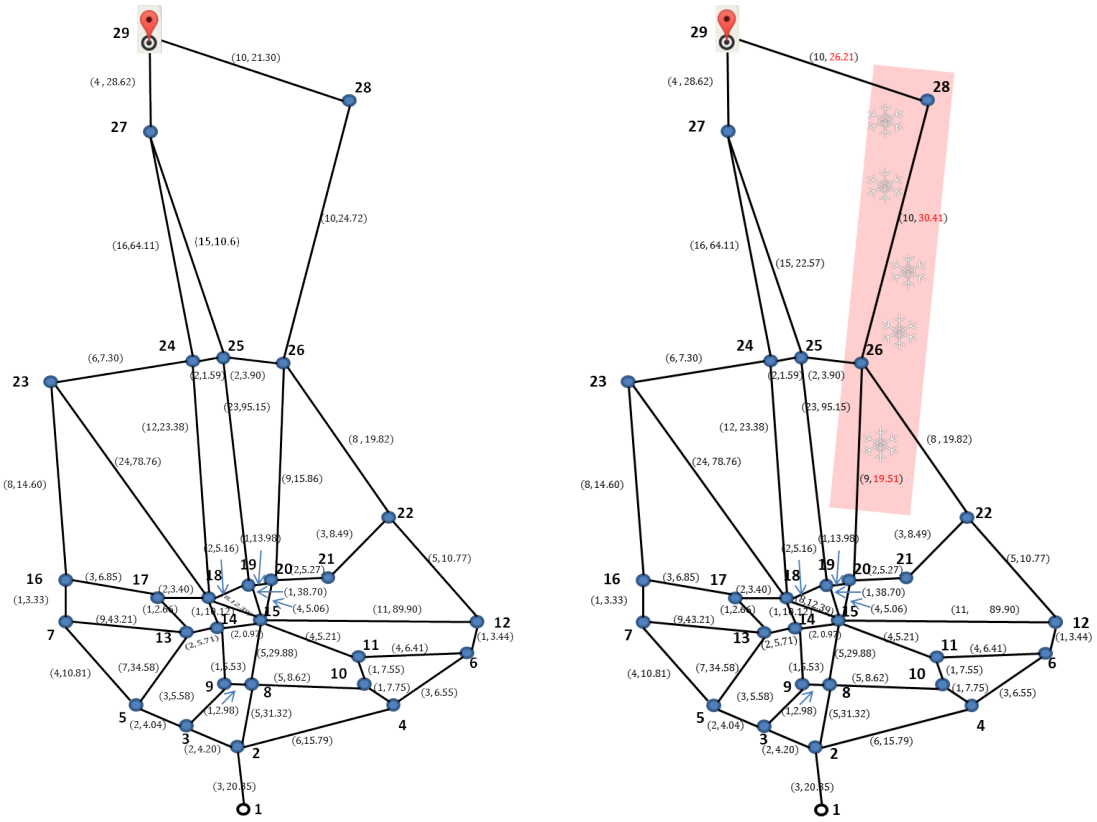
(a) $\alpha = 0$ (b) $\alpha = 0.2$

Figure 6.13: Optimal routes with different α's

each road segment. These changes result in a safety-optimal route as 1-2-3-9-14-15-20-26-25-27-29 which is visualized in Figure 6.15(a). The expected travelling time is 45 minutes and the total risk index is 204.42. The final cost $J = 45 + 0.2 \times 204.42 = 85.88$. We note that this route is also the "safest" route when independent of time ($\alpha = \infty$).

6.6.4 Route changes with weather conditions

We now change our travel time back to 04:30 pm on Thursday and suppose it is snowing in east Columbus as seen in the pink area of Figure 6.14(b). As discussed in Section 6.2, the RRIs in the affected roads are adjusted using a correction factor $\beta = \frac{452.68}{396.77}$, where 452.68 and 396.77 are the average number of daily accidents in snowy days and overall average number of accidents, respectively. A snow cover can significantly impact travel time as well as risk. While attributes of a road such as



(a) Abstracted road network map (b) Roads with local snow to the east

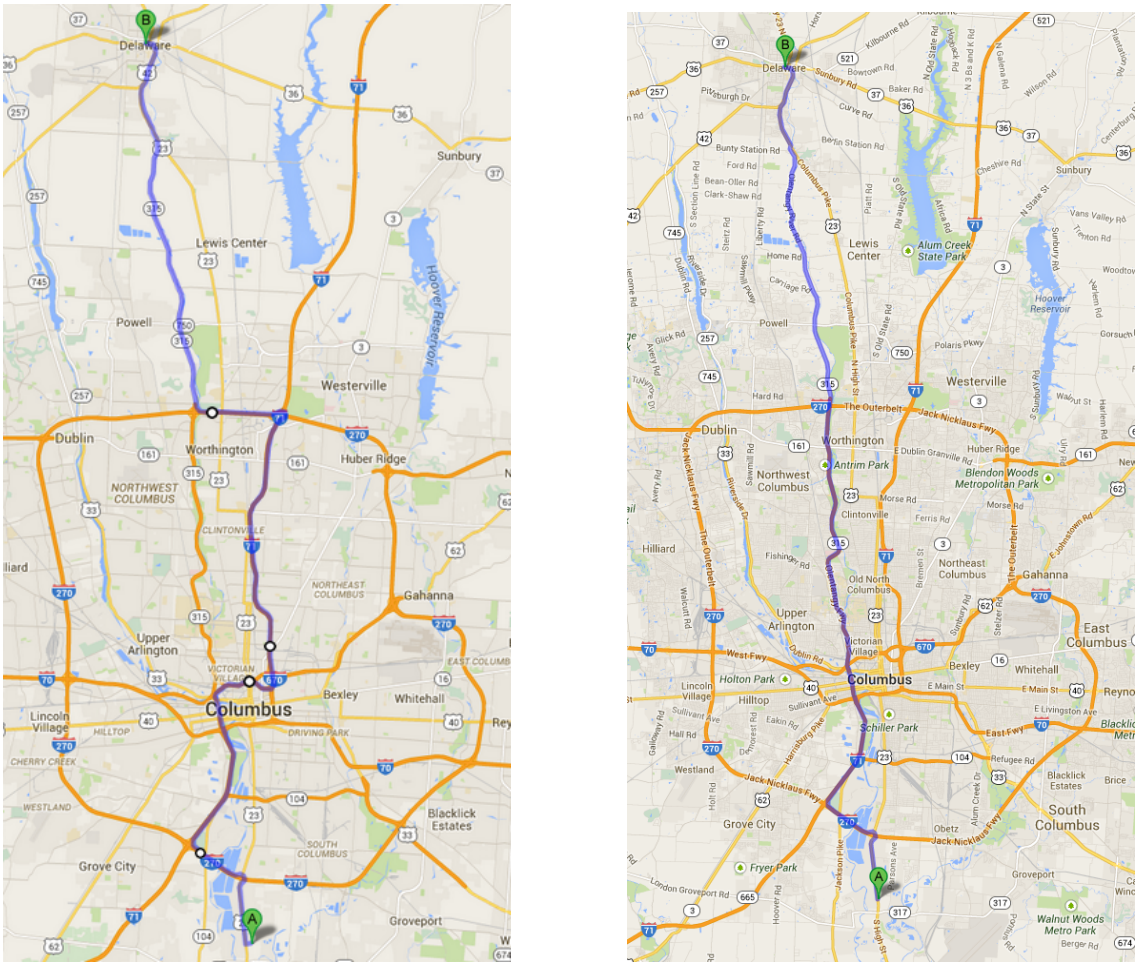
Figure 6.14: Abstracted road maps

grade and curvature would certainly factor into travel speed reduction, in this work, we adopt a representative 20% travel time increase based on a report showing data indicating a 5-19% speed reduction range in snow [104]. With the updated time, RRIs (in red in Figure 6.14(b)) and weight $\alpha = 0.2$, the optimal route generated with Cplex changes to 1-2-3-9-14-18-24-25-27-29. The expected traveling time is 42 minutes and the total risk index is 161.89. The final cost is $J = 42 + 161.89 \times 0.2 = 74.38$.

Remark 6.2. In practice, several modes (*e.g.*, time-optimal, balanced, safe) can be predefined by assigning specific α's. User feedback about the planned routes can also be exploited to obtain the optimal weighting factors with an adaptive scheme, which will be considered in our future work.

6.7 Summary and discussion

In this chapter a safety-based route planner was proposed. This planner optimized over both travel time and road risk metrics. Primary contributions of the work were



(a) Optimal route for travel at 1 am on Saturday

(b) Optimal routes with local snow.

Figure 6.15: Route changes with real-time RRI adjustments

in processing of accident data to an RRI metric in route optimization that balances time and safety metrics. Advantages of its V2C2V implementation included access to extensive computational and regularly-updated database resources not available onboard. We demonstrated that a hybrid neural network model can be developed to model road risk index based on available accident data. We showed that this model outperforms several model alternatives in terms of root mean square error. A sensitivity analysis of this model was performed showing reasonable trends.

The route planning problem was reduced to a Mixed Integer Programming problem and solved with Cplex. Real world case studies were considered that demonstrate changes in the route when safety is included in the optimization. We also illustrated how real-time information such as weather, time of day, and day of the week can be factored into route planning. Future work will include RRI database management,

tradeoff analysis of how long the route buffer is downloaded, and a demonstration with a V2C2V-equipped vehicle.

CHAPTER 7

Cloud-aided comfort-based route planning

7.1 Introduction

People spend significant time in their vehicles commuting every day. Based on a report by the AAA Foundation for Traffic Safety [105], Americans drove an average of 29.2 miles with an average duration of 46 minutes each day. Ride comfort is therefore increasingly demanded and has become one of the differentiating features that car manufacturers are pursuing [106].

Factors that affect ride comfort are classified by Corbridge [107] into three categories: dynamic factors including shock, vibration and acceleration, ambient factors such as temperature, air quality and acoustic noise, and spatial factors associated with ergonomics. While all three factors are important, in this chapter, we focus on dynamic factors associated with vibration and accelerations. Numerous advanced control and system design technologies are being pursued to improve ride comfort from the perspective of vibration and accelerations. For example, semi-active and active suspension systems have been intensively studied and implemented to isolate vehicle occupants from road vibrations [37, 35, 108, 109]. Noise Vibration and Harshness (NVH) performance analysis has been exploited in the design of vehicle trim and transmission systems [106, 110].

Due to variations in roughness, geometries and anomaly distribution, some roads are more comfortable than others. In this chapter, we propose to improve ride comfort through a comfort-based route planning framework that identifies routes with good ride comfort as well as low travel time. While modern vehicle navigation systems are able to generate optimal routes in terms of travel time, distance or fuel economy, in this chapter, we augment such cost terms with comfort-based metrics.

