Wireless Weigh-In-Motion: using road vibrations to estimate truck weights

Ravneet Bajwa

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2013-210
http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-210.html

December 16, 2013

Copyright © 2013, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Wireless Weigh-In-Motion: using road vibrations to estimate truck weights

by

Ravneet Singh Bajwa

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

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Pravin Varaiya, Chair
Professor Kameshwar Poolla
Professor Claudia Ostertag

Fall 2013

Wireless Weigh-In-Motion: using road vibrations to estimate truck weights

Copyright 2013
by
Ravneet Singh Bajwa

Abstract

Wireless Weigh-In-Motion: using road vibrations to estimate truck weights

by

Ravneet Singh Bajwa

Doctor of Philosophy in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Pravin Varaiya, Chair

Vehicle count and weight data plays an important role in traffic planning, weight enforcement, and pavement condition assessment. This data is primarily obtained through weigh stations and weigh-in-motion stations which are currently very expensive to install and maintain. This dissertation presents a wireless sensor-based solution that is relatively inexpensive, and uses the measured pavement vibrations to estimate weight of moving vehicles.

The proposed wireless sensor network (WSN) consists of vibration sensors that report pavement acceleration and temperature; vehicle detection sensors that report a vehicle's arrival and departure times; and an access point (AP) that controls the sensors and processes the incoming sensor data. The system can enable many new applications in infrastructure monitoring and intelligent transportation. We present energy-efficient algorithms for three such applications: automatic vehicle classification for categorizing each passing vehicle based on its axle count and inter-axle spacings; weigh-in-motion for estimating individual axle weight and total weight of trucks while they are traveling at normal speeds; and estimating pavement displacement from measured acceleration.

The wireless vibration sensor developed for this project has a high resolution ($\approx 400\ \mu g$) and is immune to traffic sounds that are generally picked up by MEMS accelerometers. The prototype system was deployed on real highways and results for vehicle classification, weigh-in-motion, and displacement estimation were compared against reference measurements. The system passed the accuracy standards for weigh-in-motion (WIM) systems and outperformed a nearby commercial WIM station, based on conventional technology. Since sensors are embedded directly in the pavement, the system can also enable real-time monitoring of pavement condition.

Contents

List of Abbreviations

ADC Analog-to-digital convertor

AP Access Point

ASCII American Standard Code for Information Interchange

AVC Automatic Vehicle Classification

CPU Central processing unit

DAQ Data Acquisition

ENOB Effective number of bits

FHWA Federal Highway Administration (US)

FWD Falling weight deflectometer

HVS Heavy Vehicle Simlulator

HVS Heavy Vehicle Simulator

IEEE Institute of Electrical and Electronics Engineers

ISM Industrial, scientific and medical

LAN Local Area Network

LET Layered Elastic Theory

LQI Link Quality Indicator

LTPP Long-Term Pavement Performance

LVDT Linear Variable Differential Transformer

MAC Medium Access Control

MAF Moving Average Filter

MEMS Microelectromechanical systems

MSPE Mean-Squared Percentage Error

NI National Instruments

PLR Packet loss rate

PS Pulse Span

PTZ Pan tilt zoom

RSD Relative Standard Deviation

RSE Relative Standard Error

RSSI Received Signal Strength Indicator

SAR Successive approximation register

SNR Signal-to-noise ratio

TDMA Time division multiple access

VDLR Vehicular data loss rate

VPLR Vehicular packet loss rate

WAN Wide Area Network

WIM Weigh-In-Motion

WSN Wireless Sensor Network

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank UC Berkeley for my formal education, both my undergraduate and graduate studies. I will forever be in debt to this great institution for both my professional and personal growth.

This work would not be possible without the guidance of my advisor, Dr. Pravin Varaiya, and I sincerely thank him for teaching me so much. I am very fortunate to have worked on such an awesome research problem with him. Pravin has been the perfect advisor and mentor for me, giving me the freedom to pursue my intellectual curiosities and being there for advice whenever necessary. His depth and breadth of knowledge has always amazed me and his practical approach to research will continue to be a source of inspiration for me. Thanks to Bharathwaj Muthuswamy (Bart) for putting us in touch when he heard about the project. Special thanks to Ram Rajagopal for being a second mentor to me, for his help with scientific publications, and for many important technical contributions throughout this project.

I would also like to thank members of my dissertation committee, Prof. Kameshwar Poolla and Prof. Claudia Ostertag, for their support and valuable input on this work. Special thanks to Prof. Kris Pister for being on my Qual. exam committee and sharing his keen insights on wireless sensor networks.

Sensys Networks, Inc. was instrumental to the success of this project. It would have been impossible to get this far without their financial, administrative, and technical support. Thanks to Robert Kavaler for his help with hardware and firmware design, Mike Hui for being the Linux wizard that he is and all his technical help, Todd Stiers for his help with access point setup, Kian Kiani for being the go-to mechanical engineer for me and helping us with so many research experiments, Dave Baca for his help with collecting Sunol data, George Liu for his advanced soldering skills and help with hardware debugging, Wenteng Ma for in-sensor implementation of the AVC algorithm and its commercial testing, Daniel Xing for his help with vehicle detection sensor and speed estimation, and Adam McKee for our competitive ping pong games and many interesting conversations about life and technology.

Special thanks to Christopher Flores for leading our R&D group at Sensys, coordinating the installations and truck experiments, and for many informative discussions on a weekly basis. Many thanks to Ben Wild for his contributions to the load estimation algorithm and for sharing his startup experience with me. Special thanks to Erdem Coleri for his expertise in pavement modeling, the temperature compensation model and Appendix A of this dissertation, and for many other technical contributions and discussions. It was a privilege to work with this group.

This journey would not be possible without the love and support of my family. Words cannot express my gratitude towards my parents who have always encouraged me to follow my passions and infused the love for education in me at a young age. I thank my sister for convincing me to come to US for college, and for her love and encouragement throughout my life. Special thanks to my uncle and aunt who gave me a second home when I came to US for college. I will always cherish my time with them. Finally, I thank my wife for putting up

with my long and odd working hours during this first year of our marriage. She has been the primary source of my strength and happiness during this time, and writing this dissertation would not have been such an amazing experience without her.

Chapter 1

Introduction

Roads are one of the most valuable assets owned by the government and contribute significantly to a nation's economy. However, construction and maintenance of road pavements is currently very expensive. According to the US Federal Highway Administration (FHWA), $46 billion[1] is spent annually on highway construction and maintenance but significant savings can be made by improving our pavement monitoring and management systems [6]. Pavement monitoring enables engineers to predict road deterioration and design an optimal repair (or rehabilitation) schedule to minimize costs and extend pavement lifetime. In

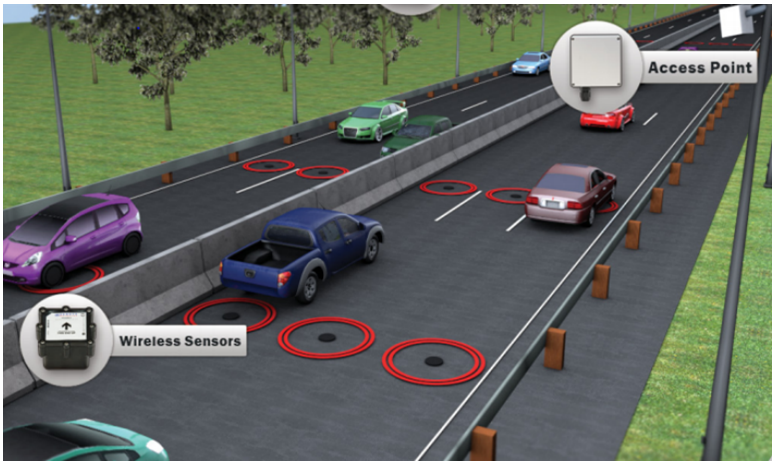


Figure 1.1: Image showing a general setup of the system.

the United States, the long-term pavement performance (LTPP) program collects important data for pavement monitoring and management such as traffic volume, truck weights, climate, structural and material properties for pavement sections, and other field testing

[1] Based on a report published by the Congressional Budget Office in 2007.

data [4]. Since heavy trucks cause significantly more pavement damage [13], accurate truck count and weight data must be collected. This information is primarily collected using Weigh-In-Motion (WIM) systems that are installed on highways and are capable of weighing trucks at high speeds. However, current WIM technologies are very expensive for widespread deployment, thus, resulting in shortage of data required for accurate assessment of pavement condition. This work explores an alternative solution that is cheap, accurate, and convenient to install in new or existing roads.

In this dissertation, we present a wireless sensor network (WSN) that can estimate the weight of moving trucks from measured road vibrations. Figure 1.1 is a general depiction of the system. Multiple wireless sensors are embedded in road to detect vehicles, and to measure vehicle speed and the corresponding pavement acceleration and temperature. Since the system measures actual pavement response, it can also be used for structural health monitoring of pavements. The wireless system could enable many new applications in infrastructure monitoring and intelligent transportation. We present energy-efficient algorithms for three such applications:

1. **Automatic vehicle classification:** Without disrupting the traffic flow, moving vehicles are automatically classified into categories such as passenger cars, vans, buses, and trucks of various sizes and configurations.
2. **Weigh-In-Motion (or load estimation) for trucks:** Trucks traveling at highway speeds (or lower) are automatically detected and weighed with high accuracy ($\pm$ 10% error).
3. **Displacement (or deflection) estimation:** While MEMS sensors that can measure acceleration accurately have been available for many years, methods for estimating displacement from these measurements are lacking. Displacement estimation can enable real-time pavement condition monitoring and many other applications outside the area of infrastructure monitoring.

We now give a brief introduction to pavement design and maintenance. This will help the reader understand the broader applicability of the system in enabling smart pavements of the future that can sense damage at an early stage and request for timely repair. We then state the problems addressed by this dissertation and related challenges in Section 1.2. Section 1.3 provides a review of related work in literature and existing commercial technologies in the field. The last section describes the organization of this dissertation.

1.1 History of pavement design

We start this section by describing the famous American Association of State Highway Officials (AASHO) road test, data from which is still used for pavement design. We then

give a brief history of the LTPP program and the data it collects to reduce the associated design and maintenance costs. The mechanistic-empirical pavement design is then described to explain the state-of-the-art techniques for pavement design. We end the section with a discussion of our current and future needs in pavement monitoring.

1.1.1 AASHO road test

The AASHO Road Test was a series of experiments carried out by the American Association of State Highway and Transportation Officials to determine how traffic contributed to the deterioration of highway pavements. The test was carried out in the late 1950s in Ottawa, Illinois, and is frequently quoted as a source of experimental data for pavement design, vehicle taxation etc. The results were used to develop a pavement design guide, first issued in 1961, with major updates in 1972 and 1993.

The AASHO road test introduced many new concepts in pavement engineering, including the equivalent single-axle load (ESAL). An ESAL is defined as the single axle load that causes the same amount of pavement damage as the given load and axle type. It was also found that the heavier vehicles caused pavement deterioration in a much shorter time than light vehicles, and the damage caused by a loaded axle is related to the 4th power of its weight. A major drawback of the AASHO road test data was that it provided little information on long-term environmental effects and no direct information on pavement response and performance under actual highway traffic [51]. At the conclusion of the AASHO Road Test, a program was purposed to establish a series of test roads around the US to study the effects of different environments, soil types, and pavement materials on pavement performance [13].

1.1.2 LTPP program

The Long-Term Pavement Performance (LTPP) program was started in 1987 with a goal of extending the life of pavements through investigation of long-term performance of various designs of pavements under different conditions. Data collection began in 1989 from nearly 2500 in-service pavement test sections throughout US and 10 Canadian provinces. The LTPP program has operated under the US Federal Highway Administration (FHWA) since 1992 [6]. The benefits of the program over 20 years of its existence can be broadly classified into three categories: LTPP database; advances in pavement performance measurement; and contributions to pavement design and management.

The LTPP database has been the foundation for pavement research and advancements in pavement design and management. It is composed of 21 modules containing data categories such as truck volumes and loads, climate, materials testing, and pavement testing for distress (longitudinal and traverse road profile, and structural response) [6]. Truck volume and weight data are collected through numerous automatic vehicle classification (AVC) systems, Weigh-In-Motion (WIM) systems, and static weigh stations. The LTPP data led to the

development of Mechanistic-Empirical Pavement Design Guide (MEPDG) that addresses many limitations of the previous design procedures from AASHO test.

1.1.3 Mechanistic-empirical pavement design guide (MEPDG)

The previous design procedures involved direct calculation of required layer thicknesses for the pavement to accommodate specified traffic, materials, environmental conditions, and performance requirements over a particular design life. However, the results depended on empirical data that was collected over a limited range for loading and environmental conditions. The Mechanistic-Empirical (M-E) design uses both theoretical pavement response models and the experimental data. The new approach evaluates the impact of traffic, climate, materials, and subgrade stiffness on performance and accounts for the interactions among these components [12, chap. 1].

The MEPDG predicts individual performance measures (e.g. transverse cracking, fatigue, smoothness, rutting) based on data inputs for a given pavement section [12, chap. 1]. The design engineer follows the following iterative process under this approach: assume an initial pavement structure, compute distresses over the required design life, adjust the initial design if any distress exceeds acceptable levels, re-compute the distresses, and continue in this way until a satisfactory design is achieved. Three different levels of design are available and the most accurate level requires extensive and detailed data inputs. As an example, traffic load data might be available at different levels of detail. The M-E design uses the axle load distribution obtained from traffic measurements to simulate the effect of applied loads on pavement performance. At the most detailed level, truck weight information could be available from a nearby WIM system. At the intermediate level, truck count and classification data can be used to estimate the axle load distribution. At the lowest level of detail, only vehicle count data might be available. The best pavement distress predictions are obtained when accurate weight data is available.

1.1.4 Pavement monitoring needs of today and the future

The FHWA recommends an increase in the number of WIM systems on highways to adequately monitor the truck loads being applied. However, current technologies are too expensive ($\approx$ \$0.5 M/installation) and cheaper WIM systems are needed. Further, since pavement response is significantly affected by pavement temperature [49, 58], the M-E design would also benefit from accurate temperature data. The current approach uses data obtained by the nearest weather station to estimate pavement temperature. Lastly, an assessment of pavement strength for rehabilitation projects is typically done using a Falling Weight Deflectometer (FWD) test. Calibrated loads are applied to the pavement and the measured pavement deflection is used to back-calculate the stiffness of various layers of the pavement structure. The FWD tests, however, are expensive and disrupt the normal traffic flow as they require lane closures. Consequently, there is a need for novel measurement systems that can

estimate pavement-stiffness without affecting highway traffic [7]. A pre-weighed truck can be used for this purspose if the corresponding pavement displacement can be measured [2].

1.2 Problem statements and challenges

1.2.1 Problem statements

The dissertation addresses 4 main problems. The first problem involves building a WSN to detect on-going traffic and measure corresponding ultra-low pavement vibrations. For detecting traffic, we use off-the-shelf vehicle detection sensors[2] which can also estimate vehicle speed and length [31].

Problem 1.1. *A vehicle moves at an unknown speed, ranging from a few miles per hour to highway speeds. Design a wireless vibration sensor that can be embedded in the pavement to measure corresponding pavement acceleration and temperature.*

Once a capable WSN is available, we can use it to solve the following application-specific problems.

Problem 1.2. *A vehicle of N axles moves at an unknown speed, ranging from a few miles per hour to highway speeds. Use the designed WSN to automatically detect axles of the vehicle and classify it using FHWA's vehicle classification scheme.*

Problem 1.3. *A truck of N axles moves at an unknown speed, ranging from a few miles per hour to highway speeds. Use the designed WSN to estimate the individual axle weights and the gross-weight of the truck.*

The above two applications can enable a cost-effective way of monitoring traffic for pavement design. To enable real-time pavement condition monitoring, we need to estimate displacement from noisy acceleration data.

Problem 1.4. *Loads are applied to the pavement using real trucks or nondestructive testing tools such as the FWD. Estimate the pavement displacement accurately using noise-corrupted acceleration measurements.*

1.2.2 Challenges

The system needs to overcome several challenges in order to solve the above problems:

Measurement
The road pavement is designed to experience very small vibrations from vehicle movement [16]. The vibration sensor must measure these small vibrations while being immune

[2]Provided by Sensys Networks, Inc.

to the high environment noise arising from vehicle sound and traffic in neighboring lanes. The sensor must also be well-coupled to the pavement for accurate measurements. Finally, the collected data must be transmitted reliably to the AP in a challenging environment with vehicles frequently blocking the line-of-sight between the sensor and the AP.

Design
The sensors must be robust enough to withstand the tire forces. The system should also be insensitive to vehicle wander i.e. vehicles traveling off the lane center. It should also be cost-effective and convenient to install and maintain, and have a lifetime of at least 4 years [15].

Modeling
The relationship between applied axle load and pavement acceleration is not well-understood. Most pavement response models use pavement deflection but estimating pavement deflection from acceleration is a challenging problem in itself [8, 29]. Moreover, the response is highly dependent on pavement temperature and speed of the vehicle, and these variables must be properly accounted for. Another challenge is to estimate static load from dynamic load [16]. The suspension oscillations caused by road roughness lead to a time-varying load being applied to the pavement, and therefore, weight estimated by a single sensor (at a fixed location) can be much different from the static weight. Current technologies deal with this challenge by installing a smooth concrete pavement around the force sensor but this approach is very expensive and undesirable for our system.

Signal processing
The pavement response at any given time and location is an accumulated response due to all vehicles in the vicinity, therefore response due to other vehicles needs to be filtered out. An even harder challenge is to extract the pavement response due to each axle because at any given time, all axles of a vehicle are affecting sensor measurements. Lastly, the signal processing algorithms have to be simple enough for real-time execution and efficient enough to conserve energy for a longer sensor lifetime.

1.3 Related work

We identify four areas related to this work: applications of sensor networks in transportation, applications in infrastructure monitoring, AVC and WIM technologies, and algorithms that estimate displacement from noisy acceleration.

Intelligent transportation
Applications of WSNs in transportation have been growing over the last decade. WSNs have been used for vehicle detection using wireless magnetic sensors [19, 37, 55]. Multiple magnetic sensors, spaced at a fixed distance apart from each other, are used to accurately estimate the speed and length of the vehicle. These sensors have proven to be a very effective replacement of wired magnetic loops which are more expensive to install and maintain [31]. They have

also been used for vehicle re-identification and for accurate travel time predictions [38].

Wireless sensors can be used to determine available spots in a parking lot. In fact, companies like Sensys Networks, Nedap AVI, and Streetline provide commercial wireless sensors for parking lot management. Further, new cars include increasingly sophisticated sensing technologies, and wireless communication between vehicles has been proposed to increase road safety by inter-vehicular information sharing [53]. While sensor networks are having a major impact on traffic monitoring, much less has been done in terms of monitoring the road infrastructure itself.

Infrastructure monitoring

Structures like bridges, buildings, dams, pipelines, among others, are complex engineered systems that are vital for economic and industrial prosperity. In order to ensure the safety of these structures, standardized building codes and design methodologies have been created. However, these structures are frequently subjected to harsh environmental and loading conditions that results in long-term structural deterioration [43]. Over the last decade, advances in MEMS sensors and their availability on small wireless motes have led to an increased interest in using these easy-to-install sensors for structural health monitoring. The measurements are processed using damage detection algorithms, for example in [36], to screen for structural damage.

Monitoring infrastructures using network of accelerometers has been studied for structural monitoring of bridges [35], buildings [17], and underground structures such as caves [40]. Many other examples exist in literature. Wired embedded sensors in concrete structures have been investigated [46] but usually require complex installation procedures and have limited lifetime if used in roads.

A novel self-powered strain sensor for continuous structural health monitoring of pavements is presented in [7]. However, the sensor data is read through a RFID reader mounted on a moving vehicle. For pavement monitoring and management systems, it would be more beneficial if the sensors can report useful data to a remote server in order to help authorities take an appropriate action.

Estimating displacement from acceleration

Estimating pavement deflection (or displacement) from acceleration is a challenging problem in itself. Simple double integration amplifies the low frequency noise leading to a large unpredictable drift [17]. Popular techniques for drift correction include fitting a polynomial to estimate drift, and subtracting it from the calculated displacement [29, 8]. However, corrected signals are highly sensitive to the choice of the drift polynomial, and these techniques do not perform as well for measurements with a low signal-to-noise ratio (SNR).

WIM and AVC technologies

In 1985, Maine Department of Transportation developed an algorithm, called Scheme "F", that can accurately classify vehicles using their axle count and inter-axle spacings [33]. This scheme enables vehicle classification using the time of arrival for each axle and speed of the

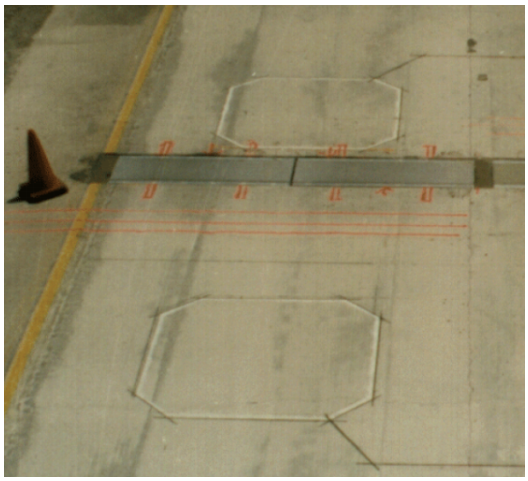
vehicle. It also makes AVC and WIM two very closely related problems. In AVC systems, the goal is to predict whether a load is being applied or not whereas in WIM systems, the magnitude of this load is also estimated. Regardless of the WIM technology, it is assumed that a WIM system includes the capability of vehicle classification. On the other hand, there exist AVC systems that cannot measure vehicle weights.

AVC technologies. Systems for vehicle classification can be divided into intrusive and non-intrusive technologies [27]. Most common non-intrusive schemes are based on digital imaging [30], range sensors [32], acoustic [47], infrared [42], and microradar sensors [18]. These systems suffer from loss of accuracy with varying daylight, weather, and traffic conditions; have special requirements for setup; and multiple systems are required for high accuracy. More importantly, they may require special arrangements for measuring multiple lanes at sites without an overpass. The most common intrusive systems are based on either piezoelectric sensors or magnetic loop detectors. Intrusive technologies are generally more accurate but very expensive to install and maintain.

More closely related to this work, a vehicle classification scheme based on vehicle length and magnetic signature classification [19] has been proposed and evaluated, but it was shown to be very data intensive. A WSN system for vehicle detection and classification was proposed in [26] that combines acoustic, infrared, and seismic sensing. However, its main application is for classification of vehicles in open fields, and its performance is dependent on environmental and other conditions.

WIM technologies. The most widely used WIM systems consist of a pair of wired magnetic loops and a force sensor, as shown in Figure 1.2(a). The magnetic loops detect on-going vehicles and estimate their speed. The force sensors (piezoelectric plates, load cells or bending plate sensors) measure the instantaneous load applied by the tires of a vehicle. A major drawback of these technologies is that they require smooth concrete pavement to be built around the force sensors to achieve the desired accuracy. Figure 1.2(b) shows the typical installation procedure: the force sensors are installed first and smooth concrete pavement is built around it. Pavement roughness excites the vehicle's suspension system, thus, causing the instantaneous axle load to be different from the static load. The difference between the instantaneous and static load, known as the *dynamic component* of applied load, is reduced by having a smooth pavement. However, this construction increases the system cost and the installation time, typically requiring several days or even weeks of lane closure. As an alternative to this approach, the use of *multiple force sensors* on existing pavement has been suggested to improve the estimate of static load [16], but current technologies are too costly to make this approach practical.