To implement comfort-based route planning, metrics are needed to reflect relative comfort level of road segments. While subjective assessment may be a natural

and direct way to indicate the ride quality, its use for route planning is impeded by variability between subjects and the difficulty of subjective data collection. As an alternative, studies have focused on analyzing the stimuli that affect ride comfort [107, 111, 112, 113, 114]. Vibration-related indices have been developed. International Roughness Index (IRI) [115] has been widely used and generally accepted as a standard metric to describe the roughness of road segments. However, its measurement relies on sophisticated road profilers which are expensive to acquire and operate. Power Spectral Density (PSD) and Vibration Does Values (VDV) are also widely used for vibration comfort analysis under British Standard 6841 (BS 6841) [111] or International Standard 2631 (ISO 2631) [112]. BS 6841 recommends four acceleration measurements, three on the seat (fore-aft, lateral, vertical) and one at the backrest (fore-aft). ISO 2631 recommends measurement of the three-dimensional acceleration on the seat pan. While these metrics reveal good correlation with ride comfort, their use for route planning is limited due to sensor requirements.

Recently, inexpensive and easy-to-implement estimation methods have been explored. In [45], a standalone vehicle accelerometer is used to measure vehicle body acceleration which is then mapped to road power spectral density with a transfer function generated from a half-car model. Also, in [44], accelerometers in a smart phone are used to correlate vertical acceleration with road roughness. While these approaches are easy to implement, they are only able to classify road roughness level and may not be sufficient for comfort assessment.

In this chapter, we formulate three objective comfort-based metrics by exploiting the simultaneous road profile estimation and anomaly detection framework developed in Chapter 3. We first define a Roughness Index (RI) based on estimated road profile. The notion of Roughness Weighted Time (RWT) is further developed by integrating time and RI. In addition, a Road Anomaly Cost (RAC) metric is defined based on driver aversion to road anomalies such as potholes and bumps. Finally, an Intersection-Induced Cost (IIC) is developed by considering potential stop-and-go's and turns at intersections. While there are many other factors affecting riding comfort, the developed route-dependent metrics, RWT, RAC, and IIC, are considered representative factors suitable for route planning. A real-world case study, route planning from Ford Research and Innovation Center, Dearborn Michigan, to Ford Rouge Factory Tour, Dearborn Michigan, is presented to demonstrate how the "comfortable" route can differ from the fastest route.

In route planning algorithms, road networks are traditionally modeled as graphs where nodes represent main intersections and edges represent road segments [92]. A

route planner generates an optimal route that minimizes total cost. Costs are traditionally assigned to route segments, *e.g.,* time, distance, fuel *etc.* However, in the proposed comfort-based route planning framework, route segment transition costs such as turns at intersections correspond to costs associated with pairs of connected edges are considered. The presence of these edge-transition costs complicates the design of a planning algorithm. There are generally two approaches to deal with transition costs. The first approach exploits edge-based graphs in which edges in the original graph are replaced by nodes and pairs of consecutive edges modeled as edges in the new graph [116, 117]. This corresponding road network representation can model edge costs such as travel time as well as edge-transition costs such as intersection turns. However, additional nodes are introduced so computational complexity increases. Another approach employs an exception list that stores transition costs for adjacent edges [118]. The planning algorithm can then query the transition cost list during the route optimization. This approach retains the traditional network representation but a list is needed with associated query increasing computation time. In this chapter, we exploit the latter approach, a planner with a transition cost list, to perform comfort-based route planning.

The proposed comfort-based route planner is envisioned with a Vehicle-to-Cloud-to-Vehicle (V2C2V) implementation as illustrated in Figure 7.1. V2C2V-enabled vehicles are used as mobile sensors to estimate road profile and detect road anomalies as discussed in Chapter 3. These data are sent to the cloud and a data processing module extracts useful information from aggregated reports. Road profile and anomaly information is then stored in a cloud database for use in comfort-based route planning. The user specifies the destination and user preferences which reflect the user's trade-off between comfort and travel time. A comfort-based route planner then combines real-time travel information from the web, road information from a database, and road profile and anomaly data from a database to generate an optimal comfort-based route. The benefits of V2C2V implementation include road data crowd-sourcing and sharing, real-time traffic data update, and fast route optimization.

This chapter is organized as follows. Section 7.2 discusses comfort-related factors and develops objective comfort metrics for route planning. The comfort-based route planning problem is formulated and solved in Section 7.3. A real-world case study is presented in Section 7.4 while Section 7.5 provides a summary and discussion.

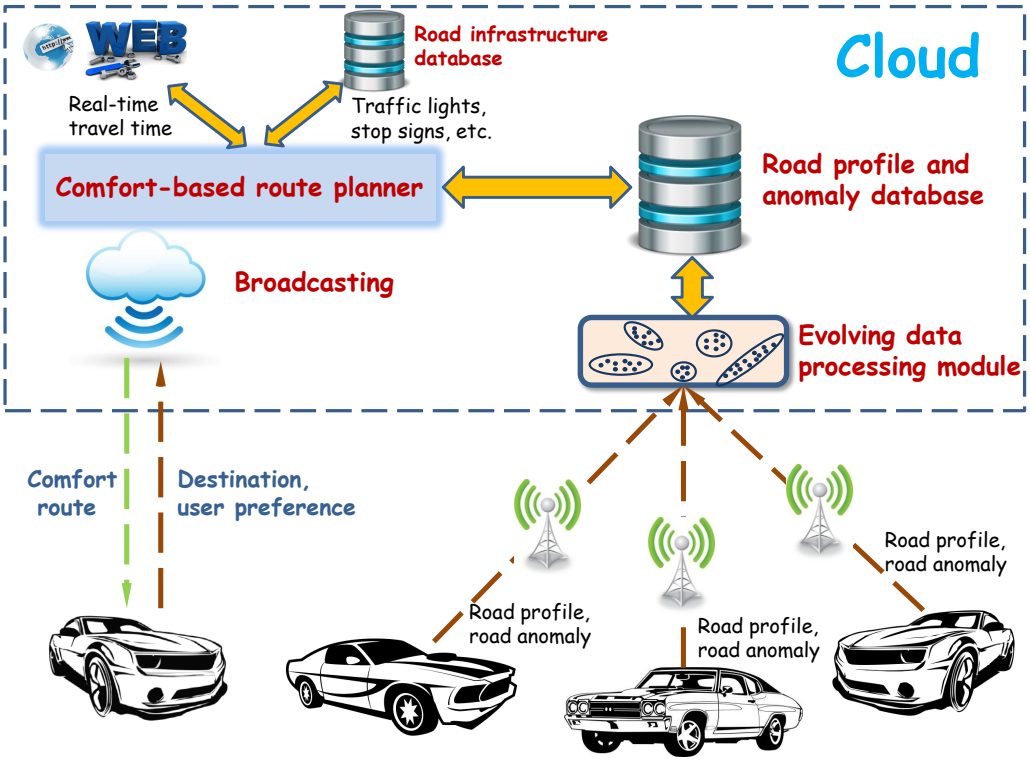


Figure 7.1: Architecture of cloud-aided comfort-based route planning

7.2 Comfort-related factors and metrics for route planning

In [107], factors affecting riding comfort are divided into three categories: vibration and acceleration, ambient, and spatial. While ambient and spatial aspects are important in vehicle air conditioning and ergonomic design, this chapter considers ride comfort from the perspective of vibration and accelerations.

7.2.1 Comfort-related vibration and acceleration factors

For the purpose of route planning, in this subsection, we consider route-dependent stimuli that influence riding comfort.

7.2.1.1 Road roughness

Rough roads can cause discomfort and driver fatigue. The International Roughness Index (IRI) [115] is a standard metric to measure road roughness. However, expensive road profilers are needed to collect IRI. Also, the road conditions evolve over time so use of a profiler is further impeded.

In Chapter 3 we developed a systematic approach to estimate road profile using commonly-available vehicle sensors. This road profile estimator computes road inputs at front left and right wheels. For a specific road segment, we use the average of the root mean square (RMS) of the estimated inputs in front left and right wheels as an indicator of roughness index (RI):

$$RI(i) = \frac{1}{2}\big(RMS(\hat{V}_{l,i}) + RMS(\hat{V}_{r,i})\big), \tag{7.1}$$

where i is road segment number; $RMS(\cdot)$ denotes the root mean square of a vector or sequence. $\hat{V}_{l,i}$ and $\hat{V}_{r,i}$ represent the sequence of roadway velocity input estimates at the left and right wheels on road segment i, respectively.

Based on the experimental results in Chapter 3, the estimated RI from (7.1) for the road surfaces illustrated in Figure 7.2(a), Figure 7.2(b) and Figure 7.2(c) are 0.62, 0.85 and 1.03, respectively.

(a) Surface 1 (b) Surface 2 (c) Surface 3

Figure 7.2: Three road surface types; roughness increases from left to right. The roughness indices (RI) are 0.62, 0.85 and 1.03, respectively.

7.2.1.2 Road anomalies

Besides rough roads, road anomalies such as potholes and bumps can induce sudden acceleration changes and worsen the driving experience. These anomalies can cause back or neck pain. In Chapter 3, we presented an algorithm to detect and label the anomalies. The anomalies can be detected with onboard sensors and accurately labeled as large pothole, small pothole (or manhole), speed bump or road joints as illustrated in Figure 7.3. For comfort-based route planning, different costs can be specified for each of the anomaly types based on the driver's evaluation of the anomalies.

Note that hitting an existing anomaly is a probabilistic event. Anomaly locations can be determined using the clustering algorithm developed in Chapter 4 with aggre-

(a) Pothole (b) Manhole (c) Speed bump (d) Road joints

Figure 7.3: Road anomaly types

gated anomaly reports. With cloud queries, probabilities of hitting these anomalies can be estimated.

7.2.1.3 Turns at intersections

Turns at intersections are undesirable for safety and efficiency reasons. The United Parcel Service (UPS) eliminated left turns for their delivery trucks which has reduced fleet-wide fuel consumption [119]. Turns at intersections can also degrade ride comfort due to induced lateral accelerations and longitudinal deceleration. Left and right turns are illustrated in Figure 7.4. We hypothesize that left turns cause more discomfort than right turns because these turns typically cross multiple oncoming traffic lanes.

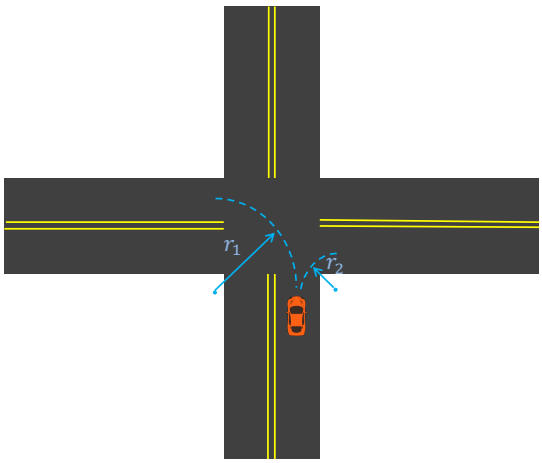


Figure 7.4: Left and right turns at intersections

7.2.1.4 Stop signs and traffic lights

Stop signs can also cause discomfort because of induced longitudinal acceleration while the vehicle decelerates to a full stop and accelerates again. Similarly, traffic

lights may require stop-and-go behavior with likelihood based on duration statistics. Note that left turn, right turn and straight travel have different statistics at each intersection.

7.2.1.5 Driving duration

Travel time determines how long the driver is exposed to the vibrations and thus plays an important role in ride comfort. Prolonged driving can cause fatigue and further reduce driving quality.