While WIM technologies have not advanced much in the last decade, focus has shifted on using multiple WIM sensors to improve system accuracy as opposed to requiring special-material pavement near the sensors [27, 14]. A novel WIM sensor based on perturbation theory of microwave resonant cavities is presented in [41], and a special fiber optic sensor based on measuring light loss under mechanical stress is discussed in [45]. However, both sen-



(a)

(b)

Figure 1.2: Current WIM technologies. (a) A typical WIM system consisting of wired loops detectors and force sensors. (b) Installation process for a WIM station.

sors were tested in a controlled laboratory setting, and challenges regarding road installation and sensor durability under heavy loads were not addressed.

1.4 Dissertation organization

In Chapter 2, we describe the wireless sensor network used to measure pavement response and to detect on-going vehicles. The vibration sensor was specifically created for this project and is described in detail. The sensor uses a wireless communication protocol designed to maximize lifetime. The chapter also presents the sensor calibration and testing results.

Chapter 3 describes the used test sites and the experiments performed during the course of the project. The purpose of each experiment is described along with the sensor layout. We also provide references to later chapters where the data corresponding to each experiment is analyzed.

In Chapter 4 we discuss the problem of axle detection and automatic vehicle classification. Two cases are considered: classification using a sparse sensor array and using a dense sensor array. We describe an energy-efficient algorithm that distributes the computation between the AP and the sensors. The sensors extract useful features from the raw acceleration signal and compress the data before wireless transmission. The AP combines the compressed data from all sensors to automatically classify vehicles. The algorithm is tested using 525 random trucks on highway.

In Chapter 5 we discuss the problem of load estimation or WIM. We use the observations

from the various experiments in Chapter 3 and propose a new pavement-vehicle interaction (PVI) model for the purpose of estimating individual axle loads. A procedure to estimate axle weights and gross-weight is presented using all raw acceleration data. We test the system using 75 randomly chosen trucks that were weighed at a static weigh station for accurate ground truth measurements. We end the chapter by estimating the accuracy of the wireless WIM.

Chapter 6 starts by presenting various improvements in the PVI model and the load estimation procedure. Since collecting all raw data for load estimation is not practical, we present an adaptive data compression algorithm used by the sensor to extract the required information from raw data. The AP combines the data reported by all sensors and estimates the individual axle load and the gross weight of the truck. The system was tested using the same 75 random trucks and the results obtained were better than a nearby commercial WIM system.

Chapter 7 discusses the problem of displacement estimation from noisy acceleration measurements. Two novel algorithms are proposed: one applicable for real-time pavement monitoring and the other for controlled experiments using testing tools such as the Falling Weight Deflectometer (FWD) and the Heavy Vehicle Simulator (HVS). The results are compared against reference measurements from commercial sensors and found to be very accurate.

Chapter 8 presents the conclusions and future work.

Chapter 2

System Description

This chapter describes the wireless sensor network (WSN) used for detecting on-going vehicles, and measuring corresponding pavement acceleration and temperature. We start by describing the WSN components: vehicle detection sensors, vibration sensors, and the access point. The low-noise vibration sensor was built specifically for this project and we present that in more detail. We then explain the wireless communication protocol and end the chapter with calibration and testing results for the system.

Figure 2.1 shows a general setup of the sensor network. The system is comprised of wireless sensor nodes that are embedded in the pavement, and an access point (AP) installed on the roadside. The sensor nodes are synchronized in time with the AP, and report data for vehicle detection, pavement acceleration and temperature. The AP controls the nodes and records the incoming data coming from the sensors and other optional peripherals such as a PTZ camera.

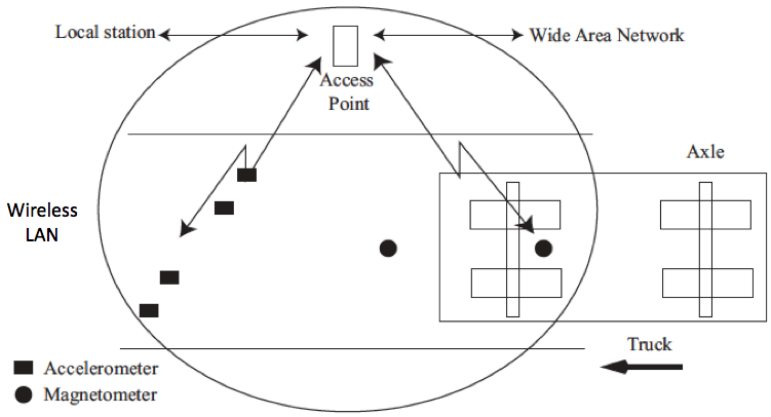


Figure 2.1: Image showing a general setup of the system.

2.1 Sensor network components

2.1.1 Wireless sensor nodes

A wireless sensor node consists of 3 main components: a sensor, a microprocessor, and a wireless transceiver (or radio). Figure 2.2 shows the block diagram for a typical wireless sensor node. The sensor converts physical quantity like temperature, acceleration, magnetic field into an analog voltage signal. This signal is passed through an analog filter (signal-conditioning stage) and sampled using an analog-to-digital converter, typically available on microprocessors. The sampled data is transmitted wireless using an on-board wireless transceiver.

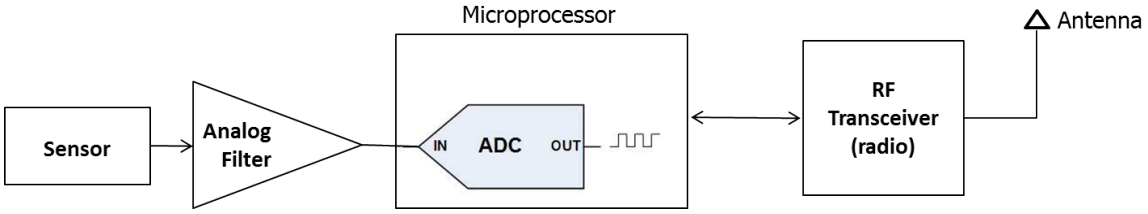


Figure 2.2: Bloack diagram for a typical wireless sensor node.

For convenience, we omit the word *node* and refer to wireless sensor nodes as wireless sensors from hereon. There are two different types of wireless sensors in our system: vehicle detection sensors that detect the presence of a vehicle using a magnetometer, and vibration sensors that measure the vertical acceleration of the road pavement.

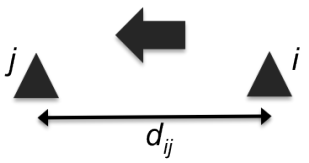
Vehicle detection sensor

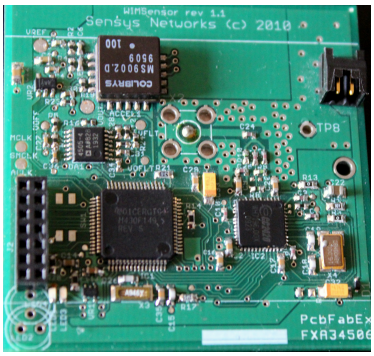
The vehicle detection sensor measures changes in the local magnetic field to infer the presence of a vehicle. The sensors have been shown to be very accurate for vehicle detection and have a lifetime of 10 years [31]. Each sensor samples the on-board magnetometer at 128 Hz and uses an edge-detection algorithm to estimate the arrival time t_a and the departure time t_d of a vehicle. A pair of sensors (i, j) can be installed at a fixed distance d_{ij} apart from each other to estimate the speed and length of a vehicle. Given the arrival times t_{ai} and t_{aj}, and the departure times t_{di} and t_{dj} at sensor i and j, the speed v, length L, and the time window T_i corresponding to the vehicle at sensor i can be estimated as,

$$v = \frac{d_{ij}}{|t_{aj} - t_{ai}|}, \tag{2.1}$$

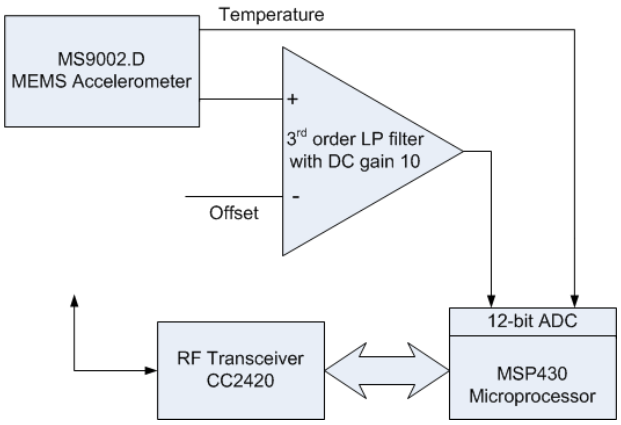
$$L = v|t_{di} - t_{ai}| \equiv v|t_{dj} - t_{aj}|, \tag{2.2}$$

$$T_i = [t_{ai},\ t_{di}]. \tag{2.3}$$





(a)



(b)

Figure 2.3: Wireless vibration sensor. (a) Picture of the sensor board. (b) Block Diagram of WIM Sensor.

Vibration sensor

The vibration sensor is needed to measure the vertical acceleration of the pavement but has many unique design challenges: sensor needs to have a very low noise level in order to measure the ultra-low road vibrations; the sensor needs to sample fast enough to capture the transient pavement vibrations and yet have low power consumption for a long lifetime; the vibrations due to truck engines and other sounds should have a minimal effect on sensor readings; the sensor must be well coupled to the pavement and be strong enough to withstand tire forces from heavy vehicle traffic.

Figure 2.3 shows a picture and the block diagram of the sensor. The MEMS accelerometer converts mechanical vibrations to an analog voltage signal. This signal is filtered to reduce noise and sound interference, and amplified to increase the sensor sensitivity ($\frac{V}{g}$). The filtered signal is then sampled by a 12-bit analog-to-digital convertor (ADC) and the samples are transmitted via a radio transceiver. The accelerometer also includes a temperature sensor whose output can be sampled by the ADC [24]. The accelerometer and analog filter are powered by a 2.5 V supply voltage that can be turned off by the microprocessor when needed.

Accelerometer Selection. The specifications for the vibration sensor were obtained using simulations done in [52] and previous tests on a highway [9]. It was found that the sensor needs to have a resolution of 500 μg at 50 Hz bandwidth, and a range of $\pm$ 200 mg. For this purpose, we evaluated two different analog MEMS accelerometers: SD1221-005 from Silicon Designs and MS9002.D from Colibrys. These were selected for evaluation from amongst many others in the market because of their very low noise density ($\frac{\mu g}{\sqrt{Hz}}$) and high sensitivity ($\frac{V}{g}$) [56, 24, 20]. As seen in Table 2.1, SD1221-005 has higher sensitivity and lower noise

density than MS9002.D, both of which are very desirable. However, SD1221-005 consumes more than 20 times the current consumed by MS9002.D and has to be operated at a much higher voltage. Consistent with the table, SD1221-005 outperformed MS9002.D during our evaluation of the devices but both devices achieved our aimed resolution of 500 μg. MS9002.D was selected over SD1221 since it is more suitable for embedded wireless applications due its low power consumption.

Table 2.1: Comparison of accelerometers SD1221-005 and MS9002.D.

	SD1221-005	MS9002.D
Sensitivity (V/g) @ 5V	2	1
Noise Density ($\mu g/\sqrt{Hz}$)	5	18
Current Consumption (mA)	8	≤ 0.4
Min. Operating Voltage (V)	4.75	2.5

Anti-Aliasing filter design and sound isolation. The filter design is very important to obtain the desired range ($\pm$ 200 mg), resolution ($\leq$ 500 μg), and rejection of sound. Typically, microprocessors for embedded systems have an on-board ADC with 10-12 effective-number-of-bits (ENOB). If we sampled the accelerometer without any filter gain (or amplification), the quantization noise from a 12-bit ADC (600 μV at 2.5 V operating voltage) would dominate the noise from the accelerometer (< 100 μV). The resolution of the system would be lowered due to relatively high quantization noise. As we increase the filter gain, the noise from the accelerometer in μV increases but the noise in μg remains unchanged. However, there is an upper bound on filter gain as it reduces the effective accelerometer range ($\frac{\pm 2g}{\text{filter gain}}$). A filter gain of 10 achieves the desired acceleration range and resolution.

Another concern that is often underreported about systems using accelerometers is their sensitivity to sound. A high resolution accelerometer behaves like a low-quality microphone within the device's bandwidth. A single pole low-pass filter is not sufficiently aggressive to prevent interference from sound under 1 kHz, and a simple clapping sound near the accelerometer is recorded as vibration. Thus, any sensor deployed in the field is vulnerable to this interference. It was learnt in our previous research [9] that a 3rd order (or higher) low-pass filter with a cutoff frequency of 50 Hz is sufficiently aggressive to filter out most of the sound in the audible spectrum. The selected filter has the following transfer function,

$$H(j\omega) = \frac{10}{(1 + \frac{j\omega}{50})^2(1 + \frac{j\omega}{500})}. \tag{2.4}$$

Analog-to-digital conversion. We used the 12-bit ADC module on MSP430F149 microprocessor chip for sampling. The module implements a SAR (successive approximation register) core, includes an internal voltage reference generator, and allows up to 16 independent ADC samples to be converted and stored without CPU intervention. We simultaneously

sample at 512 Hz two independent signals: the filtered acceleration, and the sensor temperature. Since pavement temperature changes slowly, we average the temperature readings so that only a single temperature reading per wireless packet is sent out.

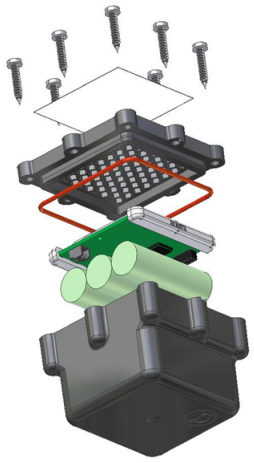


Figure 2.4: Packaging of a sensor in a sealed case.

Radio and antenna. We use a wireless transceiver from Texas Instruments, CC2420, with an effective data rate of 250 kbps and operable in the 2.4 GHz ISM band. Since the sensors are embedded in the road, only a hemispherical wireless coverage is required. We use a patch antenna to achieve this.

Sensor casing. In order to withstand large forces in a harsh environment, the sensors must be packaged for durability before installation. The circuit board and the battery are placed in a hard plastic casing as shown in Figure 2.4. The casing is then filled with fused silica and sealed air tight. This protects the electronics from rain water, oil spills etc on the road and further attenuates interference from sound.

2.1.2 Access point

Figure 2.5 shows a block diagram for the access point[1]. This equipment provides remote control and observation of the WSN. The AP contains: (i) a processor with attached radio transceiver and 2 TB hard drive storage; (ii) a power controller that controls power to each connected device; (iii) an ethernet hub through which a local area network (LAN) is setup for devices to communicate with each other; (iv) a 3G modem that acts as a gateway to the wide area network (WAN) and enables remote access to the system; (v) a Wi-Fi bridge and an ethernet data port for local access to the system; and (vi) an optional pan-tilt-zoom

[1] Also provided by Sensys Networks.

(PTZ) camera for taking roadside images. Once a remote computer is connected to the AP, it can communicate with any of the connected devices through the LAN. It can, for instance, use the power controller to turn on/off individual components in the box, send commands to the sensors via the radio, change the settings of the PTZ camera, and start collecting video and sensor data remotely.

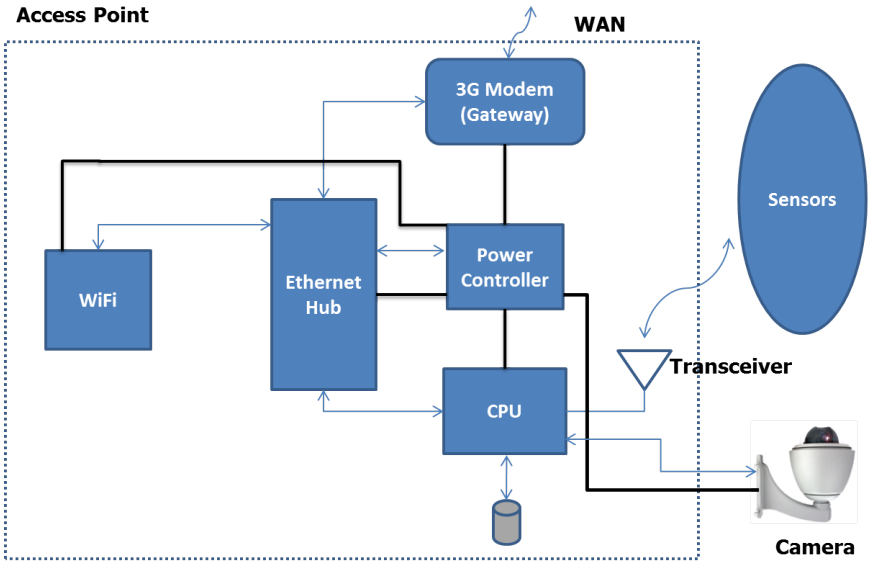


Figure 2.5: Access point.

2.2 Communication protocol

We focus on the communication protocol followed by the wireless sensor nodes and the AP. Other components use standard protocols and are not discussed here. The architecture of the protocol consists of three logical entities: an AP, an optional repeater, and wireless sensor nodes. The AP has a wired and a wireless connection, while the repeater and the sensor nodes only have a wireless connection. The WSN follows a star network topology where every node can directly communicate with the AP. Each node is assigned a unique network address (or node ID) for identification and follows a TDMA (Time division multiple access) schedule to transmit data. The AP periodically sends out clock synchronization information and records the incoming data from multiple sensors and other peripherals.

Physical Layer

The access point and all the nodes use a IEEE 802.15.4 compatible radio transceiver (CC2420). The transceiver uses the 2.4 GHz ISM radio band and can be operated on one of the 16 compliant RF channels.

Medium Access Control (MAC) Layer

The MAC protocol is TDMA based and uses headers very similar to IEEE 802.15.4 MAC layer. Time is divided into multiple frames with each frame about 125 ms long. Each frame is further divided into 64 timeslots, numbered 0 to 63, most of which can be used by the sensor nodes to transmit data. Timeslot 0 is used by the AP to send clock synchronization information and other commands to the sensors. The AP assigns every sensor node unique sets of time slots for data transmission. This schedule ensures that individual sensor nodes stay awake for a minimum amount of time while preventing any packet collisions in the network.

Application Layer

There are three main applications in our protocol: clock synchronization, sensor management, and firmware update.

1. Clock synchronization. This application runs in the background and ensures clock synchronization of all nodes within 60 μs. Sync packets are sent by the AP on a periodic basis with very low jitter. Nodes must first synchronize their clocks before transmitting. When a sensor node first starts, it listens to sync packets every 125 ms. It learns the difference between its clock and the AP's clock, and over time improves its estimate of the AP's clock. As the estimate improves, the node converges to a steady state in which it listens for a sync packet only once in 30 seconds. If a node loses sync, it repeats the above process to get synchronized again. In addition to sending clock information, the AP can also send one of the following commands to individual sensors:

Set Mode to change the operational mode of the sensor.
Reset to set the node to factory defaults.
Set Timeslot to assign a timeslot to a sensor.
Set RF Channel to change the RF channel of a sensor.
Download Firmware for over-the-air sensor firmware updates.
Set ID to change the sensor ID.

2. Sensor Management. This is the most important application for both sensors. For the vibration sensor, the application controls when to turn on the accelerometer and related circuitry, when to sample, and when to wake up the radio to transmit the measurements. There are two main modes in this application: *idle* mode and *raw data* mode.

- ***Idle mode:*** This is the default power saving mode of each sensor node. In this mode, the accelerometer and the related conditioning circuitry are powered off by the microprocessor. Even the microprocessor and the radio are put to a low power state for the majority of time. Once in 30 seconds, the sensor wakes up to receive the sync packet and transmit back a stay-alive packet.
- ***Raw Data mode:*** In this mode the accelerometer and related circuitry are powered on. The microprocessor wakes up every $\frac{1}{512}$ seconds, and samples the temperature and the filtered output of the accelerometer. In addition to waking up for the sync packet, the radio is woken up during the assigned timeslots to transmit the collected data. In one frame, we collect 64 samples of acceleration data with each sample containing 12 bits of information. Including the two bytes for temperature data, we accumulate 98 bytes of information in every frame to transmit.

For the detection sensor, this application is similar. The key difference is that instead of the *raw data* mode there is a *vehicle detect* mode. The magnetometer is constantly sampled at 128 Hz and an edge-detection algorithm is used to determine if a vehicle is present or not. Only in case of a detection is any data transmitted. Since the data throughput from detection sensors is very small, each packet is retransmitted until an acknowledgement is received from the AP.

The AP receives data packets from each sensor, appends useful information such as the timestamp, the Received Signal Strength Indicator (RSSI), the Link Quality Indicator (LQI), and records it into an ASCII text file for processing.

3. Firmware update. In order to allow for maximum flexibility, the protocol allows a user to wirelessly reprogram the entire flash memory of a sensor. The general procedure for downloading new code consists of having the AP transmit new code repeatedly and the sensor updating its code in small pieces. In order to aid the management of code in the flash memory, each firmware or program is appended with a header which contains a description of the program, its address and length, its interrupt vectors, and some other useful information. The download stream contains two copies of the code linked at different addresses. Only the data in addresses that do not overwrite the current running program are updated by the node. Based on all stored program headers after the download, the sensor picks the program that was most recently downloaded, reboots and starts the program.

2.3 Calibration and testing

2.3.1 Sensor calibration

We use the calibration procedure verified in our previous research [9]. The idea is to use gage blocks of different heights to change the inclination of the sensor, thus changing the component of gravity (g) along its sensing direction. Figure 2.6 shows the calibration setup.

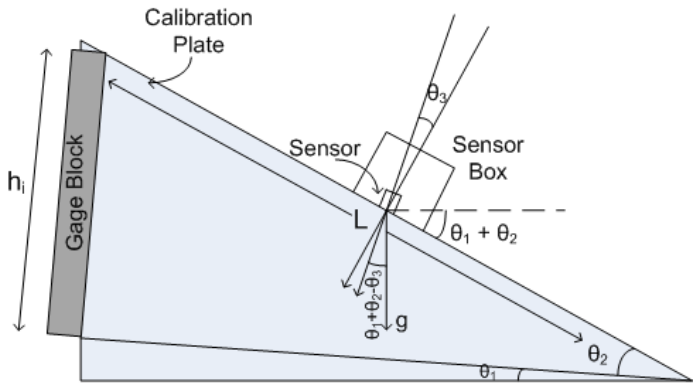


Figure 2.6: Calibration setup for vibration sensors.

We record the accelerometer output for approximately 10 seconds at each height, and calculate the mean and standard deviation of the recorded signal for each height. The mean value is used for sensitivity estimation while the standard deviation is used to estimate the sensor resolution as described below.

Let h_i be the height of the i^{th} gage block, L be the length of calibration plate, $x_i = \frac{h_i}{L}$, and $\bar{v}_i$ be the mean of sensor measurements when gage block i is used. It can be shown that the measured output can be modeled as [9], $\bar{v}_i = A\sqrt{1 - x_i^2} + Bx_i + C + \epsilon_i$, where ϵ_i represents random noise. The unknown parameters A, B, and C can be estimated by linear regression, i.e.

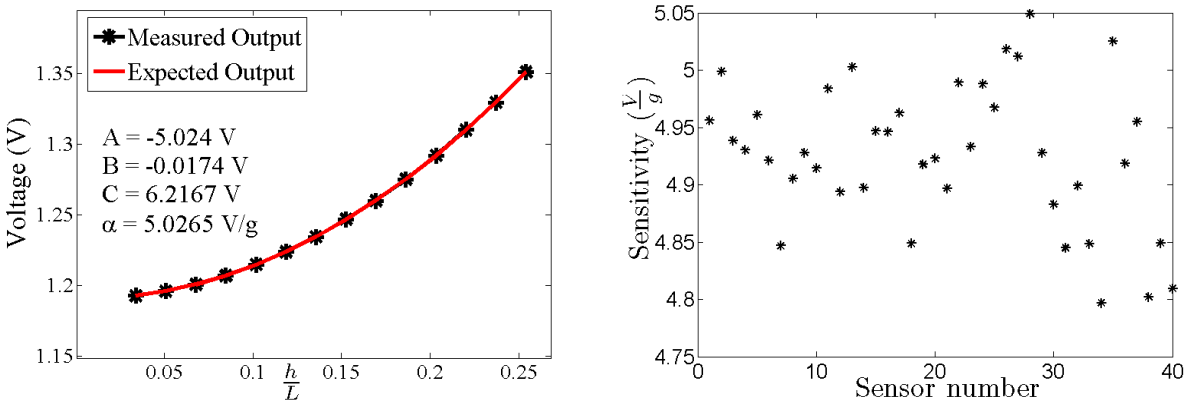
$$(A^*, B^*, C^*) = \arg\min_{A,B,C} \sum_i \left(\bar{v}_i - A\sqrt{1 - x_i^2} - Bx_i - C \right)^2 .$$

The sensitivity $(\frac{V}{g})$ of the sensor can be calculated as $\alpha = \sqrt{A^{*2} + B^{*2}}$. Figure 2.7(a) shows an example of sensitivity estimation for a vibration sensor and Figure 2.7(b) shows the variation in estimated sensitivity for 40 independent sensors. The mean sensitivity is 4.93 $\frac{V}{g}$ and the standard deviation is 0.063.

Let σ_v be the standard deviation of sensor measurements when any one of the gage blocks is used. The resolution of the sensor can be estimated as $\sigma_g = \frac{\sigma_v}{\alpha}$. This gives us a measure of noise in the recorded acceleration. For the sensor in Figure 2.7(a), the resolution was found to be 388 μg. All other sensors had very similar calibration results.

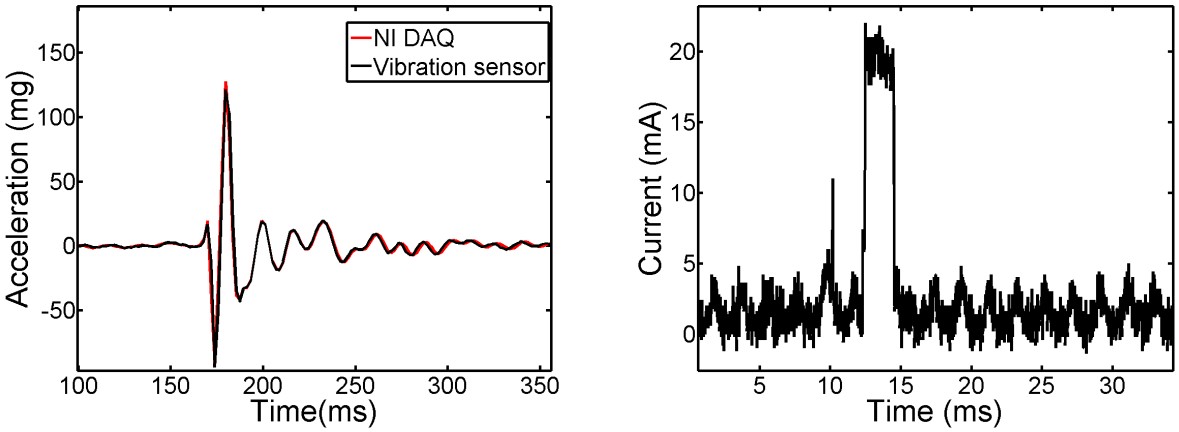
2.3.2 ADC performance

To verify the acceleration sampled by MSP430 of the vibration sensor, we compare the reported measurements with output of the amplifier stage, Figure 2.3(b), sampled by a 24-bit data acquisition system from National Instruments. The vibrations were generated by



(a) (b)

Figure 2.7: Vibration sensor calibration. (a) Sensitivity estimation for a vibration sensor by curve fitting. (b) Sensitvity for different sensors.



(a) (b)

Figure 2.8: (a) ADC comparison. (b) Current consumption in *Raw Data* mode.

applying an impulse-like input (using a hammer) to the surface on which the sensor is placed. Figure 2.8(a) shows that the two signals are in very close agreement. We also measured the sampling frequency of our sensor using an oscilloscope and it was found to be 512 ± 0.2 Hz.

2.3.3 Power Consumption

We estimated the current consumption in different modes by connecting an appropriate resistor in series to the circuit board and measuring the voltage drop across it.

- ***Raw Data mode:*** Figure 2.8(b) shows the current consumption in one of the transmit cycles. The square pulse in the plot occurs when the radio is turned on and actively transmitting. As expected, sending a packet over the radio is the most powering consuming process for the wireless sensor. However, since we only transmit for about 2 ms in 62.5 ms, the duty cycle is relatively low and the average current consumption is found to be 1.96 mA. The oscillations seen in the baseline of this plot correspond to the ADC being turned on for sampling. The average current consumption is lowered down to 1.58 mA by using an external source for ADC's reference voltage. Ignoring the radio transmissions, the average current consumption drops down to 0.37 mA. This number provides a lower limit on the sensor's current consumption.

- ***Idle mode:*** In the idle mode, most of the circuitry is turned off and maximum amount of current is consumed when the radio wakes up to receive a sync packet. The duty cycle for this mode (once in 30 s) is extremely low and the average current consumption was found to be 35 μA.

Using a 7200 mAhr battery, the sensor can last over 20 years in *idle* mode and if we continuously collect raw data from the sensor then it can last more than 6 months. If we collected data from the sensor twice a week for 12 hours each, the sensor can last over 3 and a half years. This lifetime is sufficiently long for research purposes and can be increased for commercial applications by compression of raw data before radio transmissions. We discuss the in-sensor signal processing and compression techniques in the upcoming chapters.

2.3.4 Communication link

Before we analyze the quality of wireless link between sensors and the AP, we must emphasize that our communication requirements depend on the phase of the project. During the initial learning phase, we collect all raw sensor data to known load stimuli and the data transmission scheme minimizes any loss in data corresponding to each load. Since hiring trucks for calibration or any other testing equipment in pavement engineering is expensive, having high reliability of data transfer from sensors to AP is very important. However, once a practical solution to the underlying problem is found, it can be implemented inside the sensor node to reduce the amount of data that needs to be transmitted. In this phase of the project, the required data throughput is generally small and each data packet can be retransmitted if an acknowledgement from the AP is not received back by the sensor. In case of *raw data* mode, however, the limited sensor memory is not sufficient to buffer the data samples for retransmission while the sensor waits for an acknowledgement. Therefore, any packet loss leads to loss of useful data samples.

We analyze the quality of wireless communication for the setup shown in Figure 3.8(a). To quantify the data loss we define vehicular data loss rate ($VDLR$) as the percentage of data

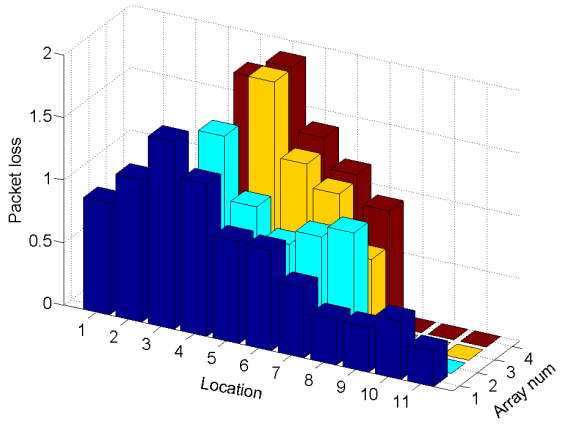


Figure 2.9: Overall packet loss rate for the WSN.

samples lost when a vehicle is over the sensors i.e.

$$VDLR = \frac{number\ of\ lost\ data\ samples \big| vehicle\ is\ over\ the\ sensor}{total\ number\ of\ samples\ transmitted \big| vehicle\ is\ over\ the\ sensor} \times 100.$$

When each sample is transmitted only once, the number of lost data samples is proportional to number of lost data packets. We can define vehicular packet loss rate ($VPLR$) as

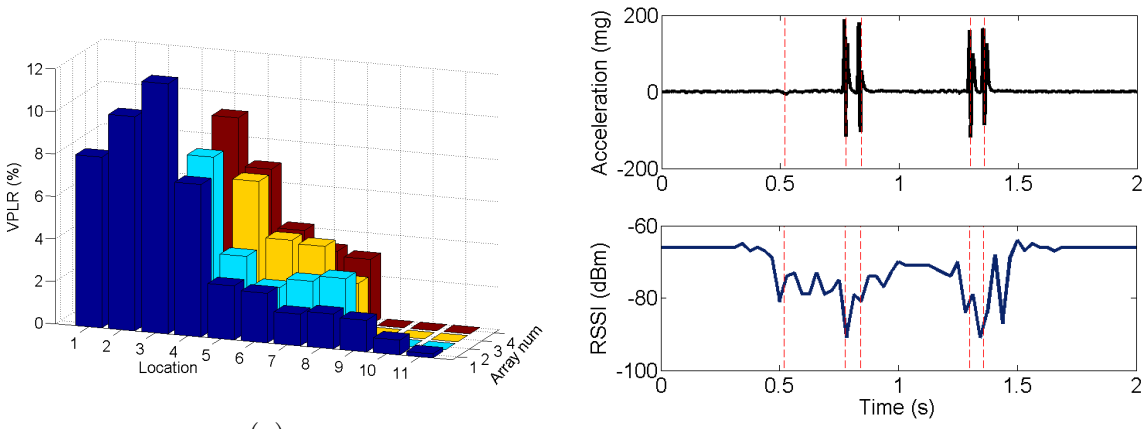
$$VPLR = \frac{number\ of\ lost\ packets \big| vehicle\ is\ over\ the\ sensor}{total\ number\ of\ packets\ transmitted \big| vehicle\ is\ over\ the\ sensor} \times 100.$$

The overall packet loss rate (PLR) is the percentage of packets lost regardless of whether a vehicle is present over the sensor or not,

$$PLR = \frac{number\ of\ lost\ packets}{total\ number\ of\ packets\ transmitted} \times 100.$$

Figure 2.9 shows the PLR profile (or equivalently profile for overall data loss rate) obtained by using a dataset of 525 random trucks from Feb 14, 2012[2]. Each sensor location is defined by its array number and location number that increments from 0 to 12. Note that sensor at location 11 of Array 1 is closer to the AP than sensor at location 1. There are two trends expected for this setup: (i) PLR should increase as the distance between the sensor and AP increases because the received signal strength (RSSI) decreases; (ii) PLR should be higher for locations that frequently have their line-of-sight to AP blocked by truck wheels. The plot shown confirms these trends. PLR in each lane increases as the location number decreases

[2]The sensor at location 4 of Array 3 was not used on this particular day.



(a) (b)

Figure 2.10: (a) VPLR for each sensor. (b) Measured RSSI while a 5-axle truck passes over the sensor.

and there are local peaks at location 3 and 10 in Array 1 that could correspond to regions that are most frequently blocked by traffic. It should be noted that the local peaks depend on the traffic lane and general traffic trends. For example, drivers in the right-most lane of a highway might prefer driving near the right edge of their lane as there is no traffic on their right (at least in North America!).

The packet loss profile shown in Figure 2.10(a) corresponds to an average increase of 300% in data loss rate when a vehicle is present over the sensors. This increase can be explained by the reduction in received signal strength (RSSI) measured by the AP. Figure 2.10(b) shows an example of measured RSSI when a 5-axle truck goes over a vibration sensor; also shown are the corresponding measurements. The red lines denote the time when each truck axle reaches the sensor. The RSSI is lowered as the truck goes over the sensor and the reduction is quite significant when axles are close to the sensor. In fact, for this particular example each axle group can be detected from the RSSI values alone, as seen in the plot.

Since we are primarily concerned with VDLR, is this data loss low enough for our applications?

We use the load estimation application as an example. As discussed later in this dissertation, multiple vibration sensors are installed in the middle of each lane for measuring pavement response in order to estimate axle loads. The application requires that at least 3 out of 5 middle sensors in each array successfully report all relevant data samples for each truck. Since renting calibration trucks is expensive, this requirement makes sure that we successfully collect data for majority of the trucks. For the most frequently used truck type on highways (FHWA class-9), a speed greater than 45 *mph* implies that at most 20 data packets will

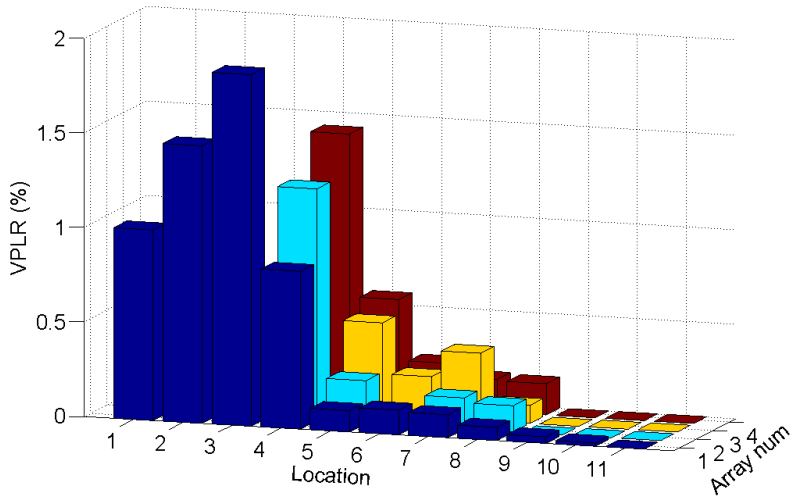


Figure 2.11: Vehicular data loss rate after adding data redundancy in transmission scheme.

be needed to transmit all required samples. We assume that packet transmission for every sensor is an independent event and does not depend on the past history (or the future) of event's success or failure. Let p_i be the probability of packet drop for the sensor at location i of some array, then we can calculate the probability (P_i) that at least one out of 20 data *packets* is not successfully transmitted,

$$P_i = 1 - (1 - p_i)^{20}.$$

This can be considered as the probability of sensor's failure to transmit all relevant data for a class-9 truck. If $P_{success}$ is the probability that at least 3 out of 5 sensors from some array have successful data transmission for the truck, then it can be written as

$$\begin{aligned} P_{success} &= P(no\ sensor\ fails) + \\ &\qquad P(at\ most\ 1\ sensor\ fails) + P(at\ most\ 2\ sensors\ fail), \\ P_{success} &= \prod_{i=1}^{5}(1 - P_i) + \sum_{i=1}^{5} P_i \prod_{j \neq i}(1 - P_j) + \sum_{i=1}^{5}\sum_{j>i} P_i P_j \prod_{k \neq i,j}(1 - P_k). \end{aligned}$$

It should be noted that we assume the presence of vehicle as a general condition in this analysis and ignored the variation from one vehicle to another. We would like $P_{success}$ to be at least 95% for each array as that implies that the data collection process is successful for more than 95% of the trucks. We now return to our original question about whether the data loss rate observed in Figure 2.10(a) is sufficiently low for our application or not. If we use packet loss rate from this figure as an estimate for p_i, we found $P_{success}$ to be only 75%.

Since the data loss rate increases when a vehicle is present, the data transmission scheme could send duplicate packets when a vehicle is nearby. For the sake of easier implementation, we added redundancy throughout our transmission scheme and transmit each data packet twice. The redundant packets can be removed by the AP using a unique identifier for each copy of a data packet. As shown in Figure 2.11, VDLR is lowered significantly after the addition of packet redundancy to the transmission scheme. Using these updated values, $P_{success}$ was found to be $\approx 99\%$ for all arrays. This level of reliability meets our application needs.

Chapter 3

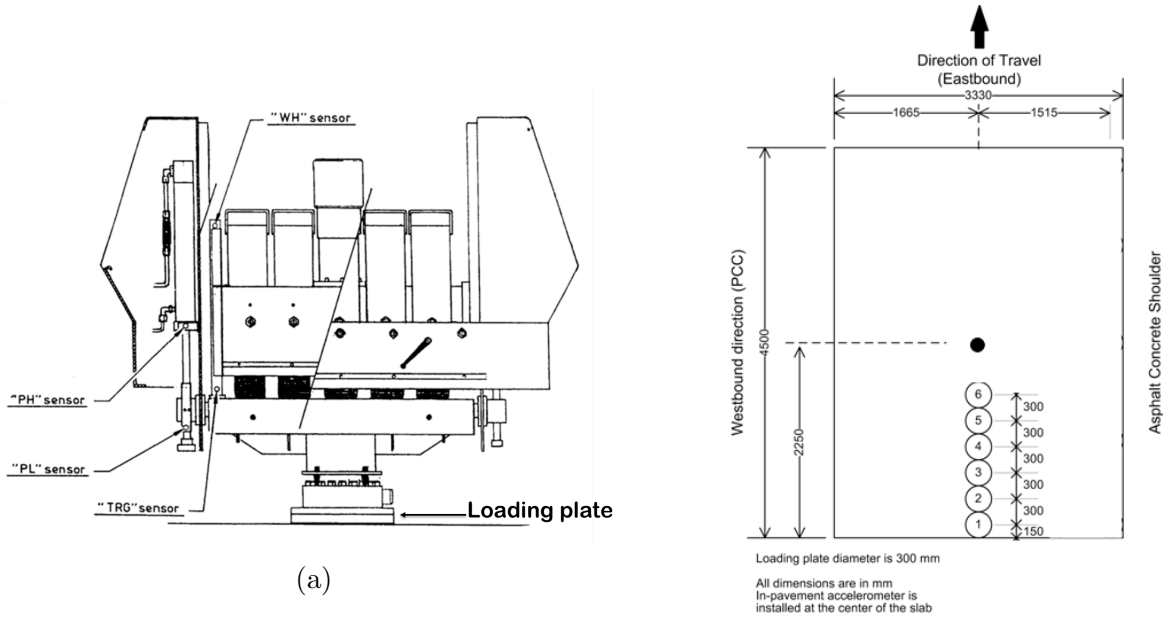
Sites and Experiments

This chapter describes the installation sites and the experiments conducted over the course of this study. A separate section is devoted to each site and experiments are presented in a chronological order. Each section starts with a general description of the test site and then explains the sensor layout, the experiments conducted, and the data collected. References to data analysis for each experiment are also provided.

3.1 Yolo County

The wired version of the WIM sensor, described in [9], was installed in the middle of a concrete pavement road in Yolo county, California. Figure 3.1 shows the site and the installed sensor. The output wire from the sensor is embedded into the road and connected to a National Instruments Data Acquisition (DAQ) box. Data is sampled at 2 kHz and transferred to a local computer. Tests were done in order to verify sensor measurements and to understand the pavement response for real truck loads. These tests are described in detail below.

Figure 3.1: Sensor installation on Road 32A in Yolo County, California. (*left*) Single-lane road in each direction. (*middle*) Sensor location being drilled. (*right*) The output wire from the sensor is embedded in the road and connected to a road-side NI DAQ box for data collection.



(a) (b)

Figure 3.2: FWD experiment. (a) Sketch of the FWD tool. Load pulses are applied to the pavement via the loading plate, and pavement displacement is measured at multiple locations. (b) The wired sensor is installed in the center and the loading plate is moved from locations 1 to 6, applying multiple loads at each location.

3.1.1 Falling Weight Deflectometer (FWD) Test

A falling weight deflectometer (FWD) is a testing device used by civil engineers to evaluate the physical and structural properties of the pavement. A plate, shown in Figure 3.2(a), is lowered to make contact with the pavement, and loads that simulate a rolling wheel are applied directly to the pavement. The magnitude of the load can be varied and measured using an on-board load cell. Deflection sensors, usually geophones, are attached in line with the center of the plate and used to measure surface deflections at multiple nearby locations. The deflection data is then used to backcalculate various pavement parameters and to evaluate structural condition of the pavement [54, 60].

Data Collection. Figure 3.2(b) shows the locations at which FWD loads were applied, numbered from 1 to 6 in chronological order of the tests. Loads of 6, 9, and 16 klbs were applied at every location and each load was repeated three times. For each test, the applied load was measured by the FWD load cell and the pavement acceleration was recorded using the vibration sensor. When the load was placed on locations 5 and 6, the FWD geophones aligned with the top of vibration sensor and the geophone measurements can be used for comparison.

Data Analysis. In Section 5.1, we show that the measured response is highly repeatable for same applied loads. We also show that measured pavement response is linearly proportional to applied load and the response decreases exponentially as the distance from the applied load increases. In Section 7.2.1, we compare the displacement estimated from measured acceleration with the reference measurements from the FWD geophones.

3.1.2 Tests using a real truck

Truck shown in Figure 3.3 was driven over the pavement multiple times and at different speeds. Table 3.1 shows the number of test runs at each speed. The data was used to show the repeatability of measured pavement response, and to analyze the effect of truck speed on pavement response in Section 5.1.

Figure 3.3: FHWA class 6 truck driven on the test section.

Speed	Runs
15	3
20	3
25	3
35	3
45	2
55	2

Table 3.1: Number of truck runs at different speeds.

3.2 UC Davis Pavement Research Center

Accelerated pavement testing is done at this site to characterize different pavement materials and structures. The center has a heavy vehicle simulator (HVS) which repeatedly applies heavy loads to the pavement to simulate the damage caused by real trucks. About 20 years of road deterioration can be caused in as little as three months, thus helping engineers understand the mechanisms of road failure caused by traffic and environmental factors.

3.2.1 HVS experiment

Figure 3.4 shows the heavy vehicle simulator. The wheels shown are loaded with weights and moved back-and-forth on a pavement section. For each HVS test, the user can specify the load, speed, and number of repetitions to be used. The HVS allowed a sufficient load range, 30 kN to 80 kN, but the maximum speed allowed was only 5.41 mph.

Figure 3.4 also shows the location of the six vibrations sensors installed at this site. These locations were carefully chosen in order to measure the pavement response along the direction of wheel movement and along its perpendicular. A linear variable differential transformer (LVDT) is placed over sensor 5 in order to compare the measured surface displacement

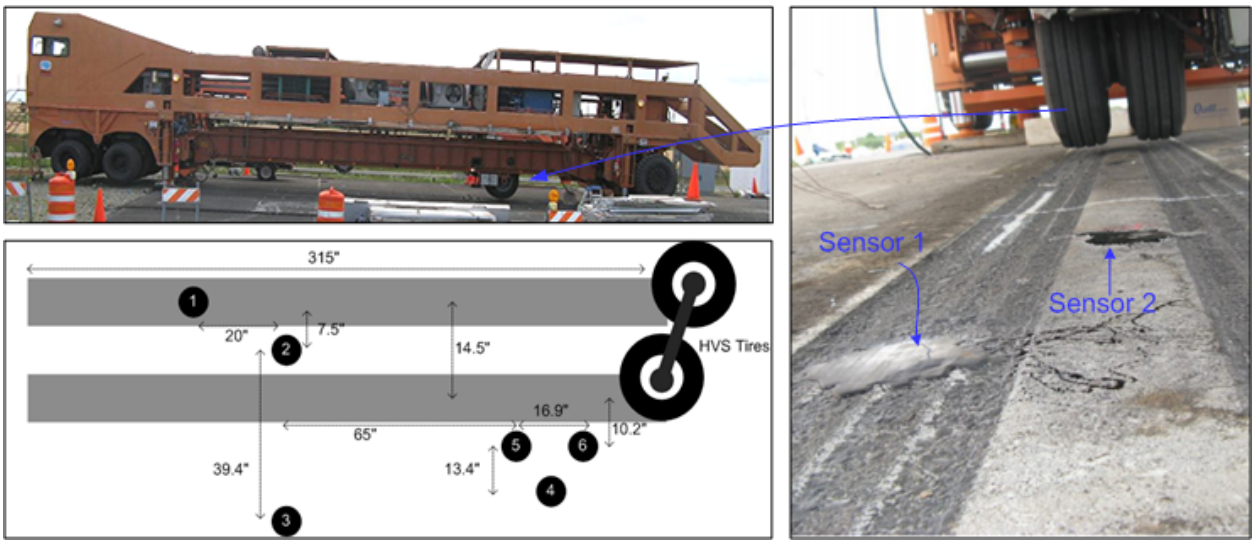


Figure 3.4: HVS experiment. (*top left*) Heavy vehicle simulator. (*right*) Loading wheels of the HVS, and vibration sensors installed in its path. (*bottom left*) Layout for the sensors. All dimensions shown are in inches.