7.2.1.6 Others

Besides factors listed above, there are other factors that also affect ride comfort. For example, road curvature and grade can affect ride comfort with similar induced accelerations. Lighting and weather conditions, lane width, and truck traffic are also influencing factors. However, in this chapter, we only consider the vibration-related factors listed above as models for metrics and experimental data were available.

7.2.2 Comfort metrics for route planning

In this subsection, we integrate the aforementioned comfort factors to develop metrics for comfort-based route planning. We first introduce roughness-weighted time (RWT), which combines the impact of travel time and road roughness in a unified metric. The RWT of road segment i is defined as

$$RWT(i) = S\big(RI(i)\big)t(i), \tag{7.2}$$

where i is the road segment number. $RI(i)$ is the roughness index of road segment i defined in (7.1), $t(i)$ is the expected travel time of road segment i, and $S(\cdot)$ is the sigmoid weighting function defined as

$$S(x) = a + \frac{b-a}{1+e^{-k(x-x_0)}}, \tag{7.3}$$

where a and b are the lower and upper bounds of the weight, respectively; x_0 is the sigmoid's midpoint and k represents the stiffness of the curve. Based on roughness estimation for different road pavements illustrated in Figure 7.2, we specify an example function as illustrated in Figure 7.5 with $a = 0.8$, $b = 1.2$, $x_0 = 0.08$, and $k = 80$. Note that if $a = b = 1$, $S(x) \equiv 1$.

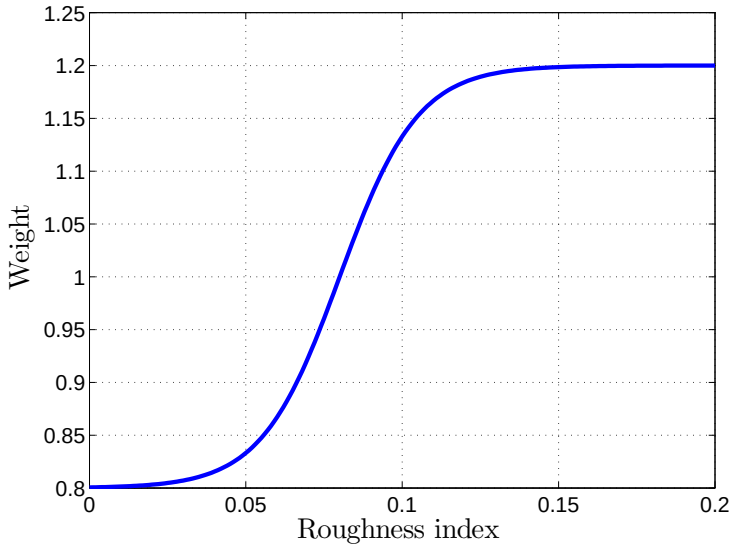


Figure 7.5: An example Sigmoid function (7.3) with $a = 0.8$, $b = 1.2$, $x_0 = 0.08$, and $k = 80$

The function indicates an ideally smooth road can be as pleasant as a twenty percent discount on travel time while travel on a very rough road is equivalent to having an additional twenty percent of travel time. The lower and upper bounds can be specified by the user based on driver sensitivity to road roughness.

We next consider the effects of road anomalies. Drivers may have different trade-offs between anomalies and travel time. A questionnaire may be used to obtain the driver's personal aversion level for each anomaly type. See Table 7.1 for the questionnaire and an example of the author's preferences. Through the questionnaire the driver rates the level of undesirable events by providing a number ranging from 0 to 10 for each event where 10 represents most undesirable. Driver preferences might be averaged over more extensive human subjective experiments in future work.

Table 7.1: Driver comfort metric questionnaire for road anomalies

Large pothole (c_{LP})	Small pothole (c_{SP})	Speed bump (c_{SB})	Road joints (c_{RJ})
8	4	8	2

Road anomaly events, *i.e.*, large pothole (LP), small pothole (SP), speed bump (SB), and road joints (RJ), can be identified using the frameworks developed in Chapter 3 and Chapter 4. The probability of hitting these anomalies can be estimated with aggregated reports. The total road anomaly cost (RAC) on road segment i is

defined as a summation:

$$RAC(i) = J_{LP}(i) + J_{SP}(i) + J_{SB}(i) + J_{RJ}(i), \tag{7.4}$$

where subscripts LP, SP, SB, and RJ, represent anomaly types. If the corresponding undesired event is not present for road segment i, then the associated cost component is set to 0. For example, if there is no speed bump on segment i, then $J_{SB}(i) = 0$. Otherwise, each cost component reflects the expected value of the number of potential encounters times driver-specified cost. For example, if road segment i has probabilities 0.5, 0.8, and 0.7 for hitting a sequence of three large potholes then the large pothole induced cost is

$$J_{LP} = (0.5 + 0.8 + 0.7) * c_{LP}, \tag{7.5}$$

where c_{LP} is the user-specified cost for large potholes in Table 7.1, *i.e.*, $c_{LP} = 8$.

As discussed, intersections are undesirable for the possible stop-and-go events as well as possible turns. We introduce an Intersection-Induced cost (IIC) metric in which we consider both intersection turns and stop-and-go's. Similar to RAC development, a questionnaire may be used to quantify the driver's aversion to these events. See Table 7.2 for the questionnaire and the author's personal preferences. This questionnaire collects driver aversion level of the undesirable events at intersections by specifying a number ranging from 0 to 10 for each event where 10 represents most undesirable. These preferences can also be more accurately obtained with human subject experiments in future work.

Table 7.2: Driver comfort metric questionnaire for intersections

Left turn (c_{LT})	Right turn (c_{RT})	Go straight (c_{GS})	Stop-and-Go (c_{SG})
8	6	4	8

As an example, a right turn at an intersection with probability of 0.3 encountering a red light has an IIC equal to $c_{RT} + c_{SG} * 0.3 = 8.4$. A left turn at an intersection with a stop sign has an IIC of $c_{LT} + c_{SG} = 16$. Note that the IIC captures the transition cost between two road segments and is associated with two edges in a graph.

Remark 7.1. The developed comfort metrics are driver-dependent. Preferences and weighting factors need to be specified to define the metrics. We note that the weights could be obtained by recursive learning with driver feedback, *e.g.*, based on driver's comfort rating after each drive, or comparisons between different routes. However, such work is beyond the scope of this thesis.

7.3 Comfort-based route planning

In Section 7.2, ride comfort metrics, RWT, RAC and IIC, are defined. We incorporate these metrics into a comfort-based route planning framework. We model the road network as a directed graph as illustrated in Figure 7.7. Intersections and road segments are modeled as nodes and edges, respectively, in this directed graph. The route planner generates an optimal route in the road network based on the specified cost functions for RWT, RAC and IIC.

7.3.1 Problem Formulation

For a directed graph $G = \{V, E\}$, $V = \{v_1, \cdots, v_n\}$, $|V| = n$, represents the list of nodes, while $E = \{e_1, \cdots, e_m\}$, $|E| = m$, is the list of edges. Each edge $e = (s, t)$ consists of a *source* node and a *target* node and the following mappings are defined,

$$\begin{aligned} source\big((s,t)\big) &\triangleq s, \\ target\big((s,t)\big) &\triangleq t. \end{aligned} \tag{7.6}$$

We define three weighting functions: $w_{RWT}(e) : E \to \mathbb{R}_0^+$ assigns each edge a RWT cost; $w_{RAC}(e) : E \to \mathbb{R}_0^+$ assigns each edge a RAC cost; and $w_{IIC}(e_1, e_2) : E \times E \to \mathbb{R}_0^+$ assigns each pair of connected edges an IIC cost.

A *path* is a list of connected edges $P = \{e_1, \cdots, e_p\}$ such that $source(e_{i+1}) = target(e_i)$ for $i = 1, \cdots, p-1$. Path weights are defined similarly as

$$\begin{aligned} w_{RWT}(P) &= \sum_{e_i \in P} w_{RWT}(e_i), \\ w_{RAC}(P) &= \sum_{e_i \in P} w_{RAC}(e_i), \\ w_{IIC}(P) &= \sum_{i=1}^{p-1} w_{IIC}(e_i, e_{i+1}). \end{aligned} \tag{7.7}$$

We also define $source(P) \triangleq souce(e_1)$ and $target(P) \triangleq target(e_p)$. Then given a start node s and destination node d in the graph, the goal is to find a path from s to d such that a predefined cost function is minimized. In this work, we define the cost function as a weighted sum of RWT, RAC and IIC costs, *i.e.*,

$$\min_{P \in \mathcal{P}} \quad w_{RWT}(P) + \alpha_1 \cdot w_{RAC}(P) + \alpha_2 \cdot w_{IIC}(P), \tag{7.8}$$

where $\mathcal{P} := \{P : source(P) = s, target(P) = d\}$, and $\alpha_1 > 0$, $\alpha_2 > 0$ are weighting factors reflecting the driver's tradeoff between RWT, RAC and IIC. For example, if $\alpha_1 = 0.1$, the driver would like to spend 0.8 minutes (roughness weighted) to avoid a large pothole based on the questionnaire response in Table 7.1.

7.3.2 Optimal route planning by Extended Dijkstra's algorithm

The conventional shortest-path problem has been extensively studied and a rich set of methods have been explored, *e.g.,* A^* algorithm [97], Dijkstra's algorithm [96], and the Genetic Algorithm [98]. However, these methods only consider costs assigned to graph edges. In our problem, the IIC models the edge-transition cost and is associated with two edges. As a result, traditional methods can not be directly applied.

As discussed above, two main approaches have been developed to handle graph search with dual-edge costs: the edge-based graph [116, 117] and table lookup. For the table lookup, the cost table data structure is over a three-node sequence. For example, let $IIC(v_1, v_2, v_3) = c$ denote the IIC cost of the edge transition $v_1 \to v_2 \to v_3$. In the graph depicted in Figure 7.7, $IIC(1, 2, 5)$ indicates the transition cost of $1{-}2{-}5$ which captures a right turn and possible stop-and-go behavior. Note that these transition costs can be stored in a $4 \times k$ matrix, where k is the number of edge transitions and each column represents the three associated nodes (v_1, v_2, v_3) and the cost value (c). In this work, we propose this approach to implement comfort-based route planning. Using the $4 \times k$ edge cost matrix, Dijkstra's algorithm can then be applied to find the optimal route. Note that additional search is required to query the IIC costs. We refer to the Dijkstra's algorithm with an additional list (or matrix) structure as an extended Dijkstra's algorithm.

7.4 Route planning case study

In this section, we present a real-world case study for the developed comfort-based route planner. We plan a route from Ford Research & Innovation Center, Michigan to Ford Rouge Factory Tour, Michigan, with the goal of finding a fast and "comfortable" route. These sites are illustrated in Figure 7.6 in which a time-optimal route is also indicated on Google Maps.