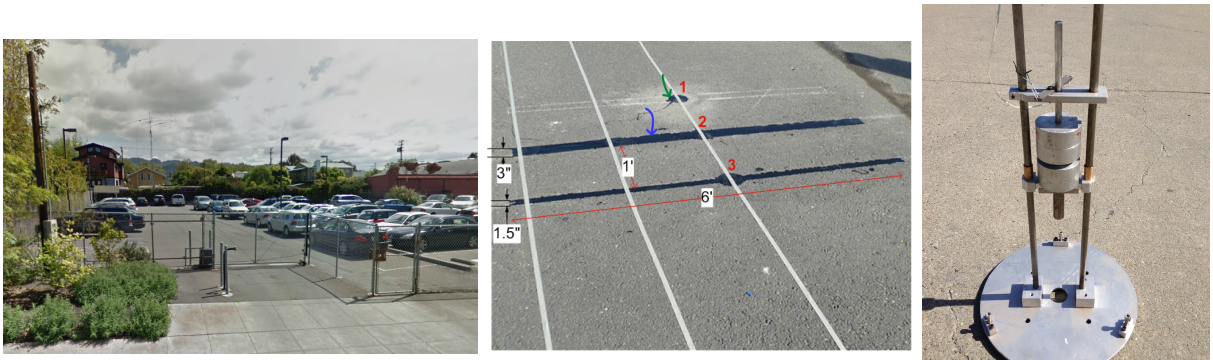
Table 3.2: *HVS experiment design.* Number of repetitions for each load level and speed. Entries are of the form (a, b), where a is the number of repetitions at 92.8 PSI tire pressure and b is the number of repetitions at 104.4 PSI tire pressure.

Speed /Loads	30kN	40kN	50kN	60kN	70kN	80kN
1.86 mph		(60,60)		(60,60)		(60,60)
3.04 mph		(60,60)		(60,60)		(60,60)
4.22 mph		(60,60)		(60,60)		(60,60)
5.41 mph	(0,60)	(60,120)	(0,60)	(60,120)	(0,60)	(60,120)

with the displacement estimated using the vibration sensors. Since sensor 1 was installed underneath the path of one tire, it can be used to understand the response when force is directly applied on the sensor.

Data collection. Table 3.2 summarizes the loads, speeds, tire pressures, and number of repetitions used for a total of 30 HVS tests. The experiment was conducted on May 28, 2010, and pavement temperature varied from 17.9 °C to 19.7 °C. For each test, raw acceleration data from the vibration sensors and reference data from displacement sensors was recorded.

Data Analysis. At low speeds, the pavement response was minimal and of the same order as the noise floor of vibration sensor. Therefore, only the highest speed runs were used for analysis. It is shown in Section 5.1 that measured response is highly repeatable and



(a) (b) (c)

Figure 3.5: (a) Picture of the parking lot. (b) Sensor layout and installation. (c) Picture of the mini-FWD tool.

linearly proportional to applied load. The estimated displacements from vibration sensor are compared with reference displacements in Section 7.2.2.

3.3 Parking Lot

A parking lot near 9th Street and Dwight Way in Berkeley, CA served as a very convenient test bed for our system. Figure 3.5(a) shows a Google street-view image of the parking lot. Testing at this site was very crucial for improving the data communication protocol and for modeling the pavement response.

3.3.1 Mini-FWD experiment

We built a tool similar to the FWD (Section 3.1.1) to understand how vertical vibrations traveled in the pavement and the epoxy. The attached weights, shown in Figure 3.5(c), are dropped from a height of 2 ft to impart a repeatable impulsive force to the pavement.

Figure 3.5(b) shows the experimental setup. Three vibration sensors, numbered from 1 to 3, are installed in a line and surrounded by epoxy channels of width 0, 1.5, and 3 inches respectively. Note that channel width of 0 just means that there is no epoxy channel but we will use this terminology for convenience. The mini-FWD applies loads at multiple locations along each channel in order to understand how they dampen the vibrations. Loads are also applied on the asphalt-epoxy boundary to study the coupling between the epoxy and the pavement.

Data Collection. Loads were applied at five different locations along each channel, and corresponding vibration data was recorded. Table 3.3 displays the weights used, and the

distance of the applied force from the sensor. Each drop is repeated at least 5 times for reliability.

Table 3.3: Distance of the mini-FWD from the sensor and weights dropped.

Distance (inch)	Weights used (lb)
0	1.25, 2, 3.25, 4.5
9	1.25, 2, 3.25, 4.5
18	4.5
27	4.5
36	4.5

Data Analysis. This experiment was primarily used to understand the effect of load location on measured response. It can be seen in Figure 5.4(b) that the measured response decreases exponentially as distance between the sensor and load increases.

3.4 Sunol

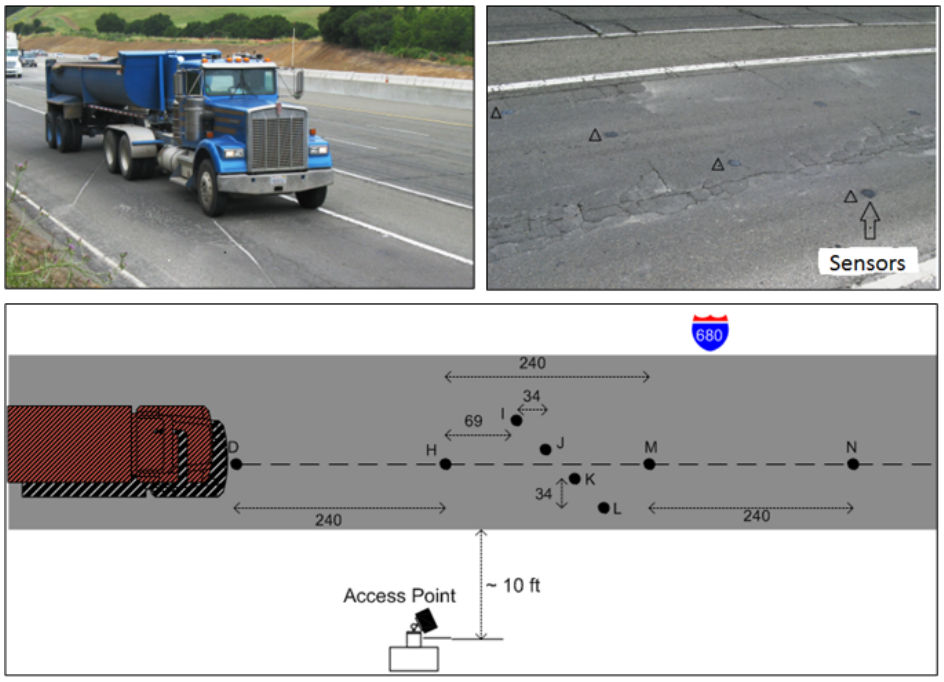


Figure 3.6: On-ramp to I-680N highway. (*top left*) An example truck approaching the sensors. (*top right*) Picture of vibration sensors. (*bottom*) Sensor layout. All dimensions shown are in inches.

With the permission of California Highway Patrol, four vibration sensors and four detection sensors were installed on California Highway I-680N in April, 2010. The site, shown in Figure 3.6, is a highway on-ramp used by vehicles coming from the Sunol weigh station. The site was particularly suitable since the vehicles slowed down at the weigh station and gave us enough time to collect ground truth data. The vibration sensors are strategically installed, as seen in Figure 3.6, to capture the pavement response along the entire width of the lane. Since trucks were trying to get on the highway, they often wandered away from the middle of the lane. Spreading the sensors also ensured that at least one sensor would be close to the tires to measure the pavement response at a high SNR.

Data Collection. Two researchers collected data: Researcher 1 at the weigh station and Researcher 2 on the side of the highway, near the AP. Researcher 1 noted the vehicle description, its number of axles, individual axle loads, and signalled Researcher 2 via a cellular phone about the upcoming test vehicle. Researcher 2 then triggered the AP to start logging the vibration and detection data from all sensors. Data for 53 random trucks, ranging from pickup trucks to 5-axle commercial trucks, was collected. Unfortunately, due to high traffic at the weigh station, each vehicle was only slowed down for weighing, as opposed to stopping, and its axle weights were eye-balled from an analog meter. This makes the ground truth readings for axle loads less reliable.

Data Analysis. An automatic axle detection algorithm was developed and tested using this dataset in Section 4.1. Axles from all 53 trucks were correctly counted.

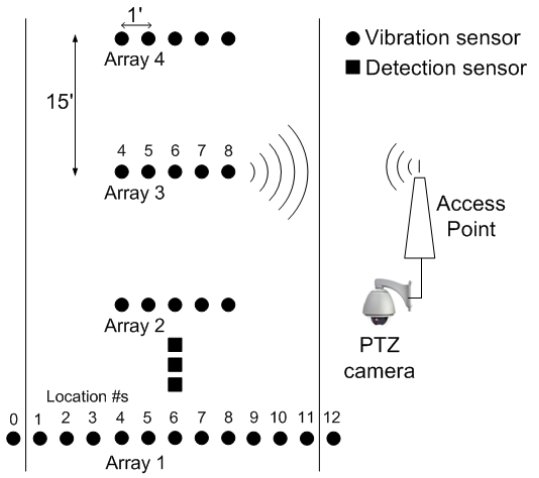
3.5 Pinole

(a) (b)

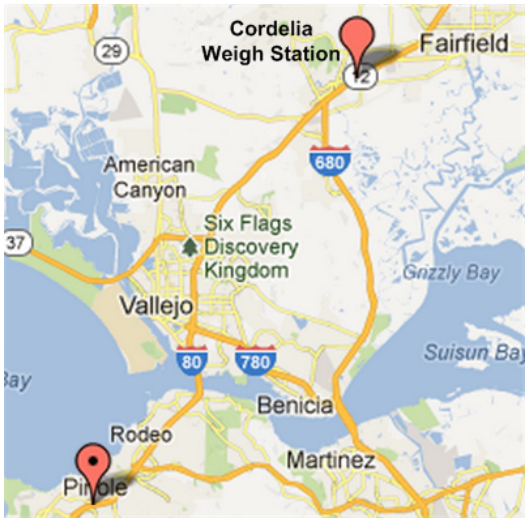
Figure 3.7: Pinole site. (a) Sensors installed in lane 2, location outlined in red. The AP and camera are installed on the road-side pole. A conventional WIM station is located right after the overpass. (b) View of the WIM station from the overpass.

For systematic long-term testing, we installed the wireless WIM system near a conventional WIM station on California highway I-80 W. The sensors were installed in lane 2 of the highway, as shown in Figure 3.7(a). The AP and the PTZ camera were mounted on a roadside pole. The conventional WIM station is located about 300 ft downstream of our system, right after the overpass seen in the picture. Figure 3.7(b) shows a view of the Caltrans-operated WIM station from the overpass. The force sensors are outlined in red, and special concrete pavement has been built around them to reduce error. Vibration and detection data from our system is used for axle detection and load estimation, whereas the data reported by conventional WIM can be used as ground truth for calibrating and testing our system.

Data from the previous sites lead to a better understanding of pavement-vehicle interaction and provided valuable insights for the sensor layout at this site. Figure 3.8(a) shows the detailed sensor layout. There are 4 arrays of vibrations sensors at a distance of 15 ft from each other. Each array contains 5 sensors in the middle that are used for load estimation. Array 1 has additional sensors that cover the entire lane and can be used for estimating wheel locations along the array. The edge sensors can also be used to study the vibrations caused by vehicles in neighboring lanes.



(a)



(b)

Figure 3.8: (a) Sensor layout for Pinole. The installation location is shown in Figure 3.7(a). (b) Google map showing Cordelia weigh station and the Pinole site. The weigh station is 21 miles upstream from the wireless WIM.

3.5.1 Experiments

Calibration truck runs. We hired a 5-axle truck loaded with lumber, and ran it over the system multiple times on Feb 14, 2012. Table 3.4 lists the different truck speeds used

and the number of repetitions at each speed. In addition to this, data was recorded for 525 random trucks during the same time period. The goal was to use the hired truck for system calibration, and compare load estimates of random trucks with the ground truth reported by WIM station. While the conventional WIM reported accurate vehicle classification data, the weight data reported was highly inaccurate and could not be used as ground truth (see Section 6.2.4).

Table 3.4: Summary of calibration truck runs.

Speed	Number of runs
15	4
35	4
55	4
65	4

Table 3.5: Summary of data collected from static weigh station.

Date (2012)	Trucks weighed	Trucks reaching Pinole	Trucks matched
May 29	47	29	18
July 3	60	32	18
Sept 10	50	37	25

Statically weighed trucks. To obtain accurate weight data, we used a static weigh station about 21 miles upstream from our sensors, as seen in Figure 3.8(b). Random trucks were stopped and weighed at the Cordelia weigh station on three separate days. Pictures of each truck were taken and matched with the road-side camera images in Pinole to extract the sensor data corresponding to these trucks. Since the station is far from our site, some of the trucks took alternative routes and only a subset of trucks that reached Pinole travelled in the sensor lane. Table 3.5 shows the number of trucks weighed on each day, the number of trucks that actually reached Pinole, and the number of trucks that were matched in the correct lane. Vibration data, pavement temperature, detection data and ground truth weights for 61 trucks were obtained this way.

Data extraction details. In order to extract sensor data for a specific truck, we need to know the time the truck arrives at the system. For the trucks weighed at cordelia, we compare the truck images with the pictures from the road-side camera. Once a visual match is made, we record the timestamps reported by the PTZ camera. The detection data reported around this time can be used to estimate the vehicle speed. Using the arrival time, departure time, and the estimated speed, a time window for the vehicle's presence on each array is calculated. The vibration and temperature data reported during these time windows is extracted and saved for further processing.

Similarly, based on the vehicle speed and distance to the convention WIM station, the truck's arrival time at the WIM is estimated. Using this time, and by comparing the vehicle length, a truck match in WIM data is found and corresponding data is extracted.

Data Collection. Combining the 16 calibration truck runs with 61 statically weighed trucks, we obtain sensor data for 75 trucks with reliable static weights. Corresponding data from the conventional WIM station is also obtained for comparison. This data includes

vehicle speed, length, classification, axle spacing, individual axle weights, and gross weight for each truck. However, data from WIM station was not available for May 29^{th}, leaving a total of 59 trucks for comparison[1]. In addition to the 75 trucks, sensor data for 525 random trucks is available for comparison of axle count and axle spacing with the conventional WIM.

Data Analysis. We develop and test an automatic vehicle classification system using the 525 random trucks in Section 4.2. Axles of all vehicles were counted correctly and the error in axle spacings was found to be within $\pm$ 0.85 ft at a 95% confidence level. A model for pavement-vehicle interaction and load estimation is developed and tested in Chapter 5. The system passed the LTPP accuracy specification for individual axle loads and the gross weight of a truck. In Chapter 7, we present an algorithm that uses the above tested model to estimate pavement displacement corresponding to every truck.

[1]Caltrans could not provide any data for this day due to technical difficulties.

Chapter 4

Automatic Vehicle Classification

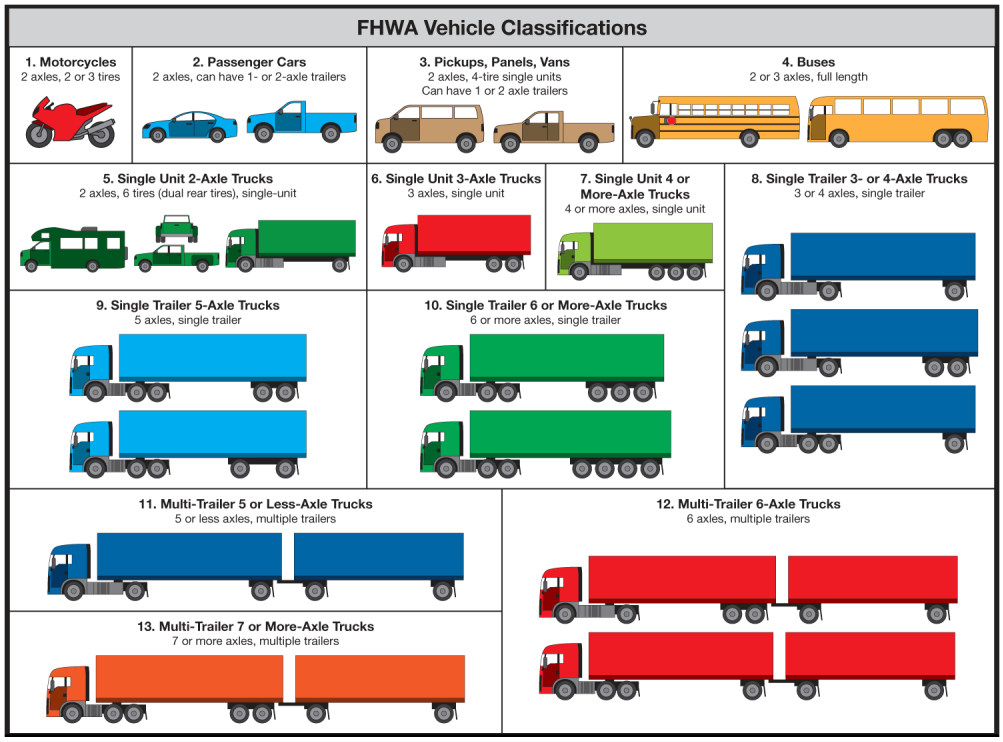


Figure 4.1: FHWA vehicle classification chart [48]. Vehicles can be classified into the 13 categories using their axle count and inter-axle distances (or *axle spacings*).

An automatic vehicle classification (AVC) system detects an on-going vehicle and categorizes it based on its size, shape, and other attributes. Typical categories include passenger cars, buses, and trucks of different sizes. Different states use different classification schemes but majority of them are based on the Federal Highway Administration's (FHWA) recom-

mended 13-class scheme shown in Figure 4.1. In 1985, Maine Department of Transportation developed an algorithm called Scheme "F" that could accurately classify vehicles based on their axle count and the inter-axle distances (axle spacings) [33]. The problem of vehicle classification is, therefore, equivalent to estimating the *axle count* and *axle spacings* of a vehicle. Typical scheme "F" based AVC systems have two important components: a mechanism for vehicle detection and a mechanism for axle detection. The general setup for our AVC system is shown in Figure 2.1. Wireless detection sensors are used to automatically detect an on-going vehicle and ground vibrations measured by vibration sensors are used for axle detection. In this chapter, we discuss in detail how axles are automatically detected from pavement vibrations and how the axle spacings are estimated. We present two scenarios for the problem: axle detection using a *sparse* sensor array, and axle detection using a *dense* sensor array. The following terminology is useful for this chapter:

- **Dense sensor array.** A sensor array having density greater than 1 sensor/ft across the width of the traffic lane.
- **Sparse sensor array.** A sensor array with density less than 1 sensor/ft across the width of the traffic lane.
- **Tire-on-top sensor.** If any tire of a vehicle goes directly on top of a sensor, the sensor is called a tire-on-top sensor for that vehicle.
- **Non-tire-on-top sensor.** If no tire of a vehicle goes directly on top of a sensor, the sensor is called a non-tire-on-top sensor for that vehicle.

4.1 Axle detection using a sparse sensor array

We start this section by providing examples of measured pavement acceleration and the motivation behind the axle detection algorithm. We then present the axle detection algorithm using a single sensor and using multiple sensors, and end the section with results for the Sunol dataset. The sensor setup and data collection procedure are described in Section 3.4.

4.1.1 Data examples

Figure 4.2 shows the measured pavement response for two 5-axle trucks (FHWA class 9) at different speeds. The axles can be visually spotted as each axle causes a separate burst of energy in the measured response. The force applied by each axle causes the pavement and sensors to vibrate such that the measured response resembles an exponentially decaying sinusoidal. An important observation is that the duration of energy burst due to each axle depends on vehicle speed. A slow moving axle causes lower-amplitude but longer-lasting pavement vibrations compared to a fast moving axle. The signal bandwidth, in other words, is directly proportional to vehicle speed.

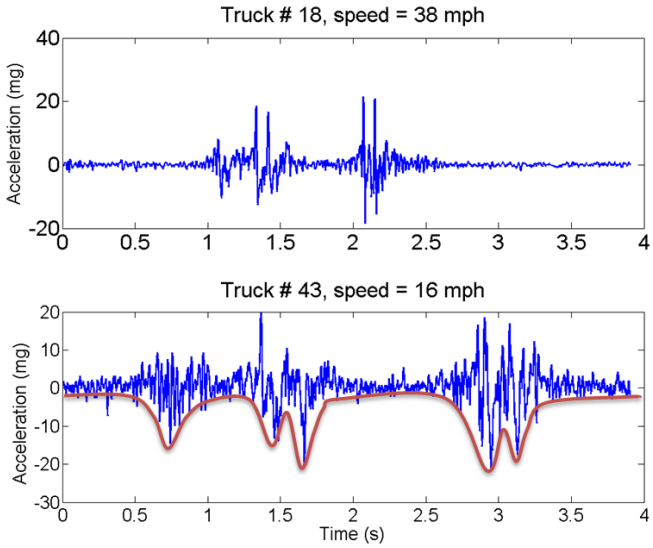


Figure 4.2: Measured pavement response for two class-9 trucks at different speeds. The red curve roughly tracks the energy bursts due to each axle.

4.1.2 Algorithm using a single sensor

Figure 4.3 shows the block diagram of the automatic axle detection algorithm using a single vibration sensor. The algorithm finds the smooth power signal from measured acceleration, and locates the peaks that are sufficiently separated in time. The raw signal a(n) is normalized such that in the absence of an axle the magnitude of the signal is below 1. The normalized signal is then squared to calculate the power p(n). This step further increases the discrepancy between the peak signal and the noise floor since noise is attenuated after squaring whereas the peak signal, which is greater than 1, is amplified.

The next step smoothes p(n) by passing it through a moving average filter (MAF) with M taps. The output of this stage is equal to the average power in last M samples and therefore proportional to the total energy in past M samples. Since the duration of the energy burst due to each axle is proportional to vehicle speed, M should also be speed dependent. We empirically defined $M(v) = \text{round}(900/v)$ by observing p(n) for four trucks at different speeds. An additional stage of a single-pole low-pass Butterworth filter (cutoff 5 Hz) is used for further smoothing. This step is optional and we observed similar results even in its absence.