To plan the route, we first abstract the road network into a graph as shown in Figure 7.7. Nodes represent intersections of main roads. For example, node 7 represents the intersection of Michigan Avenue and Oakwood Boulevard, Dearborn Michigan. The goal is to find a path from node 1 to node 25 with minimum cost per

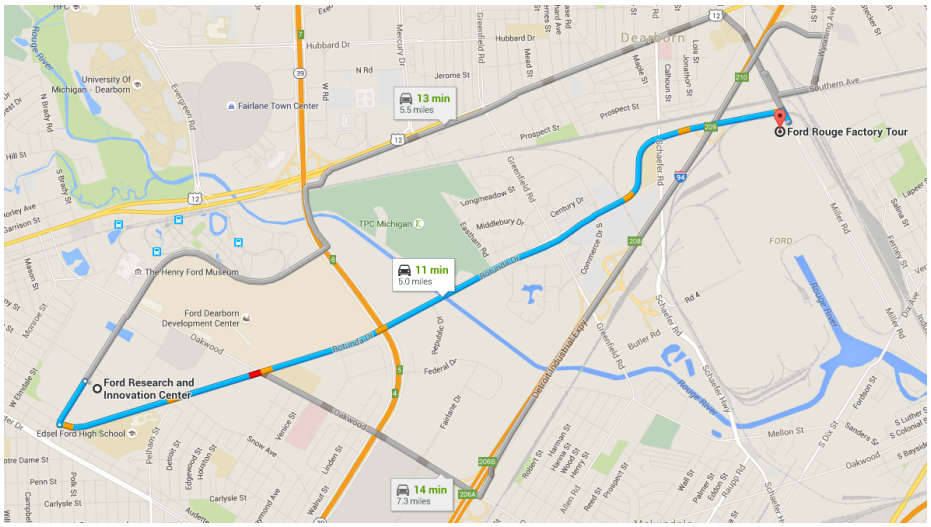


Figure 7.6: Route planning from Ford Research and Innovation Center to Ford Rouge Factory Tour. A time-optimal route is highlighted on Google Maps.

(7.8).

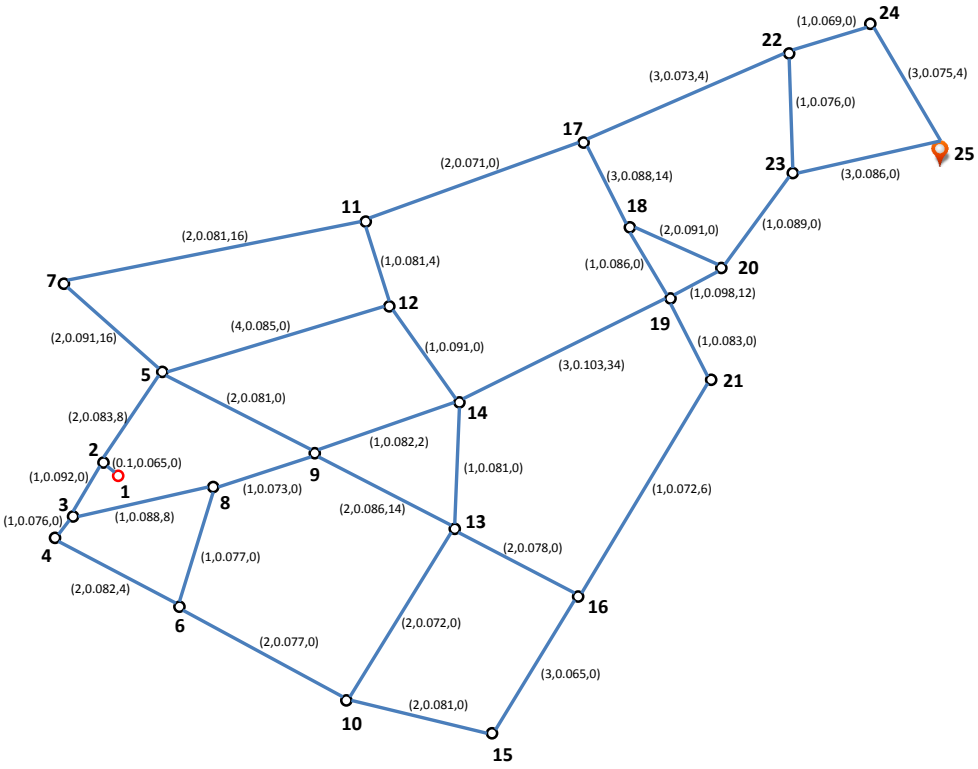


Figure 7.7: Abstracted road network map; each edge is assigned a cost triple (expected travel time, roughness index and road anomaly cost)

To obtain roughness and road anomaly information for the related route segments,

we collected data in an experimental test vehicle that implements the estimation algorithm developed in Chapter 3. The vehicle trace and detected anomalies are shown in Figure 7.8. Note that no speed bumps were encountered in the test drive. Then combining the anomalies with the driver anomaly cost specifications in Table 7.1, (7.4) and (7.5), the road anomaly cost (RAC) of each road segment was computed. Ideally data from multiple vehicles should be collected to indicate road anomalies more accurately and the probability of hitting the anomalies can be better estimated. However, in this study, test vehicle availability was limited so detected anomalies were assigned probability 1 when calculating the RAC of each road segment. Note that the above challenge can be capably handled by the implementation of a V2C2V-enabled fleet. With aggregated reports, we can detect all the anomalies and estimate the probabilities of hitting these anomalies eventually.

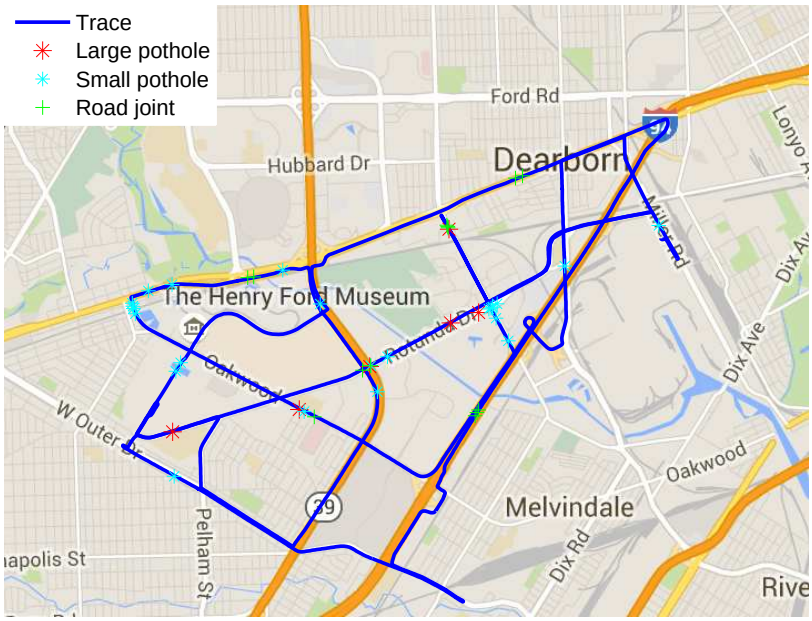


Figure 7.8: Trace of the experimental drive and detected anomalies

Road profile inputs were estimated using the simultaneous road profile estimation and anomaly detection algorithm developed in Chapter 3. The roughness index (RI) of each road segment was generated using (7.1). In this work, we use data from a single test drive as a representative study. Expected travel time was captured using Google Maps at 03:10 PM, 10/25/2015 (Sunday) EST. In the case study, we were not able to get the statistics of the traffic lights and thus will not consider the intersection-related

costs in this case study, *i.e.*, $\alpha_2 = 0$ from Equation (7.8).

As a result, each edge is associated with three metrics, *i.e.*, travel time, RI and RAC, as illustrated in Figure 7.7. Note that since the road segments we considered are all bidirectional, we approximate costs of both directions with the same metric values. To facilitate route planning, we exploit roughness weighted time (RWT) developed in (7.2) and (7.3). With the example sigmoid function in Figure 7.5, the RWT of each segment was calculated as shown in Figure 7.9. Each edge is associated with two metrics, RWT and RAC. We are now able to use Dijkstra's algorithm to find optimal routes as described below.

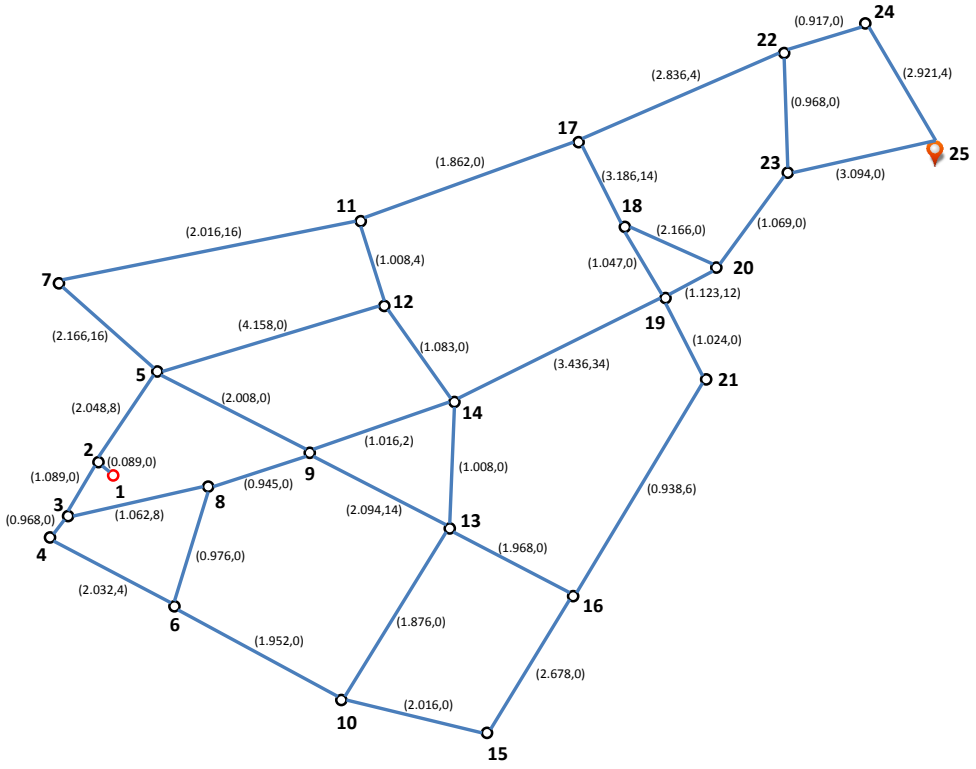


Figure 7.9: Abstracted road network map; each edge is labeled by a tuple (RWT, RAC)

7.4.1 Time optimal route

To obtain the time-optimal route without considering road roughness and road anomalies, we set $a = b = 1$ in Sigmoid function (7.3) so that the travel time is not weighted by road roughness. We then set weighting factor α_1 in (7.8) to be 0. The obtained route is 1-2-3-8-9-14-19-20-23-25 with 11.1 minutes travel time. The obtained route matches the route recommended by Google Maps (Figure 7.6).

Although it is fastest, this route contains 4 large potholes, 5 small potholes and 2 road joints which yield a total road anomaly cost of 56 ($4 \times 8 + 5 \times 4 + 2 \times 2$).

7.4.2 $\alpha_1 = 0.06$ case

We next plan routes with different α's. Let a and b set to the same values as in the example from Figure 7.5, and let $\alpha_1 = 0.06$. The obtained route is 1-2-3-8-9-14-13-16-21-19-20-23-25 as shown in Figure 7.10. The unweighted travel time is 14.1 minutes which is 3 minutes longer than the time-optimal route. However, there are only 1 large pothole, 3 small potholes and 4 road joints with an induced road anomaly cost of 28 ($1 \times 8 + 3 \times 4 + 4 \times 2$).

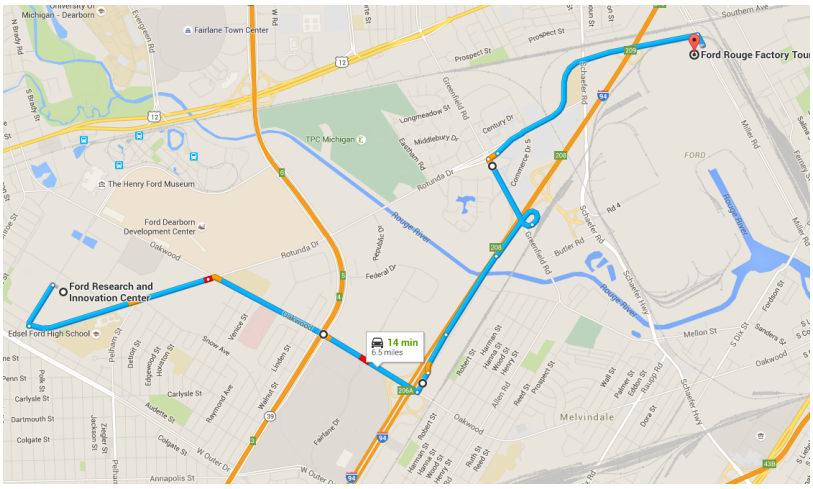


Figure 7.10: Optimal comfort route with $\alpha_1 = 0.06$

7.4.3 $\alpha_1 = 0.1$ case

By assigning $\alpha_1 = 0.1$, we obtain the route 1-2-3-8-9-14-12-11-17-22-23-25 as shown in Figure 7.11. The unweighted travel time is 15 minutes which is 4 minutes longer than the time-optimal route. However, the road anomaly cost is only 18 with 1 large pothole, 1 small pothole and 3 road joints.

7.4.4 Routes with minimum anomaly impact

To obtain the route with minimum anomaly impact, we can set α_1 to be large. By setting $\alpha_1 = 10$, we obtain route 1-2-3-4-6-10-13-16-21-19-18-20-23-25 as shown in Figure 7.12. The unweighted travel time is 20 minutes, 9 minutes longer than the

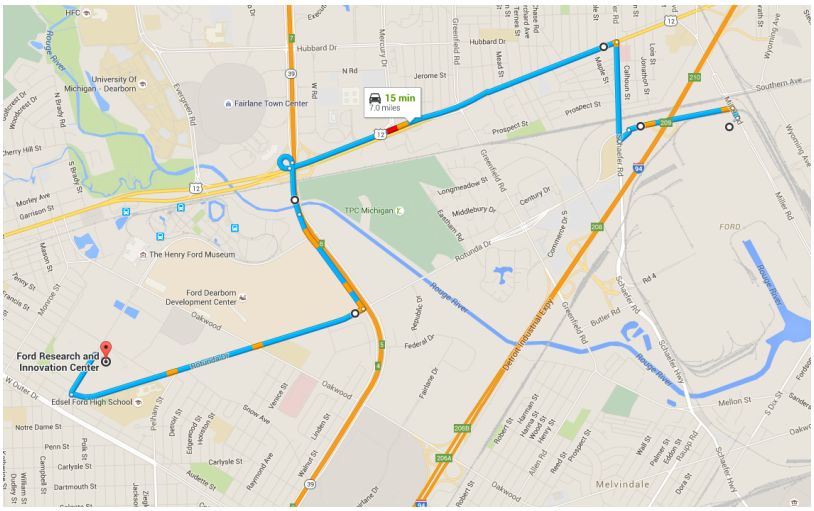


Figure 7.11: Optimal comfort route with $\alpha_1 = 0.1$

fastest route. However, this route offers minimum anomaly impact, containing 0 large potholes, 1 small pothole, and 3 road joints.

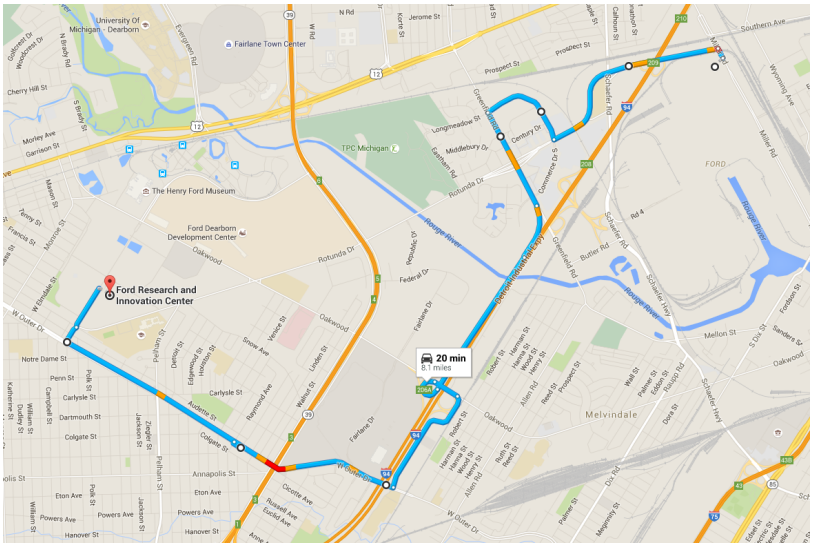


Figure 7.12: Route with minimum anomaly impact ($\alpha_1 = 10$)

7.5 Summary and discussion

In this chapter, we developed a comfort-based route planning framework that considers both time and ride comfort. Three objective metrics, roughness weighted time (RWT), road anomaly cost (RAC) and intersection-induced cost (IIC), were formulated and used in comfort-based route planning. The problem of route planning

with edge-transition cost is solved with an extended Dijkstra's algorithm. A real-world case study was presented to demonstrate how comfortable routes can differ from the time-optimal route. A simultaneous road profile estimation and anomaly detection algorithm was implemented in a test vehicle to collect road profile and anomaly data for the case study.

Future work will include a study of obtaining the driver's weights among these costs based on human subjective experiments. Additional data for intersection statistics also must be collected to include the IIC metric in the case study. We also plan to demonstrate comfort-based route planning with a V2C2V-equipped vehicle to obtain further driver feedback and develop an informative in-vehicle interface.

7.6 Acknowledgement

The author would like to thank Yuchen Bai for his help on data processing and graph editing for the case study presented in Section 7.4.

CHAPTER 8

Conclusions and future work

8.1 Summary

This dissertation has investigated several potential automotive planning and control Vehicle-to-Cloud-to-Vehicle (V2C2V) applications. In particular, a multi-phase dynamic model has been developed to characterize the response of hitting a pothole with the long-term aim of populating a crowd-sourced database of potholes and more generally road anomalies. We have also developed an algorithm to simultaneously estimate road profile and detect road anomalies, supporting this same V2C2V goal. The framework has been demonstrated in a Ford test vehicle with promising performance. This thesis also developed a clustering algorithm to process aggregated road anomaly reports to compress anomaly information. Data from the road anomaly and profile database can be used in the cloud-aided semi-active suspension control system developed in this thesis. The suspension controller exploits the road profile and anomaly data from the cloud as a preview.

A road safety metric has been developed by processing a comprehensive database from the Highway Safety Information System and a cloud-aided safety-based route planning capability has been formulated and evaluated in simulation-based case studies. We have also developed a comfort-based route planner by considering factors including road roughness, road anomaly and intersections.

8.2 Conclusions

8.2.1 Benefits of V2C2V

In this dissertation, several benefits of the V2C2V implementation have been explored. We first showed that vehicles can be used as mobile sensors to crowd-

source roadway data. In Chapter 3, we developed a framework to simultaneously estimate road profile and detect road anomalies using commonly available sensors. An evolving clustering algorithm was also developed in Chapter 4 to process anomaly reports so that the processed data can be shared with the V2C2V-equipped vehicles.

Second, we demonstrated that information on the cloud can enhance vehicle control and enable comfort and safety-based route planning. In Chapter 5, we showed that crowd-sourced road profile and road anomaly data can be used to enable cloud-based semi-active suspension control. We also showed that road safety and comfort databases can enable safety and comfort-based route planning in Chapter 6 and Chapter 7, respectively.

Third, we envisioned that cloud computing can greatly facilitate real-time computation and data access. In Chapter 5, we proposed to compute the optimal suspension mode on the cloud, which can provide a solution in real-time. We also proposed route planning optimizations in Chapter 6 and 7 on the cloud to improve computing performance and data storage capacity.

Specific developments and results are summarized below for each major contribution of this dissertation.

8.2.2 Pothole response modeling and detection

We have developed a multi-phase dynamic model to characterize the complex response of hitting a pothole. The switched system can be exploited for estimation and control purposes. Simulation comparisons between the developed model and FTire were favorable, and a pothole detection algorithm was developed from integrated application of the multi-phase model, Bayesian estimation, and an Unscented Kalman Filter.

Complex response of dynamic systems may be analyzed by decomposing it into several simplified phases based on a good understanding of the mechanism. Bayesian estimation needs a sequence of measurements and may require high-bandwidth sensors for implementation, especially for systems with fast dynamics.

8.2.3 Simultaneous road profile estimation and anomaly detection

We presented a framework for simultaneous road profile estimation and anomaly detection. The theory of JDP-based estimation was extended to the vector disturbance case, as well as the case where the jump diffusion disturbance also influences the measurement channel data. A road profile estimation and anomaly detection

algorithm was developed by combining the JDP-based estimator with a multi-input observer. The algorithm was implemented and tested in an experimental vehicle with promising results.

Road inputs to front tires were treated as disturbances in state estimation while they are treated as system inputs and estimated in the input observer. These treatments enables the algorithm which was shown to be legit based on the performance.

8.2.4 Cloud-enabled anomaly reports clustering

We developed a clustering algorithm to process aggregated road anomaly reports from cloud-enabled vehicles. The algorithm can localize isolated anomalies and compress information for stretched anomaly segments. By exploiting o-cluster and m-cluster strategies, the clustering algorithm can also reject outliers and capture the road anomaly changes over time.

In recursive computations with matrix inverse, Woodbury matrix inverse lemma can be used to resolve numerical computation issues and reduce computational complexity.

8.2.5 Cloud-aided semi-active suspension control

A novel V2C2V implementation of a semi-active suspension control system was developed. Pothole and road profile information accessed from the cloud were exploited to optimize suspension control decisions. We showed that in the simplified form, the optimal semi-active suspension mode selection problem reduces to solving three differential equations and then evaluating the predicted cost over a prescribed horizon. This application demonstrated how the crowd-sourced road data on the cloud can be used.

Many V2C2V vehicle control systems can use this similar hierarchical control strategy. Higher-level commands or specifications can be computed on the cloud and then sent to the vehicle. Real-time algorithms can be implemented onboard with received optimal operating mode or parameters. This implementation can ensure safety even communication failures.