The peaks in s(n) corresponding to each axle are assumed to represent axle locations in the measured response. The last step finds peaks in s(n) that are greater than a threshold ($\text{P}_{\text{th}} = 10$) and have a minimum time separation $\zeta(\text{v})$. This step ensures that local peaks in noise and local variations around peaks of s(n) are not detected as axles. Weigh stations

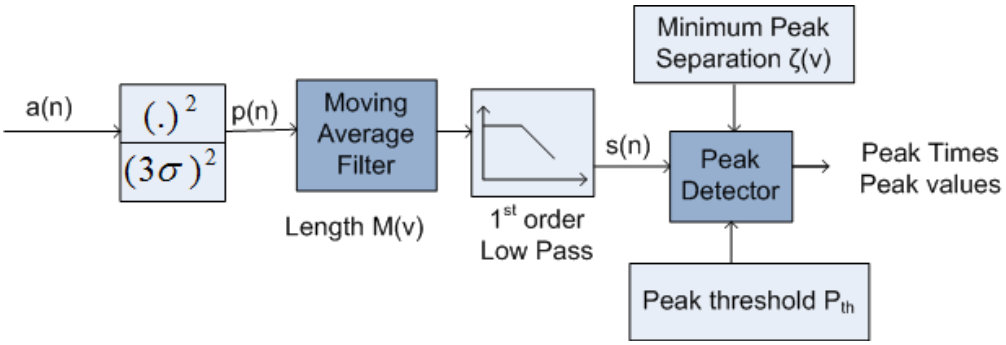


Figure 4.3: Automatic axle detection algorithm using measured response from a single vibration sensor.

typically consider tandem axles as one axle and therefore ground truth data for Sunol dataset also counted tandem axles as one axle. Tandem axles typically have a spacing between 4 and 5 ft. Minimum peak separation $\zeta(v)$ for axles was chosen by assuming that the axles are at least 6 ft apart i.e. $\zeta(v) = \frac{6}{v}$. It is important to note that by reducing the minimum separation to less than 3 ft the algorithm is able to detect both the axles in a tandem axle, as shown in Figure 4.5.

4.1.3 Algorithm using multiple sensors

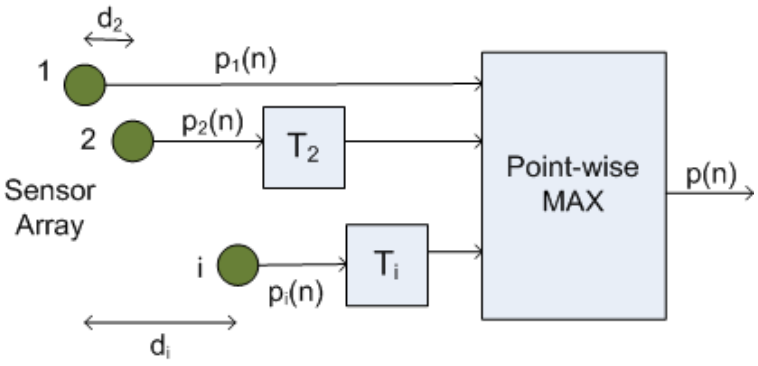


Figure 4.4: Combining data from multiple sensors for axle detection.

A single sensor is not sufficient to capture the vehicle in its entirety. If the axle detection algorithm is applied to a single sensor, the number of axles will most likely be undercounted. As the distance between a tire and sensor increases, the measured response decreases exponentially [28]. Therefore, if the sensor is sufficiently away from tires of an axle, the peak

corresponding to the axle can be below the threshold P_{th}, resulting in a missed detection. The number of missed detections will be greater for vehicles with lower speed or lower weight.

To compensate for this effect, we combine measurements from multiple sensors. Individual sensor measurements are first aligned relative to sensor 1, as shown in Figure 4.4. The time offset T_i between sensor i and sensor 1 can be computed using the vehicle speed v and the distance d_i between sensor i and sensor 1 along the direction of travel i.e. $T_i = d_i/v$. The aligned power signals are then passed through a point-wise MAX block to generate a new time series p(n). The resulting signal p(n) is equivalent to the maximum energy measured in time for each axle. This signal can be passed through the filters and the peak detection stage of the axle detection algorithm as described before.

Let t_k be the detection time reported by the algorithm for the k^{th} axle, and let v be the estimated vehicle speed from the detection sensor data (see Section 2.1.1). The axle spacing s_k between axles k and $k+1$ can be determined using

$$s_k = v(t_{k+1} - t_k) = v(\triangle T_k). \tag{4.1}$$

4.1.4 Results

To illustrate the algorithm, we use the acceleration data collected for a class-8 truck from a single vibration sensor. Figure 4.5 shows the measured signal a(n), the scaled power p(n), and the smooth power s(n). The increased discrepancy between the peaks and the noise floor can be seen by comparing a(n) and p(n). The detected axles are shown as red asterisks on s(n). Note that only one of the last two peaks in s(n) is detected, thus counting the tandem axle as only one axle. By reducing the minimum axle separation to 3 ft the algorithm successfully detects both axles of the tandem axle. It is worth mentioning that the detected peaks in s(n) do not align with the peaks in a(n) because of the delay introduced by the filters. However, the time difference between the peaks $\triangle T_k$ is unaffected.

Axle Count. Table 4.1 compares the axle count results when the algorithm is applied to each sensor individually and to the combined measurements. *Axle count error* is defined as the difference between the estimated axle count and the ground truth axle count. The worst-case axle count error is -3 when sensor I (Figure 3.6) undercounts 3 axles of a truck. By combining the measurements from all sensors, the algorithm gives the correct axle count for all 53 trucks.

Axle Spacing. Figure 4.6 shows the distribution of estimated axle spacings. Since trucks at the weigh station could not be stopped for measurements, the ground truth data for axle spacings is not available for this dataset. Instead, we provide a qualitative assessment of the results. The data appears to be naturally clustered into three different groups separated by empty bins in the histogram. The first cluster includes axles that are spaced between 3 ft and

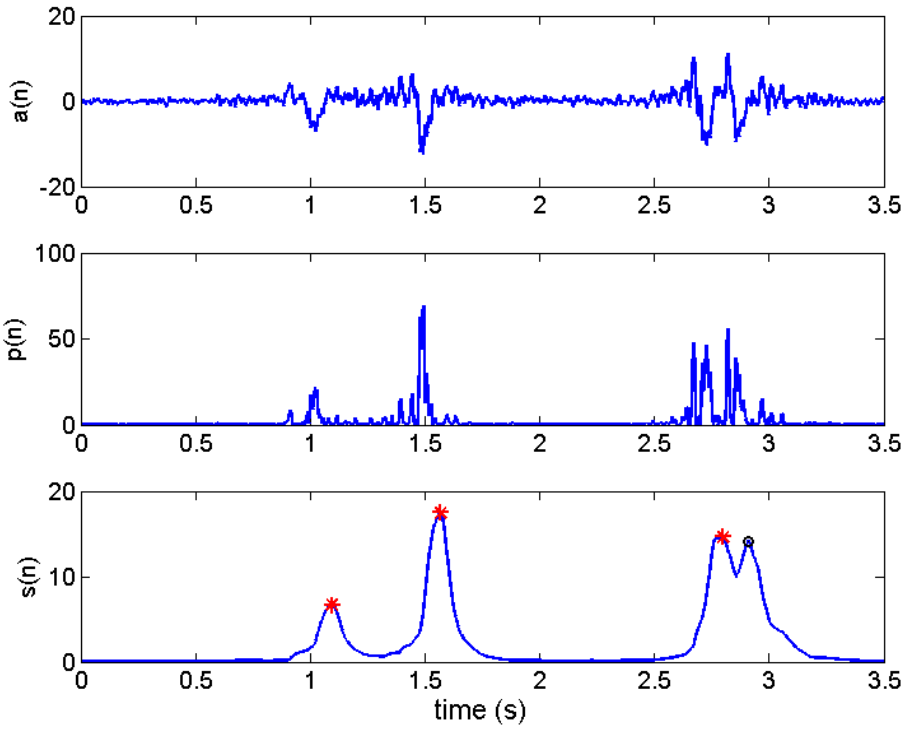


Figure 4.5: Example application of the axle detection algorithm. The axle denoted by the black circle is detected when $\zeta(\mathrm{v}) = \frac{3\ (ft)}{\mathrm{v}\ (\mathrm{ft/s})}$ is used.

6 ft. This is very typical for tandem axles and provides encouraging evidence supporting the accuracy of the estimates. The second cluster is mostly accounted by pickup trucks, small two-axle commercial trucks, and the first two axles of larger commercial trucks. The third cluster is mostly comprised of axles of the trailers. The large variation in the second and third cluster is expected and is consistent with Federal Highway Administration's data [5].

4.2 Axle detection using a dense sensor array

The algorithm described in the previous section has a few drawbacks. The measured response is lower in amplitude for lighter and lower speed vehicles, resulting in possible missed detections. A vehicle classification system, on the contrary, is expected to work for all speeds and weights. Another drawback is that the number of filter taps M and the minimum peak separation ζ are speed dependent but vehicle speed is unknown to the vibration sensors. One possible solution is to send average traffic speed to the sensors using *sync* packets but we will present an alternative in this section that makes the algorithm speed independent. The

Count Error	Sensor I	Sensor J	Sensor L	Combined
-3	1	0	0	0
-2	1	2	1	0
-1	2	2	3	0
1	3	1	1	0
Correct	46	48	48	53
Performance	86.8%	90.6%	90.6%	100%

Table 4.1: Axle count performance for the algorithm. *Count error* is the difference between the estimated axle count and the ground truth. Under each column is the observed frequency of the errors.

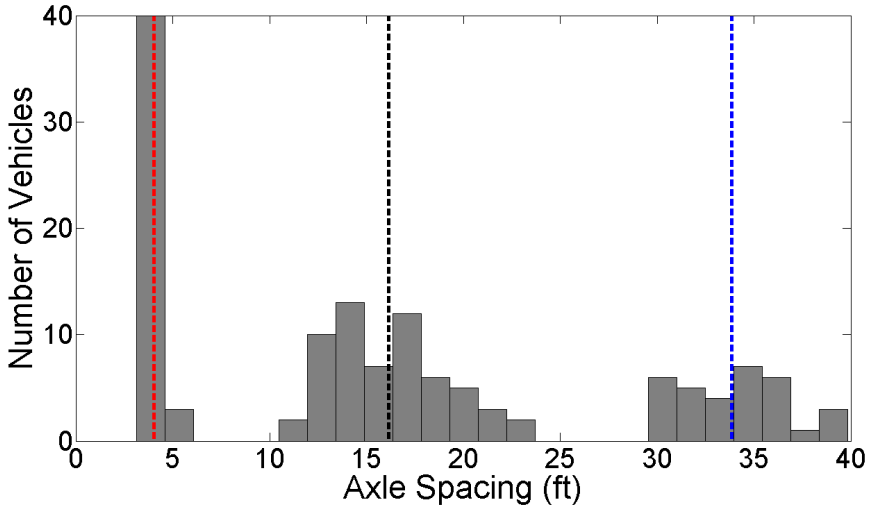


Figure 4.6: Distribution of estimated axle spacings.

biggest drawback is the transmission of raw acceleration data from sensors to access point. Transmitting data over the radio is the most power consuming process for a wireless sensor and the amount of data transmitted should be minimized [34]. We present an algorithm that addresses all these limitations.

4.2.1 Tire-on-top effect

Figure 4.7 compares the measured response by two sensors for the same truck. The top plot is for a tire-on-top sensor whereas the bottom plot is for a non-tire-on-top sensor. It was observed that the tire-on-top sensor records higher amplitudes for all axles, independent of the axle weight. The response is also less dependent on speed when compared to the non-tire-on-top sensors. This observation makes the tire-on-top sensors very suitable for axle

detection but very unsuitable for load estimation. Motivated by this effect, we installed a special sensor array in our Pinole setup (Section 3.5). As shown in Figure 3.8(a), Array 1 has a higher number of sensors covering the width of the lane. The goal is to ensure that at least one of the sensors will be a tire-on-top sensor for every vehicle and can be used for axle detection.

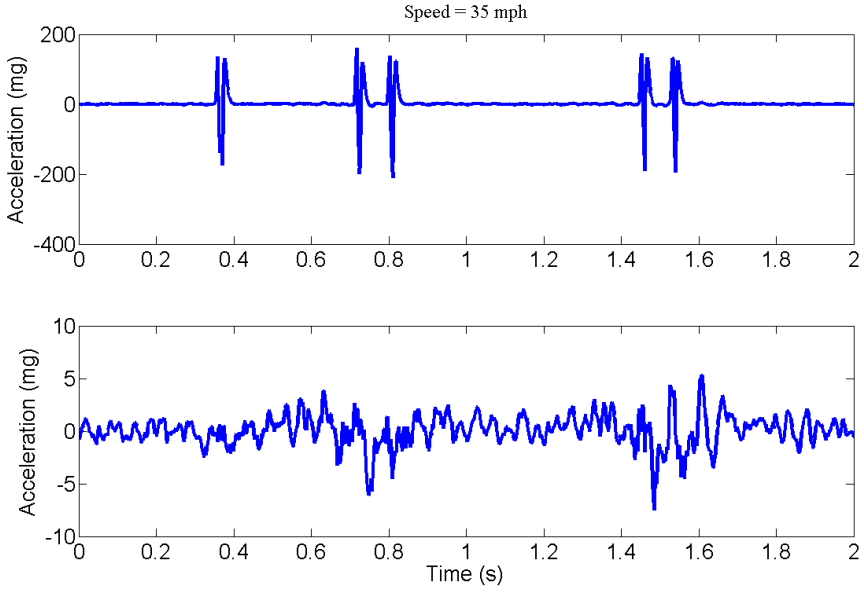


Figure 4.7: Comparison of pavement response measured by a tire-on-top sensor and a non-tire-on-top sensor.

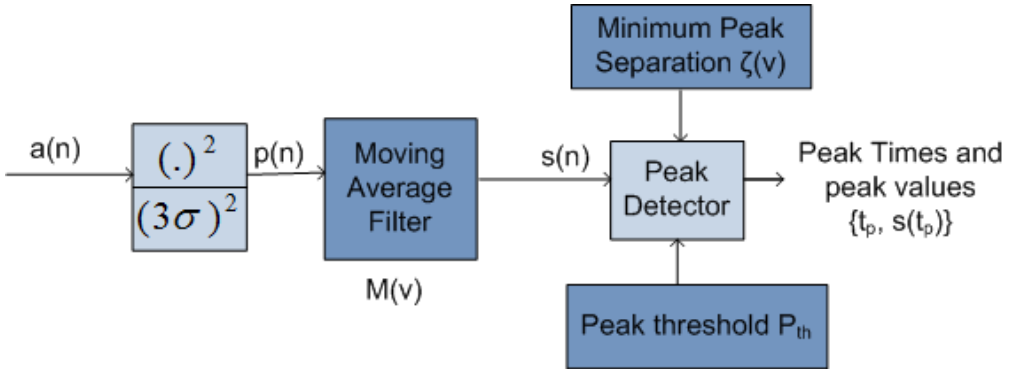


Figure 4.8: Sensor-level axle detection algorithm using a dense sensor array.

4.2.2 Sensor-level algorithm

Each sensor decides whether it is a tire-on-top or non-tire-on-top sensor by thresholding the amplitude of the measured signal. If the amplitude is greater than 25 mg, the sensor considers itself a tire-on-top sensor and applies the modified axle detection algorithm shown in Figure 4.8. The algorithm is very similar to the one described in Section 4.1.2 but with a few key differences. The moving average filter is now the only filter used and the number of taps are chosen to be independent of speed, i.e. $M(v) = 12$. The amplitudes from tire-on-top sensors are much higher and therefore, the peak detection threshold is increased ten folds to $P_{th} = 100$. Since vehicle speed is not known to the vibration sensor, we calculate the minimum peak separation using an upper bound on expected vehicle speed, i.e. $\zeta(v) = \frac{2.5\ (ft)}{132\ (ft/s)}$. This results in extra peaks reported by sensors for lower vehicle speeds but the access point can reinforce the separation constraint using the actual vehicle speed and filter out the redundant peaks. Since each sensor only sends out axle times $\{t_k\}$ and peaks $\{s(t_k)\}$, the amount of data transmitted is minimized and battery power is used more efficiently.

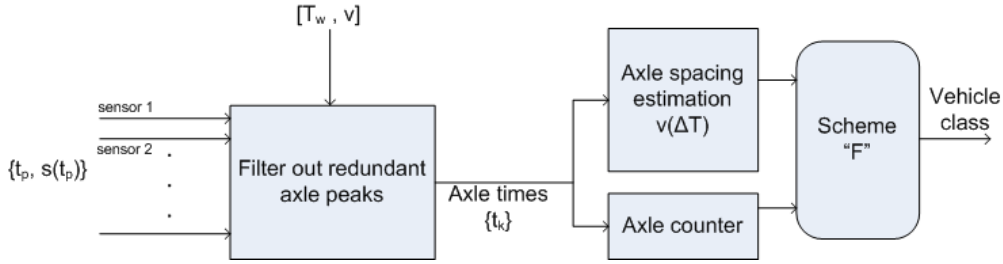


Figure 4.9: AP-level algorithm using a dense sensor array.

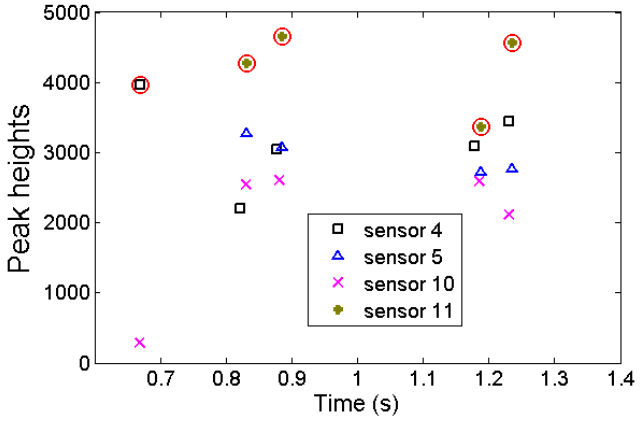


Figure 4.10: Example AP-level filtering for axle detection.

4.2.3 AP-level algorithm

The access point estimates the vehicle speed v from the detection sensor data using the procedure described in Section 2.1.1. Based on the speed, it calculates the minimum peak separation as $\zeta(v) = \frac{2.5\ (ft)}{v\ (ft/s)}$ and reapplies the constraint to all peaks reported by sensors. The AP filters out redundant peaks and selects the highest peaks that are at least $\zeta(v)$ time apart. This is illustrated in Figure 4.10. The AP filters out redundant peaks reported by four tire-on-top sensors and the detected axles are circled in red. The resulting axle times $\{t_k\}$ can be combined with estimated speed v to calculate the axle spacings using Equation 4.1. The estimated axle count and axle spacings can be used to classify the vehicle using scheme "F" [33].

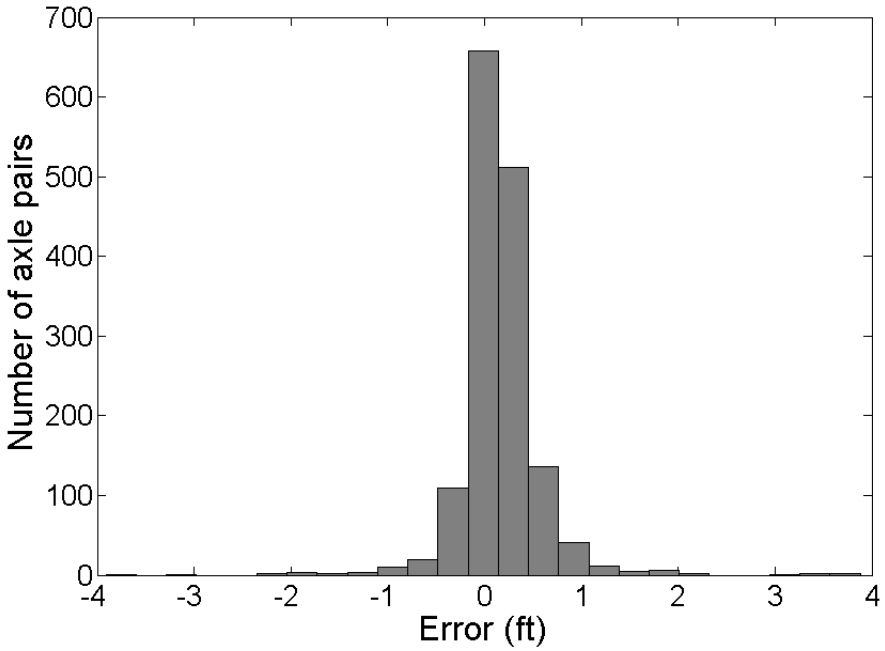


Figure 4.11: Histogram of axle spacing errors for 525 random vehicles.

4.2.4 Results

To test the algorithm, we applied the sensor-level algorithm to the raw vibration data collected for each truck. The detected axle peaks were filtered out using the AP-level algorithm and used to estimate the axle count and axle spacings for 525 random trucks. The sensor setup and data collection procedure is described in Section 3.5.1.

All axles of the 525 vehicles were successfully detected. Let the *axle spacing error* be defined as the difference between the estimated and the ground truth axle spacing. Figure 4.11 shows

the histogram of axle spacing errors for the 525 vehicles. The 95th percentile error was found to be ±0.85 ft and 97% of the estimates had an error within ±1 ft.

The above accuracy should be sufficient for classification based on Scheme "F" but it can be improved further, if needed. If we "perfectly" estimate the time difference $\triangle T_k$ between two axles, Equation 4.1 can be used to write the axle spacing error as $\epsilon_s = \epsilon_v(\triangle T_k)$, where ϵ_v is the error in the speed estimate. One convenient way to improve the axle spacing estimate is to reduce the error in the speed estimate. This can be accomplished by using more speed sensors and averaging multiple speed estimates [57]. The relationship also predicts that error ϵ_s will be greater for larger axle spacings as the corresponding $\triangle T_k$ is larger. Figure 4.12 confirms this dependency of errors on axle spacings.

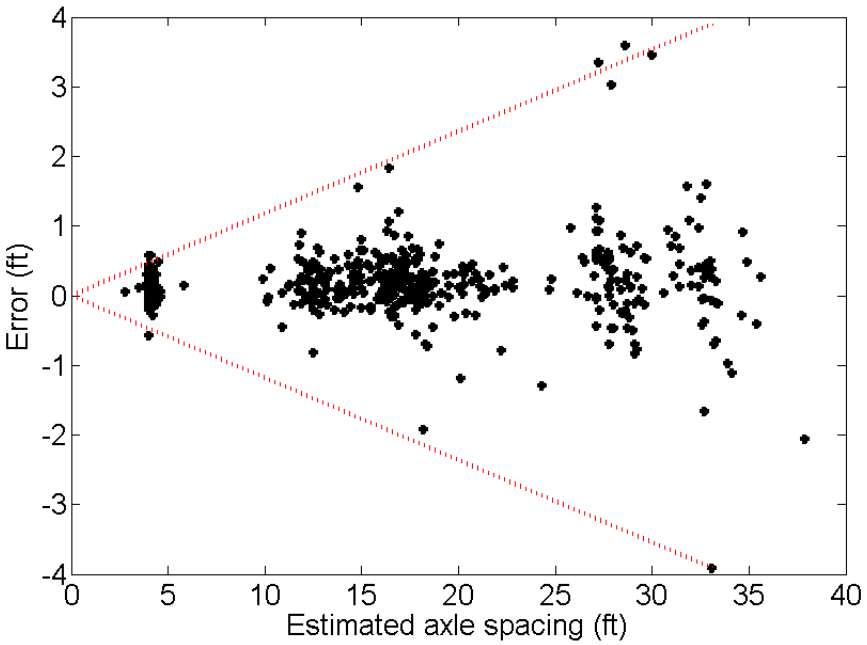


Figure 4.12: Dependency of axle spacing errors on axle spacing estimates. Only vehicles with speed between 55 and 60 mph were used for this plot.

Chapter 5

Load estimation

In this chapter, we discuss how pavement vibrations can be used to estimate truck weights. We start by presenting some exploratory data analysis for the various experiments described in Chapter 3. We then propose a model that relates pavement acceleration, pavement temperature, and vehicle speed to axle weights. The procedure for estimating axle weights is described next and we end the chapter with results for 75 random trucks from the Pinole site. The collected data and the sensor layout for this site are described in Section 3.5.1. The following terminology is useful for this and the next chapter.

- ***Peak acceleration.*** Figure 5.1(b) shows an example of the pavement response for a two-axle truck and each negative peak corresponds to a truck axle. We refer to the magnitude of each such peak as peak acceleration for the corresponding axle. The global maxima of the signal, on the other hand, is called the positive peak of acceleration.

- ***Pulse width.*** Each axle causes a pulse in the measured signal as shown in Figure 5.1(b). Pulse width is the time difference between the points on the pulse that are 10% of the peak acceleration.

- ***Dynamic load.*** The instantaneous load applied to the pavement by truck wheels ($F_I(t)$) is different from the static load (F) because of suspension oscillations of the truck. The difference between instantaneous and static load is called the dynamic load, F_d, where $F_d(t) = F_I(t) - F$.

- ***Relative standard deviation (RSD).*** The standard deviation of measurements expressed as the percentage of the mean of measurements.

- ***Relative range.*** The range of measurements (under a given test condition) expressed as a percentage of the mean of measurements. We use relative range to express the statistical dispersion when the number of measurements is smaller than 10. Note that

if the measurements follow a normal distribution $N(\mu, \sigma^2)$, then the relative range of measurements is expected to be four times the RSD value at a confidence level of 95%.

- ***Relative standard error (RSE).*** Standard error is the standard deviation of the sample mean (or any other statistic). The relative standard error is simply the standard error divided by the true mean (or its estimate) and expressed as percentage. It is important to understand the difference between RSD and RSE. Qualitatively, RSD is the degree to which individual measurements differ from the sample mean whereas RSE is an estimate of how close the sample mean is to the true mean. If the sample standard deviation is $\hat{\sigma}$ for N independent measurements, the standard error of mean is $\frac{\hat{\sigma}}{\sqrt{N}}$.

- ***Coefficient of determination (R^2).*** We use the most widely accepted definition of R^2 to express the 'goodness of fit'. R^2 value ranges from 0 to 1 and higher values imply better model fits. Let $\{y_i\}$ be the observations, $\{\hat{y}_i\}$ be the modeled values, $\bar{y} = \frac{\sum_{i=1}^{N} y_i}{N}$ be the sample mean, then the R^2 value can be calculated as

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}.$$

- ***LTPP error (e_λ).*** The LTPP recommends an error measure for WIM stations that assumes a normal distribution for measurement errors and calculates the error bound at a confidence level of 95%. Let $\{e_i\}_{i=1}^{N}$ be the observed errors, $\bar{e}$ be the mean error, $\hat{\sigma_e}$ be the standard deviation of observed errors, and t_{N-1} be the critical value at 95% confidence level for a Student t-distribution with $N-1$ degrees of freedom, then e_λ can be calculated as

$$e_\lambda = |\bar{e}| + t_{N-1}\hat{\sigma_e}. \tag{5.1}$$

5.1 Observations from measured pavement response

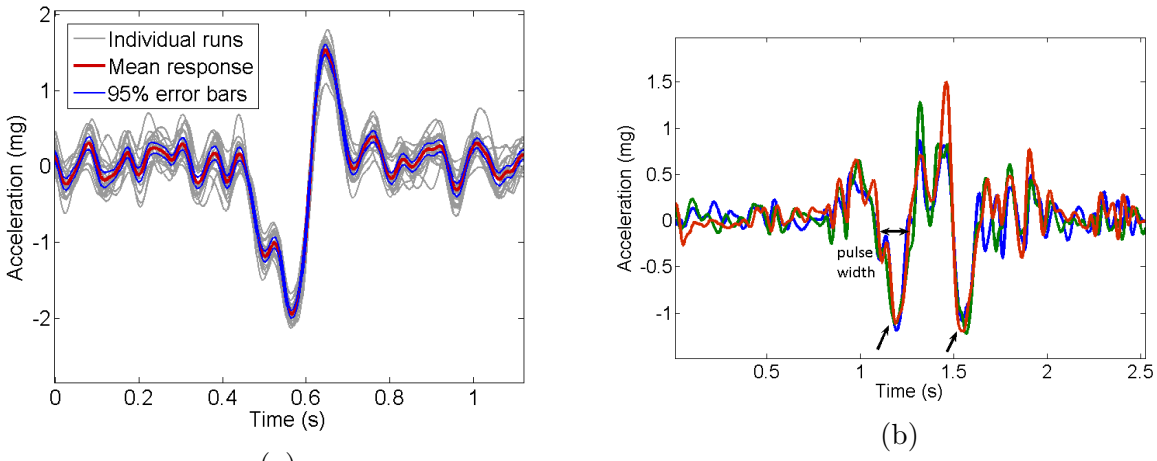
In this section, we discuss some important observations made from the experiments described in Chapter 3. The observations provide supporting evidence for the feasibility of wireless WIM and help understand the pavement response to applied loads.

5.1.1 Measurement Repeatability

A measuring device is said to have high repeatability if the variation in its measurements is small. A fundamental requirement for a WIM system is high repeatability. The relative standard deviation (RSD) in typical WIM measurements is expected to be less than 10%.

The relative range of measurements is expected to be less than 40% at a confidence level of 95%.

The vibration sensor showed high repeatability in our experiments. Figure 5.1(a) shows the measured pavement response for multiple HVS runs under similar speed and load conditions. The measurements look similar in shape and the RSD of peak acceleration was found to be 6.2% for this test. The RSD of peak acceleration ranged from 6.2% to 11% for the 12 HVS tests at 5.41 mph and better repeatability is expected at highway speeds. The HVS experiments were performed at a speed of 5.41 mph (or lower) and pavement response is minimal at such low speeds. As an extreme example, a static vehicle causes no pavement vibrations; therefore, it cannot be weighed using the wireless WIM. Similarly, slow moving vehicles cause pavement vibrations that are barely above the noise floor of the vibration sensor, resulting in a lower repeatability of measurements. It is very encouraging to get reasonably high repeatability for such low speeds.



(a) (b)

Figure 5.1: Measurement repeatability. (a) Measured response for 20 runs of the HVS test at a speed of 5.41 mph and load of 60 kN. The average sensor (or pavement) response is shown in red and shown in blue are the upper and lower limits of its confidence interval at 95% confidence level. (b) Measured response for 3 truck runs at 20 mph. The end points for the pulse width are 10% in value of the peak acceleration.

Figure 5.1(b) shows the response of a concrete pavement for multiple runs of a two-axle truck. Qualitatively, the response looks very similar in each case. Quantitatively, as shown in Figure 5.3(a), the peak acceleration is always within 15% of the mean value at different speeds; therefore, relative range of peak acceleration is less than 30% for each speed. This level of repeatability is acceptable and one way to improve it is to increase the number of vibration sensors and aggregate the measurements from them. However, it is important to note that some of the measurement variation is due to the variation in test conditions: the

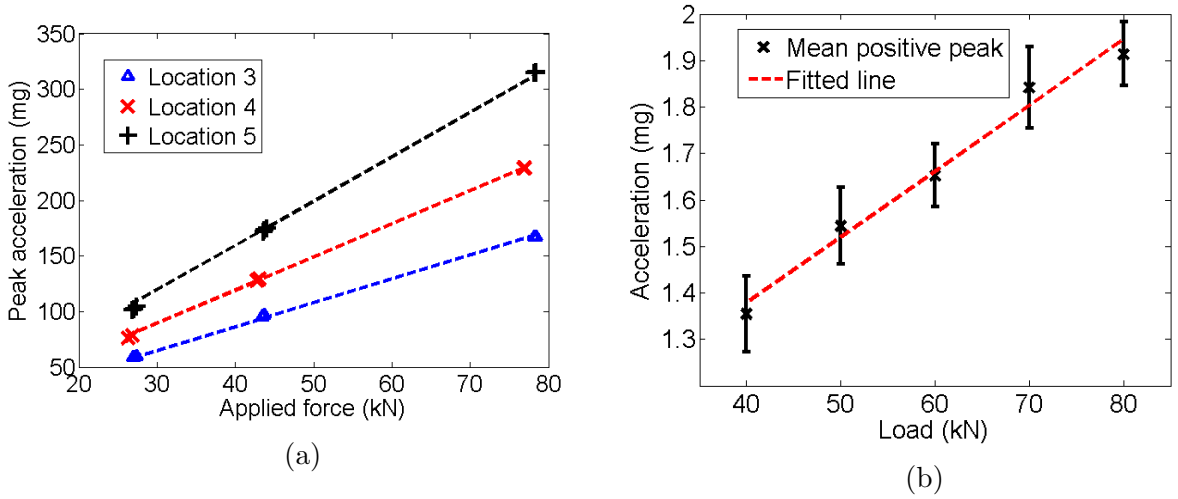
instantaneous load applied by truck axles is different in each run due to suspension oscillations (*dynamic load effect*); speeds can be slightly different in each run; and the location of truck wheels relative to the sensor location varies in each run.

For the FWD experiment, we applied 3 different loads at 6 locations near the sensor. All 18 tests were repeated 3 times each. The relative range of peak acceleration was found to be between 0.1% and 3.4%, thus, showing very high repeatability.

Observation 5.1. *The measured pavement response has high repeatability under similar test conditions.*

5.1.2 Effect of applied load on pavement response

The Euler-Bernoulli beam model for pavement-vehicle interaction suggests that the measured response is linearly proportional to the applied load at a given vehicle speed [52]. This was observed in our experiments for non-tire-on-top sensors only. For tire-on-top sensors, the measured response was found to be uncorrelated to the applied load, thus, making these sensors unsuitable for load estimation.



(a) (b)

Figure 5.2: (a) Pavement response at different load levels for the FWD experiment. (b) The mean of positive peak acceleration from the 20 runs plotted against the applied load in each run. The error bars correspond to the 95% confidence interval for the estimate.

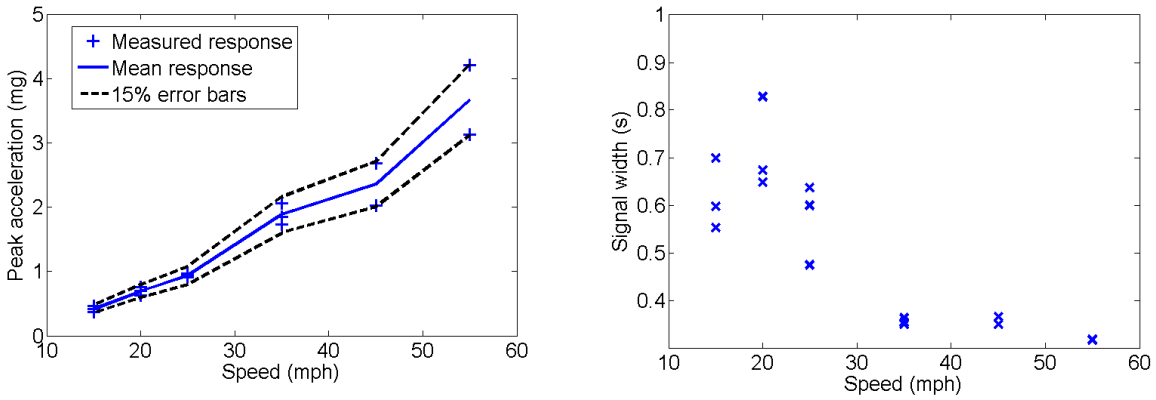
Figure 5.2(a) shows a plot of peak acceleration against the load applied by the FWD. The 3 linear fits shown in the plot have R^2 values of greater than 0.99. Signal features other than peak acceleration can also be used for load estimation. Figure 5.2(b) shows the mean of the positive peak acceleration plotted against the applied load. The mean of the positive peaks

is obtained by averaging the individual positive peaks from the 20 HVS runs at each load. The linear fit for the HVS experiment has a R^2 value of 0.98 and provides more evidence to support the following observation.

Observation 5.2. *At a given vehicle speed, measured pavement response is linearly proportional to the applied load.*

5.1.3 Effect of vehicle speed on pavement response

The Euler beam model also suggests that both the shape and amplitude of the pavement acceleration depends on vehicle speed. However, the relationship between speed and measured response is complex and depends on the material and structural properties of the pavement [52]. Figure 5.3(a) shows the peak acceleration for multiple truck runs at various speeds. The peak acceleration increases monotonically with speed and approximately follows the square law. Figure 5.3(b) shows the pulse width of the measured response at different speeds. The pulse width decreases as vehicle speed increases and vice versa. The presence of the term 'vt' in the Equation (5.2) also suggests that the pulse width is inversely proportional to truck speed.



(a) (b)

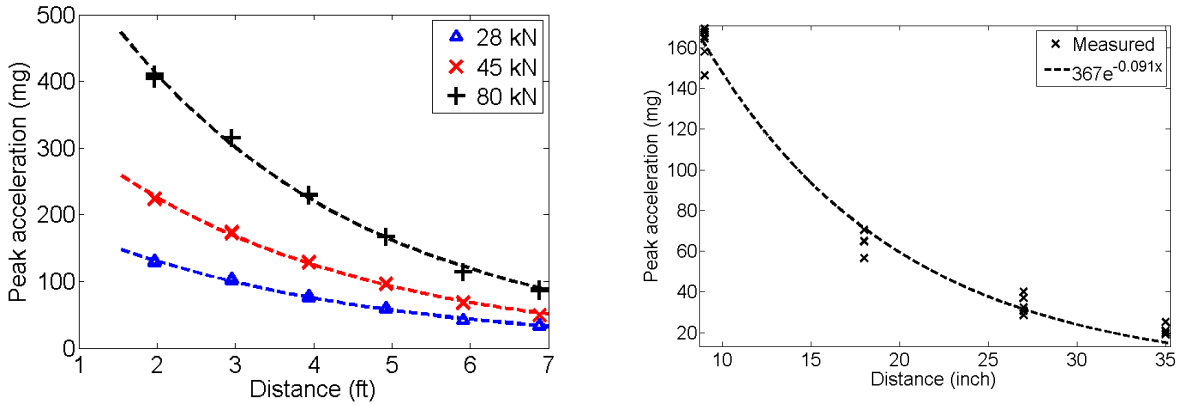
Figure 5.3: (a) Peak acceleration at different speeds for the Yolo County truck. (b) Pulse width of measured acceleration at different speeds for Axle 1 of the same truck.

Observation 5.3. *The measured pavement acceleration increases monotonically with truck speed. The pulse width of the measured response due to each axle is inversely proportional to truck speed.*

5.1.4 Effect of load location on measured response

The Euler beam model is a one-dimensional model in space and does not provide any insights into the effect of wheel (or force) location on the measured response. We use data from the FWD and the parking lot experiments to study this effect. Similar loads are applied at different locations relative to the vibration sensor and the measured response is analyzed.

Figure 5.4(a) shows the peak acceleration measured when loads of 28, 45, and 80 kN were applied at six different locations (setup in Figure 3.1). Each load was repeated three times. The plot also shows exponential fits of the form ae^{-bx}, where x is the distance from the sensor and a, b are the unknown parameters. The decay rates b estimated for the 28 kN, 45 kN, and 80 kN cases are 0.28, 0.30, and 0.31 ft^{-1}respectively.



(a) (b)

Figure 5.4: (a) Peak acceleration at different locations and load levels for the FWD experiment. The dotted curves show exponential fits to the data. (b) Peak acceleration at different locations near the sensor for the parking lot experiment.

Figure 5.4(b) shows a similar plot for the parking lot experiment using the mini-FWD (see Section 3.3). The measured response is less repeatable but it should be noted that the loads applied by the mini-FWD are not expected to be as repeatable as the commercial FWD. The estimated decay rate for the parking lot pavement was found to be 1.09 ft^{-1}. The decay constant for the asphalt pavement in the parking lot is about 3 times greater than the concrete pavement in Yolo County.

Observation 5.4. *Measured pavement response decreases exponentially as the distance between the sensor and the location of applied force increases. The decay rate of attenuation depends on the pavement type.*

5.1.5 Measured pavement response for a moving axle

In order to model the pavement response due a moving axle, we analyzed the sensor measurements for trucks from Yolo County, Sunol, and Pinole. We found that after some low-pass filtering the measured response due to an axle resembles an inverted Mexican hat function i.e. $f(t) = A(1 - \frac{t^2}{\sigma^2})e^{\frac{-t^2}{2\sigma^2}}$, where A is a negative real number. Figure 5.8 shows an example of measured response from the Pinole site. The response due to the last axle is isolated from the other axles and can be considered as the response due to an individual axle. The responses from the other sites were very similar. A slight adjustment to the low pass filter was required but this was expected since the measured response depends on pavement type. This observation provided motivation for the development of a simpler pavement-vehicle interaction model.

Observation 5.5. *The measured acceleration response due to a single truck axle resembles the Mexican-hat function i.e. second derivative of the Gaussian function. Therefore, the corresponding pavement displacement can be modeled as a Gaussian function.*

5.2 Proposed pavement-vehicle interaction model

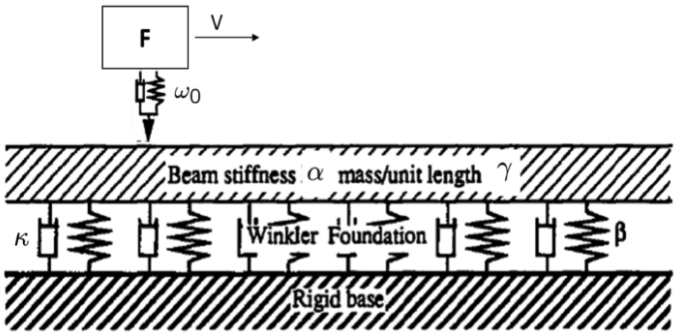


Figure 5.5: Euler-Bernoulli beam model for pavement-vehicle interaction [52].

In this section, we propose a model for pavement-vehicle interaction that directly relates pavement acceleration, vehicle speed, and pavement temperature to applied axle load. We start by describing the model at a constant temperature and then explain how measurements can be properly compensated for temperature variation.

The simplest but widely accepted model for pavement-vehicle interaction is shown in Figure 5.5. In this model, the roadway pavement is considered as a one-dimensional beam resting on an elastic foundation, and an axle is modeled as a moving force whose magnitude changes with time because of the suspension oscillations. A closed form solution to

the partial differential equation (PDE) governing the system was found in [52] and we now make some simplifying assumptions based on our experimental observations. If we ignore the effect of vehicle's suspension system (i.e. setting $\omega_0 = 0$), we can show that the solution found in [52] can be written as

$$y(t) = F\gamma^{-1}\Phi(vt), \tag{5.2}$$

where $y(t)$ is the vertical displacement or deflection of the pavement, and the function $\Phi(\cdot)$ and γ depend on the structural and material properties of the pavement. It can be seen from this expression that $\gamma^{-1}\Phi(t)$ can be interpreted as the pavement response due to a unit load, moving at unit speed. Consistent with Observation 5.2 and Observation 5.3, the model is linear in F and vehicle speed v just scales the function $\Phi(\cdot)$ in time.

Based on Observation 5.5, we now assume that the shape of pavement displacement is closely approximated by a Gaussian function. This is a simplifying assumption and in general the shape of the response also depends on v and suspension frequencies of the vehicle [52]. Let η be the amplitude and σ_0 be the standard deviation of Gaussian response for a unit load at a unit speed, then pavement displacement $y(t)$ can be written as:

$$\begin{aligned} y(t) &= F\gamma^{-1}\Phi(vt) = F\eta e^{\frac{-v^2t^2}{2\sigma_0^2}}, \\ y(t) &= F\eta e^{\frac{-t^2}{2\sigma^2}}. \end{aligned} \tag{5.3}$$

The last step is obtained by assuming $\sigma = \frac{\sigma_0}{v}$. Note that both η and σ_0 depend on pavement properties. Since we measure acceleration and not displacement, we can convert the model into a more appropriate form,

$$\begin{aligned} a(t) = \ddot{y}(t) &= F\gamma^{-1}\ddot{\Phi}(vt), \\ &= -F\eta\frac{v^2}{\sigma_0^2}\left(1 - \frac{t^2}{\sigma^2}\right)e^{\frac{-t^2}{2\sigma^2}}. \end{aligned}$$

Now, let $\Psi(t,\sigma) = -\left(1 - \frac{t^2}{\sigma^2}\right)e^{\frac{-t^2}{2\sigma^2}}$ and $\alpha = \frac{F\eta v^2}{\sigma_0^2}$, and we have the following relation for pavement acceleration due to a single axle load:

$$a(t) = \alpha\Psi(t,\sigma).$$

From the definition of α, we see that

$$\begin{aligned} F &= \frac{\sigma_0^2}{\eta}\frac{\alpha}{v^2} = \beta\frac{\alpha}{v^2}, \\ y(t) &= \alpha\frac{\sigma_0^2}{v^2}e^{\frac{-t^2}{2\sigma^2}} = \alpha\sigma^2 e^{\frac{-t^2}{2\sigma^2}}. \end{aligned} \tag{5.4}$$

The last step is obtained by combining Equations (5.4) and (5.3). The unknowns α and σ can be estimated from the measured acceleration but β depends on axle type (single or tandem) and pavement properties, and needs to be calibrated using trucks of known weights. For a K axle truck, with the i^{th} axle arriving at the sensor at time μ_i and applying a force F_i, the response can be written as the superposition of individual axle responses ($a_i(t)$)i.e.

$$a_i(t) = \alpha_i \Psi(t - \mu_i, \sigma_i), \tag{5.5}$$

$$\boxed{a(t) = \sum_{i=1}^{K} \alpha_i \Psi(t - \mu_i, \sigma_i).} \tag{5.6}$$

Using a non-linear curve fitting procedure, described in Section 5.3, we estimate α_i, μ_i, and σ_i for each axle. Once individual axle responses have been estimated, each axle can be treated separately to estimate quantities like individual axle loads (F_i) and pavement displacement ($y_i(t)$) due to each axle i.e.

$$F_i = \beta_i \frac{\alpha_i}{v^2},$$

$$\boxed{y_i(t) = \alpha_i \sigma_i^2 e^{\frac{-(t-\mu_i)^2}{2\sigma_i^2}}.} \tag{5.7}$$

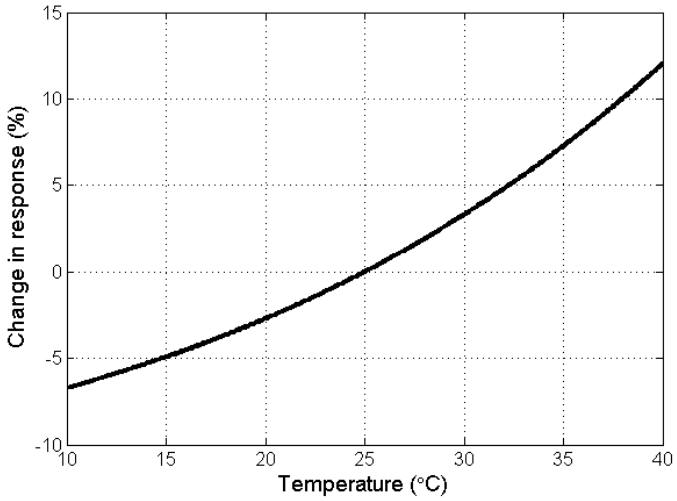


Figure 5.6: Percentage change in pavement response with temperature. For the same applied load, pavement acceleration (or displacement) increases with increasing temperature.

Temperature compensation. The above model is valid for a constant temperature but pavement response, especially for asphalt-concrete layer, is highly dependent on temperature.

Using the thickness of different layers and material parameters for the pavement at this site, we developed a layered elastic theory (LET) model to simulate the effect of temperature on the pavement response [50]. Figure 5.6 shows how the pavement acceleration changes with temperature according to the LET model. The plot shows that pavement response can change over 15% with changes in temperature alone and proper temperature compensation is needed for accurate load estimation. Appendix A contains more details on the LET model development[1].