8.2.6 Cloud-aided safety-based route planning

We proposed a safety-based route planning framework that optimizes over both travel time and road risk metrics. A road and accident database from HSIS was mined with a hybrid ANN model as shown to have superior performance over existing

models. Sensitivity analysis of the model reveals interesting trends. Dynamic factors such as time of day, day of the week and weather were first considered in a road risk index framework. The V2C2V implementation was proposed to facilitate access to extensive computational and regularly-updated database resources not available onboard. A real-world case study was performed to show how a "safe" road can differ from the fastest route.

Sensitivity analysis on developed models can indicate interesting trends and help gain confidence on the model when reasonable trends are revealed.

8.2.7 Cloud-aided comfort-based route planning

We demonstrated how crowd-sourced road profile and anomaly information can be used for a proposed comfort-based route planner. Three objective comfort-related metrics, roughness weighted time (RWT), road anomaly cost (RAC), and intersection-induced cost (IIC), were formulated and used for the comfort-based route planning. A test vehicle was employed to collect road profile and anomaly data for a real-world case study. The study demonstrated how the "comfortable" can be different from the time-optimal route.

8.3 Future work

8.3.1 Pothole response modeling and detection

In the simulation comparison between our multi-phase dynamic model and FTire, the longitudinal response of FTire has more resonance. The difference can be reduced by introducing two springs along the tires in our model. The tradeoff between model complexity and accuracy would need to be evaluated.

The developed multi-phase model was compared with the FTire simulation package for three cases. The comparison should be run multiple times (*e.g.*, 1000 times) and statistics about the error should be analyzed. Model parameters should also be tuned to improve the model performance.

Furthermore, we exploited Bayesian estimation to detect potholes, which required a sequence of measurements for each phase. Since each phase happens in a very short period of time (around 0.01s), this requirement demands high-bandwidth sensors which are not commonly available today. Other detection algorithm might be developed based on the multi-phase model with lower sensor sampling rate requirements.

8.3.2 Simultaneous road profile estimation and anomaly detection

The thresholds th_1 and th_2 in the anomaly detection algorithm (Algorithm 3.1) were chosen based on limited experimental data. In the future work, we would like to exploit machine learning techniques to determine optimal thresholds with more experimental data.

Due to sensor availability in the test vehicle, a half-car model was used in the algorithm design. Consequently, repeated patterns were revealed as expected and additional logic was required to discriminate repeated patterns from new anomalies. Based on simulation results, one more suspension deflection measurement on the right side can actually result in good estimation results even in a full-car model. The suspension sensors on the right side are typically available in most vehicles with semi-active suspension but they were not available in the vehicle used in our tests. In the future, a full-car model can be exploited augmented with the right suspension measurements. This augmentation should facilitate estimating inputs at four wheels and discriminating repeated patterns.

The algorithm was only demonstrated in one vehicle. Implementation in multiple vehicles should be performed for robustness analysis and in order to obtain more representative road information.

8.3.3 Cloud-enabled anomaly reports clustering

The clustering algorithm was demonstrated with simulations. We would like to apply the results in Chapter 3 to real-world data. The question of how to efficiently store and retrieve the data should also be considered in the future work.

Parameter selection was briefly discussed in this study. Future work should include a more comprehensive study on parameter selection based on GPS characteristics and traffic density information. Real-world data should be collected and machine learning techniques can be applied to find optimal parameters.

8.3.4 Cloud-aided semi-active suspension control

A simplified semi-active suspension control problem was considered by choosing the best mode for a certain time horizon. Future research is needed to extend these results to handle semi-active suspension state and control constraints.

Pothole locations were assumed to be exactly known and stored in a cloud database. An alternative way to model the potholes is by a jump (or Poisson) process. This model enables the suspension mode selection problem to be analyzed with similar

Jump-Diffusion Process (JDP) developments in Chapter 3. Statistics of the processes can be stored in the cloud instead of the detailed spatial profile.

Also, further research is needed to understand how the road profile information should be optimally stored, updated, and retrieved. The selection of the optimization horizon should also be considered in future work.

8.3.5 Cloud-aided safety-based route planning

We would like to find the road characteristics for Michigan and build a road risk database for Michigan road segments. We are also interested in tradeoff analysis of how long the route buffer is downloaded. The planner should be evaluated in human subjective experiments with a V2C2V-equipped vehicle to fully quantify its benefits.

Traffic density was considered in the road risk metric. Suppose drivers follow the safe routes indicated by the planner, the traffic density of related roads may also change. This dynamic evolution of traffic density due to the safety-based route planner should also be analyzed.

8.3.6 Cloud-aided comfort-based route planning

Road roughness, road anomaly, intersection, and time were considered in our comfort-based route planning problem formulation. We would like to include factors such as grade and curvature to develop a more comprehensive comfort metric. Future work should also include a demonstration with a fleet of V2C2V-equipped vehicles.

The weighting factors in the cost function are specified by the user. Future research should consider applying learning techniques to obtain the optimal weights based on driver feedback. For example, the planner may guide drivers through different routes and adjust the weights by route comparisons based on driver feedback.

Safety and comfort metrics have both been developed. We would like to extend our route planning framework to a multi-objective route planning with time, comfort and safety metrics. Road and vehicle data from University of Michigan Mobility Transformation Center (MTC) [120] and University of Michigan Transportation Research Institute can also be further analyzed. Those data may enable additional planning functionalities such as workload-based route planning.

APPENDICES

APPENDIX A

Infinitesimal generator for systems driven by a jump diffusion process

Lemma A.1 ([121]). *Consider a stochastic system in the following form,*

$$dx(t) = [A(t)x(t) + f(t)]dt + D(t,x)d\zeta + G(t,x)d\eta, \tag{A.1}$$

where A, D, G are known (matrix) functions of appropriate dimensions, f is a known function, ζ is a standard vector Wiener process and η is a vector jump process. The processes η and ζ are assumed to be independent of each other. The infinitesimal generator of a given sufficiently smooth function $V(t,x)$, denoted by $LV(t,x)$, can be shown to have the following form,

$$\begin{aligned} LV(t,x) = \frac{\partial V}{\partial t} &+ [A(t)x(t) + f(t)]^T V_x'(t,x) \\ &+ \frac{1}{2} trace(D(t,x)^T V_{xx}'' D(t,x)) \\ &+ \int_{\mathbb{R}^m} \Big[V(t, x + G(t,x)s) - V(t,x)\Big]\lambda\phi(s)ds, \end{aligned} \tag{A.2}$$

where m is the dimension of η, V_x' and V_{xx}'' denote the first and second partial derivatives of V with respect to x; λ is the rate parameter of the jumps; and $\phi(s)$ is the joint jump size probability density function, assuming hereafter that the jump size is a continuous random vector.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] "NHTSA Aims To Speed Implementation Of Vehicle-To-Vehicle Communication," http://www.motorauthority.com/news/1098340_nhtsa-aims-to-speed-implementation-of-vehicle-to-vehicle-communication, Accessed: 2015-12-13.

[2] Changjoo, K., Yong-seuk, P., and Sunhee, S., "Spatial and Temporal Analysis of Urban Traffic Volume," *2008 ESRI International User Conference*, August 2008, pp. 1–23.

[3] Mell, P. M. and Grance, T., "SP 800-145. The NIST Definition of Cloud Computing," Tech. rep., Gaithersburg, MD, United States, 2011.

[4] Filev, D., Lu, J., and Hrovat, D., "Future Mobility: Integrating Vehicle Control with Cloud Computing," *ASME Dynamic System and Control Magazine*, Vol. 1, 2013, pp. 18–24.

[5] Fernando, N., Loke, S. W., and Rahayu, W., "Mobile cloud computing: A survey," *Future Generation Computer Systems*, Vol. 29, No. 1, 2013, pp. 84 – 106. doi:http://dx.doi.org/10.1016/j.future.2012.05.023, Including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures.

[6] Zhang, S., Zhang, S., Chen, X., and Huo, X., "Cloud Computing Research and Development Trend," *Future Networks, 2010. ICFN '10. Second International Conference on*, Jan 2010, pp. 93–97. doi:10.1109/ICFN.2010.58.

[7] "Ericsson Connected Vehicle Cloud," http://www.ericsson.com/us/ourportfolio/transport/connected-vehicle-cloud, Accessed: 2015-05-15.

[8] Mangharam, R., "The Car and the Cloud: Automotive Architectures for 2020," *The Bridge*, Vol. 42, No. 4, 2012.

[9] Shrinath, A. and Emadi, A., "Electronic control units for automotive electrical power systems: communication and networks," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, Vol. 218, No. 11, 2004, pp. 1217–1230.

[10] Mao, R. and Mao, G., "Road traffic density estimation in vehicular networks," *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*, April 2013, pp. 4653–4658. doi:10.1109/WCNC.2013.6555328.

[11] Liu, C.-S. and Peng, H., "Road friction coefficient estimation for vehicle path prediction," *Vehicle system dynamics*, Vol. 25, No. S1, 1996, pp. 413–425.

[12] Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., and Balakrishnan, H., "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring," *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, ACM, New York, NY, USA, 2008, pp. 29–39. doi:10.1145/1378600.1378605.

[13] "Progressive MyRate Program," https://www.progressive.com/newsroom/article/2008/june/myrate-launch/, Accessed: 2015-05-15.

[14] "Statefarm Drive&Save with In-Drive," https://www.statefarm.com/insurance/auto/discounts/drive-safe-save/indrive, Accessed: 2015-05-15.

[15] "MyFord Mobile: Communicate with Your Ford Focus Electric," http://www.at.ford.com/SiteCollectionImages/2013_NA/July/MyFord_Mobile.pdf, 2012, Accessed: 2015-05-15.

[16] Ozatay, E., Onori, S., Wollaeger, J., Ozguner, U., Rizzoni, G., Filev, D., Michelini, J., and Di Cairano, S., "Cloud-Based Velocity Profile Optimization for Everyday Driving: A Dynamic-Programming-Based Solution," *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 15, No. 6, Dec 2014, pp. 2491–2505. doi:10.1109/TITS.2014.2319812.

[17] Li, Z., Chen, C., and Wang, K., "Cloud Computing for Agent-Based Urban Transportation Systems," *Intelligent Systems, IEEE*, Vol. 26, No. 1, Jan 2011, pp. 73–79. doi:10.1109/MIS.2011.10.

[18] Kolmanovsky, I., McDonough, K., and Gusikhin, O., "Estimation of fuel flow for telematics-enabled adaptive fuel and time efficient vehicle routing," *ITS Telecommunications (ITST), 2011 11th International Conference on*, Aug 2011, pp. 139–144. doi:10.1109/ITST.2011.6060041.

[19] Gusikhin, O. Y., MacNeille, P., and Cohn, A., "Vehicle Routing to Minimize Mixed-Fleet Fuel Consumption and Environmental Impact." *ICINCO*, Vol. 1, 2010, pp. 285–291.

[20] Szwabowski, S. J., MacNeille, P., Kolmanovsky, I. V., and Filev, D., "In-Vehicle Ambient Condition Sensing Based on Wireless Internet Access," Tech. rep., SAE Technical Paper, 2010.