Let $\tau(T)$ be the ratio of the modeled response at 25°C and at temperature T. To compensate for temperature, we normalize all our measurements to the reference temperature of $25°C$ as $a(t, T = 25°C) = a(t, T)\tau(T)$, where $\tau(T)$ is obtained using the LET model. It can be seen from Equation (5.6) that $\alpha_i(T = 25°C) = \alpha_i(T)\tau(T)$, and accordingly

$$\boxed{F_i = \beta_i \frac{\alpha_i}{v^2}\tau(T).} \tag{5.8}$$

5.3 Load estimation procedure

Figure 5.7 shows a block diagram for the load estimation procedure. Sensor measurements are first combined to calculate the average pavement response for the truck. A curve-fitting procedure is then used to estimate individual axle response from the average pavement response. This estimate is combined with the speed, temperature, and calibrated parameter β_i to calculate the axle weight. We discuss the calibration procedure for β_i at the end of this section.

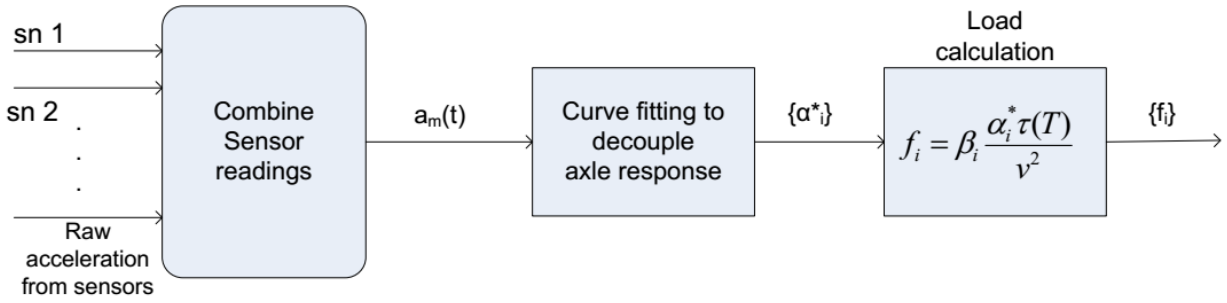


Figure 5.7: Block diagram for the load estimation procedure.

Average pavement response. The average pavement response requires aligning the measurements from each sensor. Each time-series of measurements (or each signal) is first passed

[1]Contributions of Erdem Coleri.

through a cascade of two moving average filters with numbers of taps M and $2M$ respectively. The filters are used to smooth the measured signals and are chosen to be speed dependent, $M(v) = 1440/v$. Amongst the filtered signals, the highest amplitude signal is designated as the reference and signals from other sensors are time-shifted to align with the reference. Let $a_m^k(t)$ be the time-shifted signal for the k^{th} sensor, and I be the number of available non-tire-on-top sensors, then the average pavement acceleration $a_m(t)$ can be estimated as:

$$a_m(t) = \frac{1}{I}\sum_{k=1}^{I} a_m^k(t). \tag{5.9}$$

Figure 5.8 shows an example of raw acceleration data from a sensor, and the average pavement response $a_m(t)$. The figure also highlights another important challenge in estimating individual axle loads. Because of high speed and relatively short axle spacing, Axle 2 of the truck arrives at the sensor before the pavement has relaxed from Axle 1's load. Therefore, response due to each axle needs to be decoupled and extracted from $a_m(t)$. To extract each $a_i(t)$ from $a_m(t)$, we use the following procedure.

Curve Fitting Procedure. Let $a(t)$ be the modeled response for a K-axle truck, given by Equation (5.6). Let $\epsilon(t)$ be the error between the measured and modeled response for the truck at time t i.e. $\epsilon(t) = a_m(t) - a(t)$. We can now write the measured response as:

$$a_m(t) = a(t) + \epsilon(t) = \sum_{i=1}^{K} \alpha_i \Psi(t - \mu_i, \sigma_i) + \epsilon(t). \tag{5.10}$$

We estimate the unknown parameters $\{\alpha_i\}_{i=1}^K$, $\{\sigma_i\}_{i=1}^K$, and $\{\mu_i\}_{i=1}^K$ by minimizing the mean squared error i.e.

$$\begin{aligned}(\alpha_i^*, \sigma_i^*, \mu_i^*) &= \arg\min_{\alpha_i,\sigma_i,\mu_i} \int_{-\infty}^{\infty} (a_m(t) - a(t))^2 dt, \\ (\alpha_i^*, \sigma_i^*, \mu_i^*) &= \arg\min_{\alpha_i,\sigma_i,\mu_i} \int_{-\infty}^{\infty} (a_m(t) - \sum_{i=1}^{K} \alpha_i \Psi(t - \mu_i, \sigma_i))^2 dt.\end{aligned} \tag{5.11}$$

This is a non-linear (and non-convex) least-squares problem and the solution found by standard optimization algorithms, such as the gradient descent or Gauss-Newton algorithm, is only guaranteed to be a local minima. However, global minima can be found in our case by carefully choosing the initial points for the solution search. For μ_i, we use the corresponding time reported by the tire-on-top sensors as our initial guess. For α_i, we use the corresponding peak acceleration in the measured signal and the initial guess for σ_i is made using the measured speed of the vehicle. Figure 5.8 shows an example of how well the modeled response fits the average pavement response obtained from sensor measurements.

Load calculation. Once $\{\alpha_i^*\}_{i=1}^K$, $\{\sigma_i^*\}_{i=1}^K$ and $\{\mu_i^*\}_{i=1}^K$ have been estimated, the pavement displacement due to each axle can be calculated using Equation (5.7). The individual axle

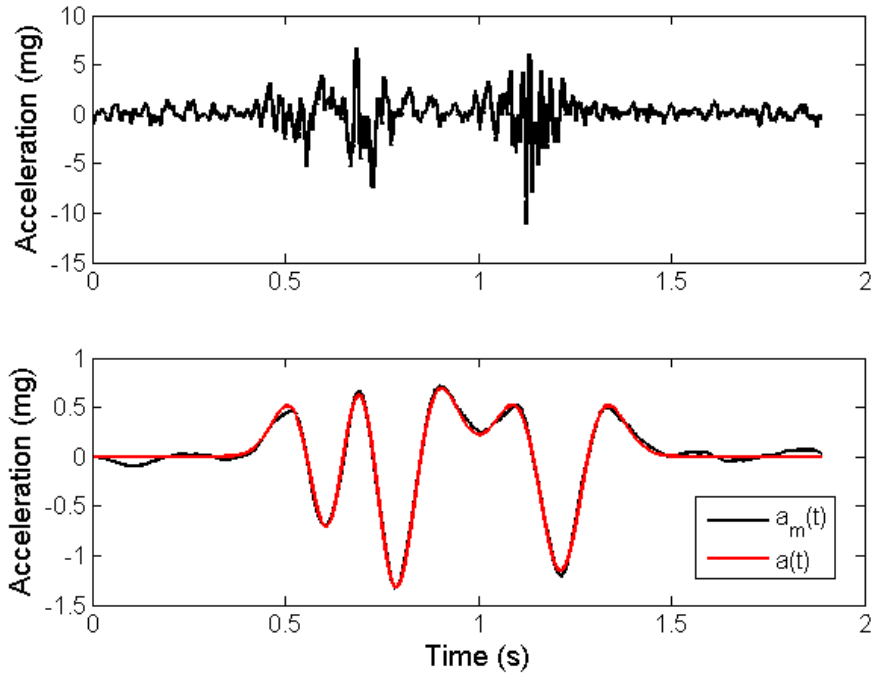


Figure 5.8: Top plot shows the raw acceleration signal measured by the reference sensor. Bottom plot shows the average pavement response $a_m(t)$ and the fitted response $a(t)$. There are 3 mexican-hat functions in $a(t)$ (at 0.6, 0.8, and 1.2 s resp.), each corresponding to an axle group. The response due to the last axle is well isolated from the others but the response for the first two axles needs to be decoupled.

loads can be estimated using Equation (5.8), where the parameter β_i is estimated using the calibration procedure described below.

Model calibration. In general, β_i depends on pavement properties at the site but a set of pre-weighed trucks can be used to estimate it. Let N be the number of trucks in the training set, $\hat{f}_i^n$ be the load (or force) estimate for the i^{th} axle of n^{th} truck, f_i^n be the corresponding true weight, v_n be the speed, T_n be the corresponding pavement temperature, α_i^n be the corresponding fitted parameter α_i^*, and e_i^n be the error associated with the load estimates.

The optimal β_i can be calculated by minimizing the mean squared error for load estimates,

$$\hat{f}_i^n = \beta_i \frac{\alpha_i^n}{v_n^2} \tau(T_n), \tag{5.12}$$

$$e_i^n = \beta_i \frac{\alpha_i^n}{v_n^2} \tau(T_n) - f_i^n,$$

$$\beta_i^* = \arg\min_{\beta} \frac{1}{N} \sum_{n=1}^{N} (e_i^n)^2,$$

$$\boxed{\beta_i^* = \arg\min_{\beta} \sum_{n=1}^{N} (\beta \frac{\alpha_i^n}{v_n^2} \tau(T_n) - f_i^n)^2.} \tag{5.13}$$

Equation (5.13) is a standard linear least-squares problem and can be solved for β_i^*. Once β_i^* is known, individual axle loads can be estimated using Equation (5.12). The estimate for total truck weight $\hat{f}_T^n$ is simply the sum of individual axle loads,

$$\hat{f}_T^n = \sum_{i=0}^{K} \hat{f}_i^n. \tag{5.14}$$

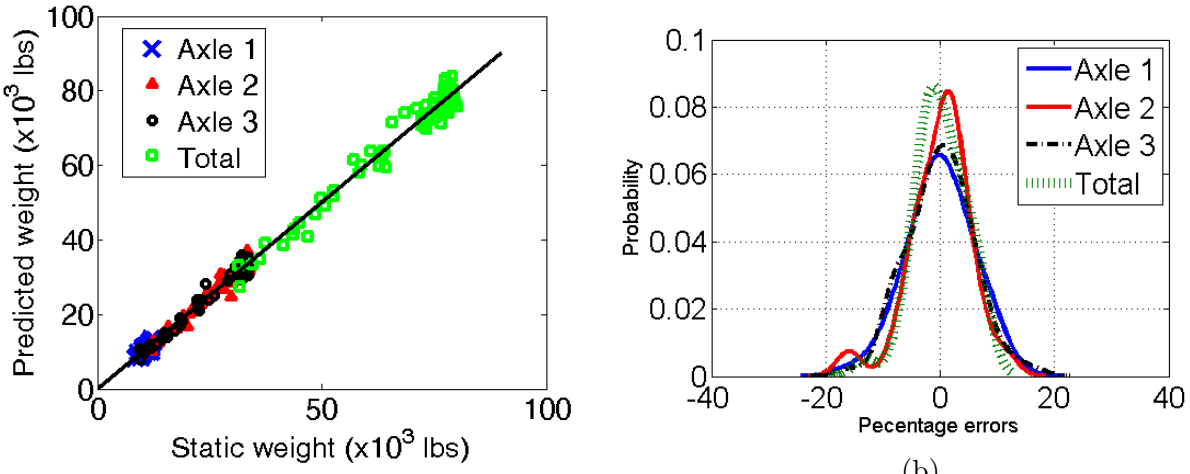
5.4 Results

The results discussed in this section serve two important goals: to verify that the proposed model fits the data well, and to evaluate the accuracy of wireless WIM.

5.4.1 Model verification

We calibrate the model using all 75 trucks and examine how closely it explains the data. Figure 5.9(a) compares the axle weights estimated by our system with their true weights. The estimated loads track the true loads very closely, with a R^2 value of 0.99 for the fit. Figure 5.9(b) shows the estimated probability distribution functions of the corresponding errors. The means and standard deviations associated with these bell-shaped curves are summarized in Table 5.1.

Figure 5.10(a) shows the percentage errors of load estimates at different pavement temperatures. The errors are uncorrelated to temperature, implying that compensation factor $\tau(T)$ captures the effect of pavement temperature well. Figure 5.10(b) shows that the errors are much higher when no temperature compensation is used (i.e. $\tau(T) = 1\ \forall T$). The mean errors are negative for low temperatures and positive for high temperatures. Consistent with pavement models, without temperature compensation loads are overestimated at higher temperatures and underestimated estimated at lower temperatures. This is because pavement response for any load increases with temperature [50]. The error distributions for both scenarios are summarized in Table 5.1.



(a) (b)

Figure 5.9: (a) Plot shows the estimated weights against the ground truth static weights. (b) Plot shows the probability distribution of percentage errors in load estimates for each axle and the entire vehicle.

Table 5.1: Effect of pavement temperature on load estimation. The errors are lower when data is compensated for temperature variation.

	Temperature compensation			No compensation		
	Mean of errors (%)	Std of errors (%)	LTPP Error	Mean of errors (%)	Std of errors (%)	LTPP Error
Axle 1	-0.35	5.83	11.98	-0.18	6.41	12.95
Axle 2	-0.68	5.36	11.35	-0.22	6.15	12.49
Axle 3	-0.78	5.67	12.08	-0.41	6.26	12.89
Total	-0.63	4.46	9.5	-0.25	5.32	10.85

5.4.2 Wireless WIM accuracy

For the results above, we use the entire dataset for training our system. In general, this leads to overfitting and overestimation of the predictive power of a model. For a more realistic evaluation of the system accuracy, we use the *repeated random sub-sampling validation* technique of cross-validation.

We simulate 1000 different training and testing trials. In each trial, we randomly select 25 out of 75 trucks for training our model[2] and use the calculated β_i's for estimating truck weights in the testing set. We then calculate the LTPP errors (e_λ) for all 1000 test sets and

[2]The 16 calibration truck runs from February 14^{th} are always included in the training set and the other 9 trucks (out of 25) are chosen randomly.

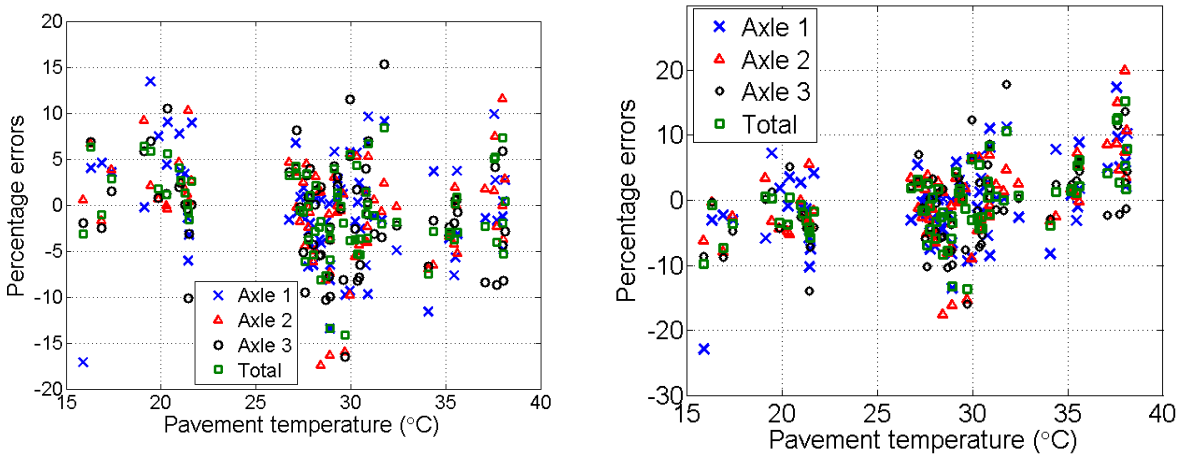


Figure 5.10: (a) Plot shows the percentage error in load estimates against pavement temperature when measurements are compensated for temperature variation. (b) Plot shows the percentage errors when temperature compensation is not applied. Errors increase with increase in temperature.

report the mean LTPP error ($\bar{e_\lambda}$)as an evaluation metric for system accuracy.

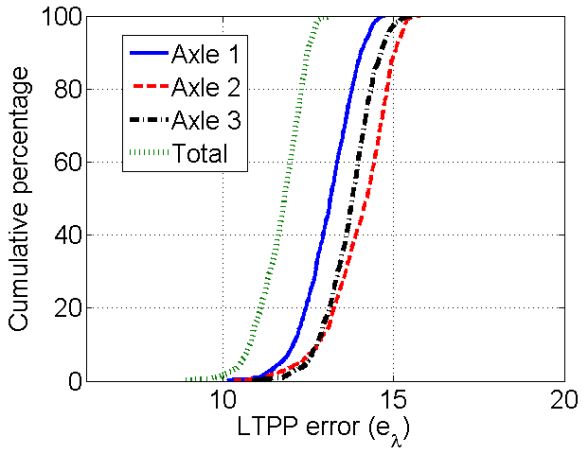


Figure 5.11: Plot shows cumulative distribution for the LTPP errors from the 1000 trials. Majority of trials pass the LTPP specification for allowed errors.

Table 6.5 contains the mean LTPP error ($\bar{e_\lambda}$) from the 1000 trials and the maximum allowed errors by the LTPP standard. The mean LTPP error for each case is less than the allowed error, except for the total weight. We discuss in next chapter how the estimate of total weight

can be improved to pass the LTPP accuracy standards. Figure 5.11 shows the cumulative distribution of e_λ based on the LTPP errors from the 1000 trials. For individual axles, the LTPP error is less than the maximum allowed error for majority of the trials.

Table 5.2: Mean LTPP errors for 1000 trials.

	Wireless WIM error	Maximum allowed error
Axle 1	13.08	20
Axle 2	14.01	15
Axle 3	13.71	15
Total	11.69	10

Chapter 6

Load estimation: improvements and adaptive data compression

In the previous chapter we proposed a pavement-vehicle interaction model and presented a load estimation procedure based on the model. It was shown that the wireless WIM meets the LTPP's specification for individual axle loads but fails to meet the specification for total truck weight. Moreover, the described procedure does not address the challenges posed by the limited energy and processing power available to wireless sensors.

The first half of this chapter discusses some improvements in the model and the load estimation procedure to increase the system accuracy. The second half of the chapter discusses the challenges posed by limited sensor resources and presents an energy efficient-algorithm for the wireless WIM.

6.1 Improvements in load estimation

In this section, we present some changes to our model and the load estimation procedure to improve system accuracy. We start by optimizing the dependency of vehicle speed on pavement response. This leads to a more accurate pavement-vehicle interaction model but a slightly more complex calibration procedure. We then compare two alternative cost functions for calculation of β: mean-squared percentage error (MSPE) and the LTPP error (e_λ). Next, we analyze the array-by-array performance of load estimates and discuss how to optimally combine measurements from different arrays. We derive an analytical solution for optimal $\boldsymbol{\beta}$ in Theorem 6.1 and end this section by presenting an alternative procedure for estimating gross weights.

6.1.1 Model updates

One of the simplifying assumptions that we made in our model was that the amplitude of the pavement displacement η is independent of vehicle speed v, see Equation (5.3). However, there is a weak dependency between η and v as peak displacement decreases with increasing vehicle speed [16]. We model this dependency as, $\eta(v) = v^{-\theta}$. Substituting this in equation (5.3) and following the derivation for $a(t)$ and F, we can obtain the following equations:

$$y(t) = F\eta v^{-\theta} e^{\frac{-t^2}{2\sigma^2}},$$

$$a(t) = -F\eta \frac{v^{2-\theta}}{\sigma_0^2}\left(1 - \frac{t^2}{\sigma^2}\right) e^{\frac{-t^2}{2\sigma^2}},$$

$$\alpha = \frac{F\eta v^{2-\theta}}{\sigma_0^2} = \frac{F\eta v^{\rho}}{\sigma_0^2}, \tag{6.1}$$

$$F = \beta \frac{\alpha\tau(T)}{v^{\rho}}. \tag{6.2}$$

The updated model has an addition unknown parameter ρ that needs to be calibrated. However, this parameter can be estimated using a single truck of known or unknown weight. Equation (6.1) shows that α for each axle increases with speed and follows a power law. Therefore, repeated runs of the same truck at different speeds can be used to estimate ρ. For this purpose, we use the 16 runs of the calibration truck[1] from February 14^{th}.

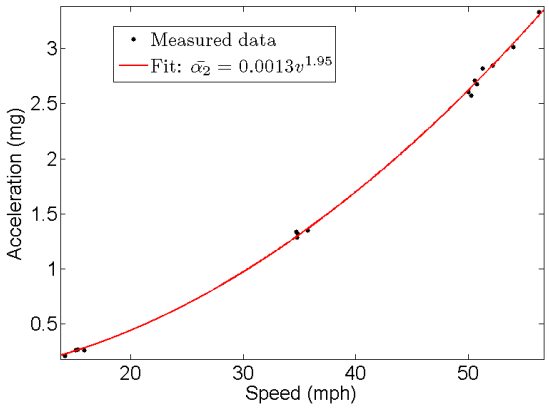


Figure 6.1: Effect of speed on measured sensor response.

Figure 6.1 shows the individual axle response α_2 plotted against the truck speed. A curve fit for the power-law is obtained using the data and ρ for Axle 2 is found to be 1.95. Similarly,

[1]For easier comparisons, we included these trucks in every training set of the 1000 trials used to benchmark system accuracy in the last chapter.

ρ for Axle 1 and 3 were found to be 1.92 and 1.94 respectively. To estimate the LTPP error after this change in model, we follow the same procedure of 1000 cross-validation trials described in the last chapter. We estimate the optimal β by minimizing the MSE for the training set and use it to estimate the loads in each test set. Table 6.1 compares the mean LTPP errors ($\bar{e_\lambda}$) for the updated model with the previous model. The errors decrease by at least 1.6% for individual axles and 2% for the gross-weight. The resulting errors now pass the LTPP specification.

Table 6.1: Effect of speed compensation on system accuracy (LTPP error). ρ is calculated for each axle using multiple runs of a calibration truck at different speeds.

	v^2	$v^{2-\rho}$
Axle 1	13.08	11.06
Axle 2	14.01	12.37
Axle 3	13.71	12.08
Total	11.69	9.74

6.1.2 Cost function for β calibration

Our system performance is based on percentage error in weights and not the error in tons, whereas we find the optimal β by minimizing the MSE for load estimates in tons. In this section, we change the cost function $c(\beta)$ to the MSPE and the LTPP error. We drop the index i for each axle but the following discussion is applicable to all axles.

MSPE as the cost function. Let n denote the truck number, α_n be the estimated parameter for some axle of the n^{th} truck. We can write the error e_n for this axle as a fraction of its true load f_n,

$$e_n = \frac{\hat{f}_n - f_n}{f_n},$$
$$e_n = \beta \frac{\alpha_n \tau(T_n)}{v_n^\rho f_n} - 1 = \beta\gamma_n - 1. \tag{6.3}$$

The optimal β can be found by minimizing the mean-squared percentage error over the training set of N trucks,

$$\beta^* = \arg\min_\beta \sum_{n=1}^{N} (\beta\gamma_n - 1)^2. \tag{6.4}$$

Equation (6.4) is a standard linear least-squares problem with known γ_n and can be solved for β^*. Using the new cost function we repeated the cross-validation procedure and Table 6.2 shows the resulting mean LTPP errors. The errors reduced for Axle 2 and the gross weight, stayed the same for Axle 1, and increased for Axle 3.

Table 6.2: Effect of cost function $c(\beta)$ on system accuracy. The mean LTPP errors are lowest when the LTPP error $e_\lambda(\beta)$ is minimized for calibration.