[21] Li, Z., Kolmanovsky, I., Atkins, E., Lu, J., and Filev, D., "H_∞ Filtering for Cloud-Aided Semi-active Suspension with Delayed Road Information," *IFAC-PapersOnLine*, Vol. 48, No. 12, 2015, pp. 275 – 280. doi:http://dx.doi.org/10.1016/j.ifacol.2015.09.390, 12th IFAC Workshop on Time Delay Systems TDS 2015Ann Arbor, Michigan, USA, 28-30 June 2015.

[22] Li, Z., Kolmanovsky, I., Atkins, E., Lu, J., and Filev, D., "Robust H_∞ Control for a Class of Networked Uncertain Systems with Multiple Channels Subject to Markovian Switching," IEEE Conference on Decision and Control, to appear.

[23] "V2V: What are vehilce-to-vehicle communications and how do they work," http://www.extremetech.com/extreme/176093-v2v-what-are-vehicle-to-vehicle-communications-and-how-does-it-work, Accessed: 2015-12-13.

[24] Loos, S. M., Witmer, D., Steenkiste, P., and Platzer, A., "Efficiency analysis of formally verified adaptive cruise controllers," *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, IEEE, 2013, pp. 1565–1570.

[25] Li, Z., Kolmanovsky, I., Atkins, E., Lu, J., and Filev, D., "Road anomaly estimation: Model based pothole detection," *American Control Conference (ACC), 2015*, July 2015, pp. 1315–1320. doi:10.1109/ACC.2015.7170915.

[26] Li, Z., Kalabić, U. V., Kolmanovsky, I. V., Atkins, E. M., Lu, J., and Filev, D. P., "Simultaneous Road Profile Estimation and Anomaly Detection with an Input Observer and a Jump Diffusion Process Estimator," https://mfile.umich.edu/?path=/afs/umich.edu/user/z/h/zhaojli/Public/html/Pothole.pdf, [Online] Submitted to the IEEE 2016 American Control Conference.

[27] Li, Z., Filev, D., Kolmanovsky, I., Atkins, E., and Lu, J., "A New Clustering Algorithm for Processing GPS-based Road Anomaly Reports with a Mahalanobis Distance," Intelligent Transportation System IEEE Trans. on, in preparation.

[28] Li, Z., Kolmanovsky, I., Atkins, E., Lu, J., Filev, D., and Michelini, J., "Cloud Aided Semi-Active Suspension Control," *2014 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems*, December 2014, pp. 76–83.

[29] Li, Z., Kolmanvsky, I., Atkins, E., Lu, J., Filev, D., and Michelini, J., "Cloud Aided Safety-based Route Planning," *2014 IEEE International Conference on Systems, Man, and Cybernetics*, October 2014, pp. 2525–2530.

[30] Li, Z., Kolmanovsky, I., Atkins, E., Lu, J., Filev, D., and Michelini, J., "Road Risk Modeling and Cloud-Aided Safety-Based Route Planning," *Cybernetics, IEEE Transactions on*, Vol. PP, No. 99, 2015, pp. 1–11. doi:10.1109/TCYB.2015.2478698.

[31] Li, Z., Kolmanovsky, I., Atkins, E., Lu, J., and Filev, D., "Road disturbance estimation and cloud-aided comfort-based route planning," Control System Technology IEEE Trans. on, in preparation.

[32] Mednis, A., Strazdins, G., Zviedris, R., Kanonirs, G., and Selavo, L., "Real time pothole detection using Android smartphones with accelerometers," *Distributed*

Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on, June 2011, pp. 1–6. doi:10.1109/DCOSS.2011.5982206.

[33] "Pothole Alert Research," http://www.landrover.com/experiences/news/pothole-detection.html, Accessed: 2015-11-13.

[34] Yuan, K. Y., Yuan, W., Ju, J. W., Yang, J. M., Kao, W., and Carlson, L., "Numerical simulations of mechanical properties of innovative pothole patching materials featuring high toughness, low viscosity nano-molecular resins," *Proc. SPIE*, Vol. 8347, 2012, pp. 83471B–83471B–7. doi:10.1117/12.915401.

[35] Hrovat, D., "Survey of Advanced Suspension Developments and Related Optimal Control Applications1,2," *Automatica*, Vol. 33, No. 10, 1997, pp. 1781 – 1817. doi:http://dx.doi.org/10.1016/S0005-1098(97)00101-5.

[36] Giorgetti, N., Bemporad, A., Tseng, H., and Hrovat, D., "Hybrid Model Predictive Control Application Towards Optimal Semi-Active Suspension," *Industrial Electronics, 2005. ISIE 2005. Proceedings of the IEEE International Symposium on*, Vol. 1, June 2005, pp. 391–398. doi:10.1109/ISIE.2005.1528942.

[37] Guglielmino, E., Sireteanu, T., Stammers, C. W., Gheorghe, G., and Giuclea, M., "Semi-active Control Algorithms," *Semi-active Suspension Control*, Springer London, 2008, pp. 65–97. doi:10.1007/978-1-84800-231-9_4.

[38] Cosin, "FTire tire software," http://www.cosin.eu/, 2014, [Online; accessed 02-Sept-2014].

[39] Julier, S. J. and Uhlmann, J. K., "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, March 2004, pp. 401–422.

[40] Ribeiro, M. I., "Kalman and extended kalman filters: Concept, derivation and properties," 2004.

[41] Wei, J. and Dolan, J. M., "A robust autonomous freeway driving algorithm," *Intelligent Vehicles Symposium, 2009 IEEE*, IEEE, 2009, pp. 1015–1020.

[42] Ferguson, D., Baker, C., Likhachev, M., and Dolan, J., "A reasoning framework for autonomous urban driving," *Intelligent Vehicles Symposium, 2008 IEEE*, IEEE, 2008, pp. 775–780.

[43] Sayers, M. W., Gillespie, T. D., and Paterson, W. D., "Guidelines for conducting and calibrating road roughness measurements," Tech. Rep. World Bank Technical Paper 46, 1986.

[44] Douangphachanh, V. and Oneyama, H., "Estimation of road roughness condition from smartphones under realistic settings," *ITS Telecommunications (ITST), 2013 13th International Conference on*, Nov 2013, pp. 433–439. doi:10.1109/ITST.2013.6685585.

[45] González, A., O'brien, E. J., Li, Y.-Y., and Cashell, K., "The use of vehicle acceleration measurements to estimate road roughness," *Vehicle System Dynamics*, Vol. 46, No. 6, 2008, pp. 483–499.

[46] Kalabić, U., Kolmanovsky, I., and Buckland, J., "Multi-input observer for estimation of compressor flow," *ASME 2013 Dynamic Systems and Control Conference*, American Society of Mechanical Engineers, 2013, pp. V001T04A002–V001T04A002.

[47] Habib, M., Quimby, P., Chang, S., Jackson, K., and Cummings, M., "Wind gust alerting for supervisory control of a Micro Aerial Vehicle," *Aerospace Conference, 2011 IEEE*, March 2011, pp. 1–7. doi:10.1109/AERO.2011.5747536.

[48] Kolmanovsky, I. and Maizenberg, T., "Stochastic Stability, Estimation and Control in Systems Driven by Jump-Diffusion Disturbances and Their Automotive Applications," *Decision and Control, 2006 45th IEEE Conference on*, Dec 2006, pp. 4194–4199. doi:10.1109/CDC.2006.377122.

[49] Lloyd, S. P., "Least squares quantization in PCM," *Information Theory, IEEE Transactions on*, Vol. 28, No. 2, 1982, pp. 129–137.

[50] Aggarwal, C. C., Han, J., Wang, J., and Yu, P. S., "A framework for clustering evolving data streams," *Proceedings of the 29th international conference on Very large data bases-Volume 29*, VLDB Endowment, 2003, pp. 81–92.

[51] Pappas, T. and Jayant, N., "An Adaptive Clustering Algorithm For Image Segmentation," *Computer Vision., Second International Conference on*, Dec 1988, pp. 310–315. doi:10.1109/CCV.1988.590006.

[52] Grabmeier, J. and Rudolph, A., "Techniques of cluster algorithms in data mining," *Data Mining and Knowledge Discovery*, Vol. 6, No. 4, 2002, pp. 303–360.

[53] Greene, D., Tsymbal, A., Bolshakova, N., and Cunningham, P., "Ensemble clustering in medical diagnostics," *Computer-Based Medical Systems, 2004. CBMS 2004. Proceedings. 17th IEEE Symposium on*, June 2004, pp. 576–581. doi:10.1109/CBMS.2004.1311777.

[54] Bezdek, J. C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[55] Gustafson, D. and Kessel, W., "Fuzzy clustering with a fuzzy covariance matrix," *Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*, Jan 1978, pp. 761–766. doi:10.1109/CDC.1978.268028.

[56] Ailon, N., Jaiswal, R., and Monteleoni, C., "Streaming k-means approximation," *Advances in Neural Information Processing Systems*, 2009, pp. 10–18.

[57] Filev, D. and Georgieva, O., *An Extended Version of the Gustafson-Kessel Algorithm for Evolving Data Stream Clustering*, John Wiley & Sons, Inc., 2010, pp. 273–299. doi:10.1002/9780470569962.ch12.

[58] Woodbury, M. A., *Inverting Modified Matrices*, No. 42 in Statistical Research Group Memorandum Reports, Princeton University, Princeton, NJ, 1950.

[59] Mahalanobis, P. C., "On the generalised distance in statistics," *Proceedings National Institute of Science, India*, Vol. 2, April 1936, pp. 49–55.

[60] Gordon, T., "Non-linear optimal control of a semi-active vehicle suspension system," *Chaos, Solitons & Fractals*, Vol. 5, No. 9, 1995, pp. 1603 – 1617. doi:http://dx.doi.org/10.1016/0960-0779(94)00166-N, Some Nonlinear Oscillations Problems in Engineering Sciences.

[61] Giua, A., Savastano, A., Seatzu, C., and Usai, G., "Approximation of an Optimal Gain Switching Active Law with a Semiactive Suspension," *Proc. of the 1998 IEEE International Conference on Control Applications*, Trieste, 1998, pp. 248–252.

[62] Bangsing, N., Sularso, Bagiasna, K., and Nazaruddin, Y. Y., "An Experimental Investigation ito the Design of a Robust Semi-Active Suspension System for a Quarter-Car Model," *The 4th International Conference on Control and Automation*, Montreal, Que, Canada, 2003, pp. 971–975.

[63] Jansen, L. M. and Dyke, S. J., "Semi-Active Control Strategies for MR Dampers: A Comparative Study," *Journal of Engineering Mechanics*, Vol. 126, No. 8, 2000, pp. 795–803.

[64] de Jesus Lozoya-Santos, J., Sename, O., Dugard, L., Morales-Menendez, R., and Ramirez-Mendoza, R., "A Semi-active Control-oriented Damper Model for an Automotive Suspension," *6th IFAC Symposium Advances in Automotive Control*, 2010.

[65] Sayers, M. W. and Karamihas, S. M., "The little book of profiling," *the Regent of the University of Michigan*, Vol. 2, 1998.

[66] Chen, H. and Guo, K.-H., "Constrained H-infinity control of active suspensions: An LMI approach," *IEEE Transactions on Control Systems Technology*, Vol. 13, 2005, pp. 412–421.

[67] Shiao, Y., Lai, C., and Nguyen, Q. A., "The Analysis of a Semi-active Suspension System," *Proceedings of SICE Annual Conference*, Taipei, Taiwan, 2010, pp. 2070–2082.

[68] Fallah, M. S., Bhat, R. B., and Xie, W., "Optimized Control of Semi-active Suspension System Using H_∞ Robust Control Theory and Current Signal Estimation," *IEEE/ASME Transactions on Mechatronics*, Vol. 17, No. 4, AUGUST 2012.

[69] Boukas, E.-K. and Liu, Z.-K., *Deterministic and Stochastic Time-Delay Systems*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[70] Butcher, J. C., *Numerical methods for ordinary differential equations*, John Wiley and Sons Ltd., 2008.

[71] Li, Z., Osserah, H., Kolmanovsky, I., Atkins, E., and Lu, J., "Control Design of Semi-active Suspension Systems using a Quasi-Linear Control Approach," American Control Conference 2016, submitted.

[72] NHTSA, "2012 Motor Vehicle Crashes: Overview," Tech. rep., National Highway Traffic Safety Administration, 2012.

[73] Vahidi, A. and Eskandarian, A., "Research advances in intelligent collision avoidance and adaptive cruise control," *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 4, No. 3, Sept 2003, pp. 143–153.

[74] van Arem, B., van Driel, C., and Visser, R., "The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics," *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 7, No. 4, Dec 2006, pp. 429–436.

[75] McCall, J. and Trivedi, M., "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 7, No. 1, March 2006, pp. 20–37.

[76] Wang, J., Schroedl, S., Mezger, K., Ortloff, R., Joos, A., and Passegger, T., "Lane keeping based on location technology," *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 6, No. 3, Sept 2005, pp. 351–356.

[77] "US Road Assessment Program," http://www.usrap.us/home/, Accessed: 2014-04-12.

[78] Shaw-Pin, M. and Harry, L., "Modeling vehicle accidents and highway geometric design relationships," *Accidents Analysis and Prevention*, Vol. 25, No. 6, 1993, pp. 689–709.

[79] De Leur, P. and Sayed, T., "Development of a Road Safety Risk Index," *Journal of the Transportation Research Board (TRB)*, Vol. 1784, No. 02-2814, 2002, pp. 33–42.

[80] Wu, H. and Zhang, Z., "A Framework for Developing Road Risk Indices Using Quantile Regression Based Crash Prediction Model," *Transportation Research Board 91st Annual Meeting*, No. 12-3339, 2012.

[81] Chaturvedi, S., Titre, R., and Sondhiya, N., "Review of Handwritten Pattern Recognition of Digits and Special Characters Using Feed Forward Neural Network and Izhikevich Neural Model," *Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on*, Jan 2014, pp. 425–428. doi:10.1109/ICESC.2014.83.

[82] Slavin, D., Abou-Nasr, M., Filev, D., and Kolmanovsky, I., "Empirical modeling of vehicle fuel economy based on historical data," *Neural Networks (IJCNN), The 2013 International Joint Conference on*, Aug 2013, pp. 1–6. doi:10.1109/IJCNN.2013.6707111.

[83] "INRIX traffic," http://www.inrixtraffic.com/, Accessed: 2014-04-10.

[84] Highway Safety Information System, *Guidebook for the Ohio State Data Files*, August 2007.

[85] Koenker, R. and Hallock, K. F., "Quantile Regression," *Journal of Economic Perspectives*, Vol. 15, No. 4, 2001, pp. 143–156.

[86] Bezdek, J. C., Ehrlich, R., and Full, W., "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, Vol. 10, No. 2, 1984, pp. 191 – 203.

[87] NHTSA, "Time of Day and Demographic Perspective Of Fatal Alcohol-Impaired-Driving Crashes," Tech. rep., National Highway Traffic Safety Administration, 2011.

[88] Meseguer, J., Calafate, C., Cano, J., and Manzoni, P., "DrivingStyles: A smartphone application to assess driver behavior," *Computers and Communications (ISCC), 2013 IEEE Symposium on*, July 2013, pp. 000535–000540.

[89] Lu, J., Filev, D., Prakah-Asante, K., Tseng, F., and Kolmanovsky, I. V., "From vehicle stability control to intelligent personal minder: Real-time vehicle handling limit warning and driver style characterization," *Computational Intelligence in Vehicles and Vehicular Systems, 2009. CIVVS'09. IEEE Workshop on*, IEEE, 2009, pp. 43–50.

[90] Wang, Q., Lei, Z., and Qu, S., "Design of car remote monitoring system based on Internet," *Modelling, Identification Control (ICMIC), 2012 Proceedings of International Conference on*, June 2012, pp. 631–635.

[91] Hameed, S., Khalifa, O., Ershad, M., Zahudi, F., Sheyaa, B., and Asender, W., "Car monitoring, alerting and tracking model: Enhancement with mobility and database facilities," *Computer and Communication Engineering (ICCCE), 2010 International Conference on*, May 2010, pp. 1–5. doi:10.1109/ICCCE.2010.5556796.

[92] Schultes, D., "Route Planning in Road Networks," *Science*, Vol. 316, No. 5824, 2007, pp. 566.

[93] Wang, J., Wang, H., Zhang, W., Ip, W., and Furuta, K., "Evacuation Planning Based on the Contraflow Technique With Consideration of Evacuation Priorities and Traffic Setup Time," *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 14, No. 1, March 2013, pp. 480–485. doi:10.1109/TITS.2012.2204402.

[94] Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X., and Chen, C., "Data-Driven Intelligent Transportation Systems: A Survey," *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 12, No. 4, Dec 2011, pp. 1624–1639. doi:10.1109/TITS.2011.2158001.

[95] Wang, J., Ip, W., and Zhang, W., "An Integrated Road Construction and Resource Planning Approach to the Evacuation of Victims From Single Source to Multiple Destinations," *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 11, No. 2, June 2010, pp. 277–289. doi:10.1109/TITS.2010.2040276.

[96] Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., et al., *Introduction to algorithms*, Vol. 2, MIT press Cambridge, 2001.

[97] LaValle, S. M., *Planning Algorithms*, Cambridge University Press, New York, NY, USA, 2006.

[98] Nanayakkara, S., Srinivasan, D., Lup, L. W., German, X., Taylor, E., and Ong, S., "Genetic Algorithm based route planner for large urban street networks," *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, Sept 2007, pp. 4469–4474. doi:10.1109/CEC.2007.4425056.

[99] Ademoye, T., Davari, A., Castello, C., Fan, S., and Fan, J., "Path Planning Via CPLEX Optimization," *System Theory, 2008. SSST 2008. 40th Southeastern Symposium on*, March 2008, pp. 92–96.

[100] "Gurobi optimization," http://www.gurobi.com/, Accessed: 2014-04-10.

[101] Gusikhin, O. Y., MacNeille, P., and Cohn, A., "Vehicle Routing to Minimize Mixed-Fleet Fuel Consumption and Environmental Impact." *ICINCO (1)*, 2010, pp. 285–291.

[102] Schouwenaars, T., De Moor, B., Feron, E., and How, J., "Mixed integer programming for multi-vehicle path planning," *European control conference*, Vol. 1, Citeseer, 2001, pp. 2603–2608.

[103] Wolsey, L. A., *Mixed Integer Programming*, John Wiley & Sons, Inc., 2007.

[104] Robert, H., Emily, S., Daniel, K., Hesham, R., and Mohamadreza, F., "Empirical Studies on Traffic Flow in Inclement Weather," Tech. rep., Federal Highway Administration, 2006.

[105] Triplett, T., Santos, R., and Rosenbloom, S., "American driving survey: methodology and year one results, May 2013 - May 2014," Tech. rep., AAA Fundation for Traffic Safety, April 2005.

[106] Chen, X., Wang, D., Liu, W., and Han, Y., "Simulation to design car interior trims based on NVH performance," *Computer-Aided Industrial Design and Conceptual Design, 2008. CAID/CD 2008. 9th International Conference on*, Nov 2008, pp. 499–503. doi:10.1109/CAIDCD.2008.4730619.

[107] Corbridge, C., *Vibration in vehicles: its effect on comfort*, Ph.D. thesis, University of Southampton, 1987.

[108] Ikenaga, S., Lewis, F. L., Campos, J., and Davis, L., "Active suspension control of ground vehicle based on a full-vehicle model," *American Control Conference, 2000. Proceedings of the 2000*, Vol. 6, IEEE, 2000, pp. 4019–4024.

[109] Wang, J., Wilson, D., Xu, W., Crolla, D., et al., "Active suspension control to improve vehicl0e ride and steady-state handling," *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, IEEE, 2005, pp. 1982–1987.

[110] Li, J., Qi, L., Zha, X., Wu, D., and Ma, H., "Design of transmission NVH road testing," *Computer-Aided Industrial Design Conceptual Design, 2009. CAID CD 2009. IEEE 10th International Conference on*, Nov 2009, pp. 2301–2303. doi:10.1109/CAIDCD.2009.5375005.

[111] BS, "Measurement and evaluation of human exposure to whole-body mechanical vibration and repeated shock," BS 6841:1987, British Standards Institution, London, England, 1987.

[112] ISO, "Mechanical vibration and shock - Evaluation of human exposure to whole-body vibration," ISO 2631:1997, International Organization for Standardization, Geneva, Switzerland, 1997.

[113] da Silva, M. C. G., "Measurements of comfort in vehicles," *Measurement Science and Technology*, Vol. 13, No. 6, 2002, pp. R41–R60.

[114] Smith, C. C., McGehee, D. Y., and Healey, A. J., "The prediction of passenger riding comfort from acceleration data," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 100, No. 1, 1978, pp. 34–41.

[115] Sayers, M. W., Gillespie, T. D., and Queiroz, A., "The international road roughness experiment. Establishing correlation and a calibration standard for measurements," Tech. rep., 1986.

[116] Winter, S., "Modeling Costs of Turns in Route Planning," *GeoInformatica*, Vol. 6, No. 4, 2002, pp. 345–361. doi:10.1023/A:1020853410145.

[117] Volker, L., "Route Planning in Road Networks with Turn Costs," 2008.

[118] Geisberger, R. and Vetter, C., "Efficient Routing in Road Networks with Turn Costs," *Experimental Algorithms*, edited by P. Pardalos and S. Rebennack, Vol. 6630 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2011, pp. 100–111. doi:10.1007/978-3-642-20662-7_9.

[119] Joel, L., "Left-Hand-Turn Elimination," http://www.nytimes.com/2007/12/09/magazine/09left-handturn.html?pagewanted=all&_r=0, Accessed: 2015-11-17.

[120] "Mobility Transformation Center," http://www.mtc.umich.edu/deployments/se-michigan-connected-vehicle-deployment, Accessed: 2015-12-13.

[121] Øksendal, B. K. and Sulem, A., *Applied stochastic control of jump diffusions*, Springer, 2009.