	MSE	MSPE	LTPP
Axle 1	11.06	11.07	10.84
Axle 2	12.37	12.13	12.01
Axle 3	12.08	12.22	12.22
Total	9.74	9.64	9.48

LTPP error (e_λ) as the cost function. Since we specify system accuracy using LTPP errors, the most appropriate cost function for calibration is the LTPP function itself. Let $\bar{e} = \frac{1}{N}\sum_{n=1}^{N} e_n$ be the mean error for the training set, then we can find the optimal β^* as:

$$\begin{aligned}
\beta^* &= \arg\min_{\beta} \quad \big(|\bar{e}(\beta)| + t_{N-1}\hat{\sigma}_e(\beta)\big), \\
&= \arg\min_{\beta} \quad \left(\left|\frac{1}{N}\sum_{n=1}^{N} e_n(\beta)\right| + t_{N-1}\sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(e_n(\beta)-\bar{e}(\beta))^2}\right), \\
&= \arg\min_{\beta} \quad \left(\left|\beta\bar{\gamma}-1\right| + \frac{t_{N-1}}{\sqrt{N-1}}\sqrt{\sum_{n=1}^{N}(\beta(\gamma_n-\bar{\gamma}))^2}\right). \qquad (6.5)
\end{aligned}$$

The above optimization problem can be solved using numerical convex-optimization techniques or we can use the analytical solution provided in Corollary 6.1. Using the new estimates of β^*, we calculate the mean LTPP errors using crossvalidation. Table 6.2 compares the errors for the three cost functions. Except for Axle 3, lowest errors are obtained when LTPP error is used as the cost function. Since error in gross weights is closest to its maximum allowed error, the cost function that minimizes the errors in gross weight is chosen i.e. $c(\beta) = e_\lambda(\beta)$.

6.1.3 Array-by-array analysis

The suspension oscillations due to road roughness cause the applied load to be different from static load when the truck reaches each sensor array. We average the measurements from all arrays to reduce the effect of dynamic load on load estimates. We now analyze how the system accuracy depends on the number of arrays and then discuss how measurements from different arrays can be optimally combined to improve the accuracy.

Effect of dynamic load. Let $F_I(t)$ be the instantaneous force applied by a truck axle and F be its static load. The difference between the two ($F_D(t) = F_I(t)$-F), called dynamic load, is typically within 30% of F [16] and can lead to high errors in load estimates. In commercial WIM systems, this challenge is overcome by installing a smooth concrete pavement around

the force sensor to minimize the suspension oscillations. However, this dramatically increases the cost of the WIM system. An alternative to reduce weight errors is described in [11], where measurements from multiple force sensors on existing pavement are averaged to estimate the static load. Based on this concept, we use multiple vibration sensor arrays (setup shown in Figure 3.8(a)) to reduce the effect of dynamic load. Each array essentially measures the static load with some uncertainty and by averaging multiple measurements we reduce the amount of uncertainty in our load estimate. Figure 6.2(a) shows that the LTPP error decreases as the number of arrays are increased. Thus, adding more sensor arrays to the wireless WIM can potentially increase its accuracy.

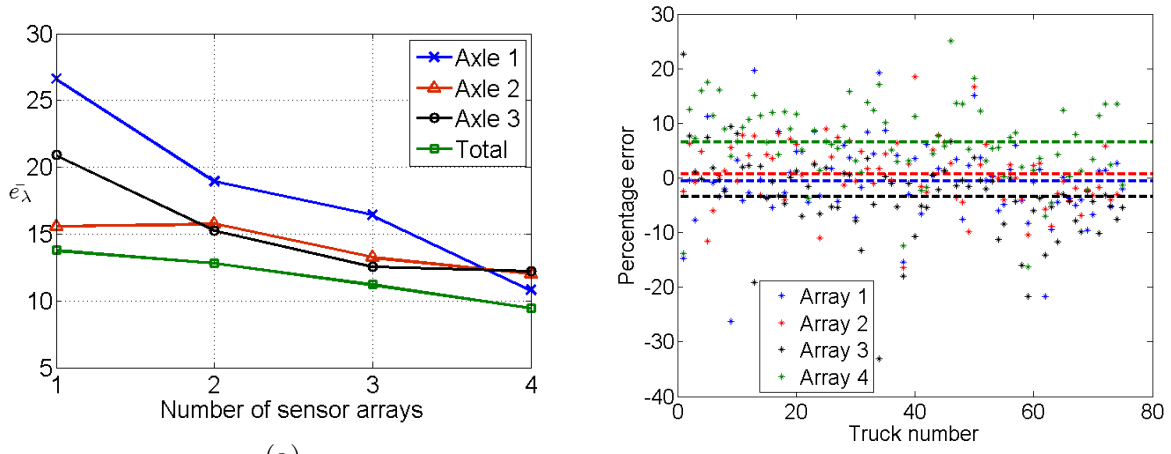


Figure 6.2: (a) Plot shows that errors in load estimates decrease as the number of arrays increase. (b) Percentage error in array-wise load estimates for Axle 2. The dotted lines represent the mean error for each array.

Optimal scaling coefficients for each array. So far the measurements from different arrays were combined using equal scaling coefficients. In this section we allow them to vary for each array and analyze how the system accuracy changes.

Using the procedure described in Section 5.3, we align and average sensor measurements from each array to obtain the *average array response*. We then extract the individual axle response and estimate the weights from each array using the previously estimated β. Figure 6.2(b) shows a different mean error (or bias) for estimates from different arrays. The bias suggests that each array should have its own unique β. For instance, Array 4 has a positive bias which suggests that a smaller β should be used for this array.

We drop the index for axle number in the following discussion but the discussion is relevant for all axles. Let N be the total number of trucks in the training set, M be the total number of arrays, $\boldsymbol{\beta} = [\beta_1 \ \beta_2 \ ... \ \beta_M]^\top$ be the vector representing the unknown scaling coefficients

or calibration factors for each array, α_{ni} be an individual axle response for the n^{th} truck obtained from the i^{th} sensor array, f_n be the true load for some axle of the n^{th} truck, $\hat{f}_n$ be our corresponding estimate, v_n be the speed, T_n be corresponding pavement temperature, then we can write the load estimates and the corresponding errors as

$$\begin{aligned}
\hat{f}_n &= \sum_{i=1}^{M} \beta_i \frac{\tau(T_n)\alpha_{ni}}{v_n^{\rho}}, \\
e_n &= \frac{\hat{f}_n - f_n}{f_n} = \frac{\hat{f}_n}{f_n} - 1 = \sum_{i=1}^{M} \beta_i \frac{\tau(T_n)\alpha_{ni}}{v_n^{\rho} f_n} - 1 = \sum_{i=1}^{M} \beta_i \gamma_{ni} - 1,
\end{aligned} \tag{6.6}$$

where $\gamma_{ni} = \frac{\tau(T_n)\alpha_{ni}}{v_n^{\rho} f_n}$. The cost function or LTPP error e_λ can be written as

$$\begin{aligned}
c(\boldsymbol{\beta}) &= \left| \frac{1}{N} \sum_{n=1}^{N} e_n \right| + t_{N-1} \sqrt{\frac{1}{N-1} \sum_{n=1}^{N} (e_n - \bar{e})^2}, \\
&= \left| \sum_{i=1}^{M} \beta_i \bar{\gamma}_i - 1 \right| + \frac{t_{N-1}}{\sqrt{N-1}} \sqrt{\sum_{n=1}^{N} \Big(\sum_{i=1}^{M} \beta_i (\gamma_{ni} - \bar{\gamma}_i) \Big)^2}, \\
&= \left| \sum_{i=1}^{M} \beta_i \bar{\gamma}_i - 1 \right| + \frac{t_{N-1}}{\sqrt{N-1}} \sqrt{\sum_{n=1}^{N} \Big(\sum_{i=1}^{M} \beta_i \delta_{ni} \Big)^2}, \\
&= \left| \boldsymbol{\beta}^{\top} \bar{\boldsymbol{\gamma}} - 1 \right| + \frac{t_{N-1}}{\sqrt{N-1}} \sqrt{\boldsymbol{\beta}^{\top} \boldsymbol{\delta}^{\top} \boldsymbol{\delta} \boldsymbol{\beta}},
\end{aligned}$$

where $\boldsymbol{\delta} = \big[\delta_{ni}\big]$, $\delta_{ni} = \gamma_{ni} - \bar{\gamma}_i$, $\bar{\gamma}_i = \frac{1}{N} \sum_{n=1}^{N} \gamma_{ni}$, and $\bar{\boldsymbol{\gamma}} = [\bar{\gamma}_1 \ \bar{\gamma}_2 \ ... \ \bar{\gamma}_M]^{\top}$. The calibration parameters minimize the above cost function, i.e. $\boldsymbol{\beta}^{*} = \arg\min c(\boldsymbol{\beta})$. This convex optimization problem can be solved numerically using standard techniques such as the Gauss-Newton algorithm. Table 6.3 compares the mean LTPP errors for the single-β case with the multi-β case. Contrary to our expectations, the errors are higher for the multi-β case but we also observed high variability in the estimated calibration parameters. Figure 6.3(a) shows the estimated $\boldsymbol{\beta}$ for Axle 3 in each of the 1000 trials. The plot shows the non-robustness of the estimated parameters to changes in the training set. To investigate the source of this non-robustness, we now derive an analytical solution for $\boldsymbol{\beta}^{*}$.

Theorem 6.1. *Let $\boldsymbol{\beta} = [\beta_1 \ \beta_2 \ ... \ \beta_M]^{\top}$ be the M unknown calibration factors and the optimal $\boldsymbol{\beta}^{*}$ minimize the cost function $c(\boldsymbol{\beta})$ i.e.*

$$\boldsymbol{\beta}^{*} = \arg\min_{\boldsymbol{\beta}} \left\{ \left| \boldsymbol{\beta}^{\top} \bar{\boldsymbol{\gamma}} - 1 \right| + \frac{t_{N-1}}{\sqrt{N-1}} \sqrt{\boldsymbol{\beta}^{\top} \boldsymbol{\delta}^{\top} \boldsymbol{\delta} \boldsymbol{\beta}} \right\}. \tag{6.7}$$

Table 6.3: Change in system accuracy after using optimal β's for each array.

	Single β	Multiple β's (no regularization)	Multiple β's (regularization)
Axle 1	10.84	11.2	11.77
Axle 2	12.01	11.87	11.81
Axle 3	12.22	13.04	12.32
Gross	9.48	9.64	9.45

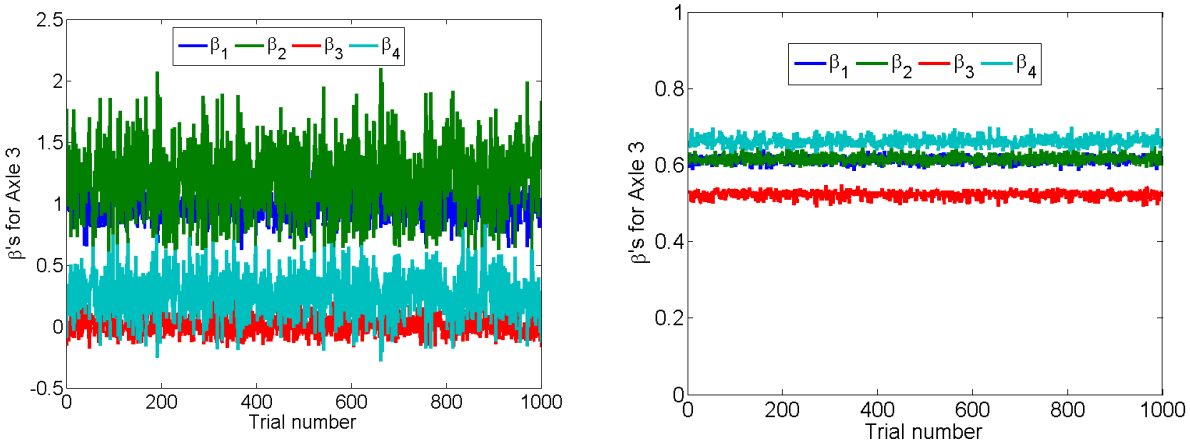


Figure 6.3: Robustness of estimated β's. (a) No regularization is used. (b) Tikhonov regularization is used.

The solution for this convex optimization problem is given by

$$\boldsymbol{\beta}^* = \begin{cases} \frac{(\boldsymbol{\delta}^\top \boldsymbol{\delta})^{-1}\bar{\boldsymbol{\gamma}}}{\bar{\boldsymbol{\gamma}}^\top (\boldsymbol{\delta}^\top \boldsymbol{\delta})^{-1}\bar{\boldsymbol{\gamma}}}, & \text{if } \sqrt{\bar{\boldsymbol{\gamma}}^\top (\boldsymbol{\delta}^\top \boldsymbol{\delta})^{-1}\bar{\boldsymbol{\gamma}}} > \frac{t_{N-1}}{\sqrt{N-1}}, \\ 0, & \text{if } \sqrt{\bar{\boldsymbol{\gamma}}^\top (\boldsymbol{\delta}^\top \boldsymbol{\delta})^{-1}\bar{\boldsymbol{\gamma}}} < \frac{t_{N-1}}{\sqrt{N-1}}. \end{cases} \tag{6.8}$$

Proof. We start by assuming that the solution lies on some plane ${\boldsymbol{\beta}^*}^\top \bar{\boldsymbol{\gamma}} = 1 + \epsilon$, where $\epsilon \in \Re$. Since ϵ can take on any real value, this plane covers the entire M-dimensional space as ϵ is varied. We will show that the $\epsilon = 0$ for the non-trivial solution which implies that the optimal solution causes the mean error for the N training samples to be zero, see Equation (6.6).

We can now rewrite the optimization problem as,

$$\begin{aligned}\boldsymbol{\beta}^{*}(\epsilon) &= \arg \min_{\boldsymbol{\beta}|\boldsymbol{\beta}^{\top}\bar{\boldsymbol{\gamma}}=1+\epsilon} \left\{ |\epsilon| + \frac{t_{N-1}}{\sqrt{N-1}} \sqrt{\boldsymbol{\beta}^{\top}\boldsymbol{\delta}^{\top}\boldsymbol{\delta}\boldsymbol{\beta}} \right\}, \\ &= \arg \min_{\boldsymbol{\beta}|\boldsymbol{\beta}^{\top}\bar{\boldsymbol{\gamma}}=1+\epsilon} \boldsymbol{\beta}^{\top}\boldsymbol{\delta}^{\top}\boldsymbol{\delta}\boldsymbol{\beta}.\end{aligned}$$

We can now use Lagrange multipliers to solve the problem. The Lagrangian can be written as

$$\Lambda(\boldsymbol{\beta}, \lambda) = \boldsymbol{\beta}^{\top}\boldsymbol{\delta}^{\top}\boldsymbol{\delta}\boldsymbol{\beta} + \lambda(\boldsymbol{\beta}^{\top}\bar{\boldsymbol{\gamma}} - 1 - \epsilon).$$

The critical points can be found as

$$\begin{aligned}\frac{\partial \Lambda}{\partial \boldsymbol{\beta}} = 0 \qquad &\Rightarrow \quad 2\boldsymbol{\delta}^{\top}\boldsymbol{\delta}\boldsymbol{\beta} + \lambda\bar{\boldsymbol{\gamma}} = 0, \\ \frac{\partial \Lambda}{\partial \lambda} = 0 \qquad &\Rightarrow \quad \boldsymbol{\beta}^{\top}\bar{\boldsymbol{\gamma}} - 1 - \epsilon = 0.\end{aligned}$$

Solving the linear equations leads to

$$\begin{aligned}\lambda &= \frac{2(1+\epsilon)}{\bar{\boldsymbol{\gamma}}^{\top}(\boldsymbol{\delta}^{\top}\boldsymbol{\delta})^{-1}\bar{\boldsymbol{\gamma}}}, \\ \boldsymbol{\beta}^{*}(\epsilon) &= \frac{(\boldsymbol{\delta}^{\top}\boldsymbol{\delta})^{-1}\bar{\boldsymbol{\gamma}}}{\bar{\boldsymbol{\gamma}}^{\top}(\boldsymbol{\delta}^{\top}\boldsymbol{\delta})^{-1}\bar{\boldsymbol{\gamma}}}(1+\epsilon). \qquad (6.9)\end{aligned}$$

The corresponding minimum cost is given by

$$\begin{aligned}c(\boldsymbol{\beta}^{*}, \epsilon) &= |\epsilon| + \frac{t_{N-1}}{\sqrt{N-1}} \frac{|1+\epsilon|}{\bar{\boldsymbol{\gamma}}^{\top}(\boldsymbol{\delta}^{\top}\boldsymbol{\delta})^{-1}\bar{\boldsymbol{\gamma}}}, \\ &= |\epsilon| + c_N|1+\epsilon|.\end{aligned}$$

This cost is minimized at either $\epsilon = 0$ or $\epsilon = -1$, depending on whether $c_N < 1$ or $c_N > 1$. Substituting the two possible values for ϵ in equation (6.9) leads to the solution given by the theorem. ■

It should be noted that when $\epsilon = -1$ the cost function $c(\boldsymbol{\beta}^{*}) = 1$ i.e. the LTPP error is 100%. The optimal solution in this case is trivial i.e. $\boldsymbol{\beta}^{*} = 0$. This solution was not observed in any training/testing trials since our measurements are highly correlated to axle loads.

Corollary 6.1. *If $\boldsymbol{\beta}$ is a scalar, then the optimal solution for the corresponding cost function (equation (6.5)) is given by*

$$\beta = \frac{1}{\bar{\gamma}}.$$

Proof. For the single-β case, $\bar{\boldsymbol{\gamma}}$ is a scalar and so is $\boldsymbol{\delta}^\top\boldsymbol{\delta}$. Therefore, the result follows from Theorem 6.1. ■

The analytical solution provides more insight into the non-robustness of the estimated parameters. Observation of the various terms in the expression for different training sets revealed that the slight variation in $\bar{\gamma}$ in each training set is magnified because the matrix $\boldsymbol{\delta}^\top\boldsymbol{\delta}$ is not well-conditioned. Tikhonov regularization [59] of this matrix results in more robust estimates of $\boldsymbol{\beta}$ as shown in Figure 6.3(b).

Table 6.3 compares the mean LTPP errors found for the three scenarios: (i) single-β case, (ii) multi-β case without regularization, and (iii) multi-β case with the matrix $\boldsymbol{\delta}^\top\boldsymbol{\delta}$ regularized using Tikhonov regularization. It is clear that regularization helped in reducing errors for the multi-β case. However, compared to the single-β case the improvements are insignificant, if any. The increased complexity in the model lead to lower errors in training sets but the predictive power of the model, based on the errors in test sets, is practically unchanged. Thus, a single-β for all arrays is preferred over the multi-β case.

6.1.4 Separate regression for gross weight

Until now the estimate of gross weight was simply the sum of individual axle weights. We now consider a separate model for the gross-weight estimates and calibrate this model by minimizing the LTPP error. Let the gross weight estimate for a truck be modeled as

$$\hat{f}_T = \sum_{i=1}^{K} \beta_{Ti} \frac{\alpha_i \tau(T)}{v^{\rho_i}}, \tag{6.10}$$

where α_i is the decoupled response for the i^{th} axle and β_{Ti} is the corresponding scaling coefficient. The calibration factors (β_{Ti}) can be estimated by minimizing the LTPP error for gross weights over the training set. Theorem 6.1 can also be used for estimating β_{Ti}. The only difference is that index i represented measurements from different arrays and now it represents the measurements for different axles. We repeated the 1000 cross-validation trials and found the mean LTPP error for gross weights to be 11.88% using this new model. Thus, there is no improvement from doing a separate regression for gross weights.

Remark 6.1. *Based on our discussion in this section, the following steps lead to the best load estimates:*

1. *Individual axle response α_i for the i^{th} axle is extracted from the* ***average pavement response****.*
2. *Calibration parameter ρ is estimated using multiple runs of the same truck at different speeds.*
3. *Calibration of a single β is done by minimizing the LTPP error for the training set.*

4. *Individual axle loads are calculated using Equation (6.2).*

5. *The gross weight is estimated as the sum of individual axle weights.*

6.2 Energy-efficient algorithm for load estimation

The load estimation procedure described so far is very inefficient for a resource constrained wireless sensor network. All raw data is transmitted by the sensors; however, wireless data transmission is the most power consuming process and the amount of data transmitted should be minimized. This section discusses how adaptive rate sampling and the distribution of computation between the access point and sensors can lead to increased sensor lifetime while only minimally affecting the system accuracy. The algorithm presented in this section was tested using offline processing of collected data. Actual implementation of this algorithm in the wireless sensor network needs to be tested.

One possible approach to compress data is to implement the curve-fitting procedure inside the wireless sensor and transmit just the fitted parameters. However, there are many drawbacks to this approach:

- It is not clear if the computationally intensive procedure can be implemented in a wireless sensor with limited processing power.
- The curve-fitting procedure is performed using acceleration for the entire truck but the vibration sensors do not know the time window for the truck's presence. This information can be transmitted by the AP but the delay through the network is too long and the limited sensor memory is not enough to store the data until the information is received.
- The parameters for curve-fitting are initialized based on tire-on-top sensor data and speed of the vehicle, both of which are unknown at sensor level. Without good initial parameters, the non-convex optimization problem can converge to a local minima and lead to inaccurate estimates of individual axle response.

An alternative to the above approach is to perform the curve-fitting procedure at the AP level but reduce the sensor data required for the fitting procedure. The vibration sensor consumes an average current of 370 μA without the radio transmissions and the required current consumption for a 2 year lifetime (using a single battery) is 410 μA. Assuming that an axle arrives about every 4.5 seconds (based on Caltrans data) and the required average current consumption to be 410 μA, we estimated a data budget of L_0=72 samples/axle for the vibration sensor. Equivalently, the duty cycle of the radio needs to be less than 0.2%. The goal now is to figure out which L ($\leq L_0$) samples should be chosen for each axle and we discuss this problem below in the context of an isolated axle.

6.2.1 Adaptive sampling

The filtered acceleration corresponding to each axle (axle pulse) can be modeled as a Mexican-hat function in the presence of additive random noise, i.e.

$$m(n) = -\alpha \left(1 - \frac{n^2}{\sigma^2}\right) e^{\frac{-n^2}{2\sigma^2}} + \epsilon(n).$$

The amplitude α is linearly proportional to axle load, σ is inversely proportional to axle speed, and the time of axle arrival μ is assumed to be zero. The vibration sensor sees a continuous stream of acceleration data that may or may not contain an axle pulse, depending on whether an axle drives over the system or not. When we send out all raw data (no redundant packets), the duty cycle for the radio is 3.2% and the average current consumption is 1.01 mA, both of which exceed our budget. The expected lifetime, based on a single 7200 mA battery, is slightly less than 10 months.

To reduce the amount of data transmitted, sensors should report data only when a loaded axle is detected. To detect an axle the vibration sensor can filter measured data and look for any negative peaks that have a magnitude greater than a chosen threshold. Alternatively, the axle detection algorithm presented in Section 4.1.2 can be adapted: $M(v)$ can be calculated based on the average traffic speed received periodically from the AP. The time period during which the axle pulse is"dominant" is called the pulse span (PS) and only the measurements that lie in this span should be transmitted. For the m(n) above, we define the PS to be [-3σ,+3σ] as majority of the pulse energy belongs to this span. This change alone drastically reduces the amount of data being transmitted by the sensor. The conventional WIM at Pinole reports that our sensor lane, on average, receives a new axle after every 4.5 seconds. If we transmit just the pulse data from the time-series of measurements, we effectively reduce the duty cycle for the radio to 0.36%. The corresponding average current consumption is 0.441 mA with an expected sensor lifetime of 1.86 years or 22 months.

To further reduce the data transmitted, we now subsample the axle pulse such that the reconstructed signal is minimally affected. Since peak acceleration is very important for load estimation, we always retain the time and the magnitude of each axle peak. Also, Observation 5.3 and the term $\Phi(vt)$ in our model suggest that the frequency bandwidth of each axle pulse is directly proportional to the axle speed and inversely proportional to the pulse span (PS). This presents an opportunity to adapt the sampling rate depending on vehicle speed (or equivalently the pulse span) i.e. sample slower for slower vehicles and faster for faster vehicles. To calculate the PS, we use an important feature of the measured signal: the zero-crossings around each axle peak. It should be noted that the zero crossings of the noise-free function m(n) are at n=$\pm\sigma$ and an estimate of the zero crossings of the measured signal can be found by setting m(n)=0. Let t_{z+}, t_{z-} be the time of the positive and negative

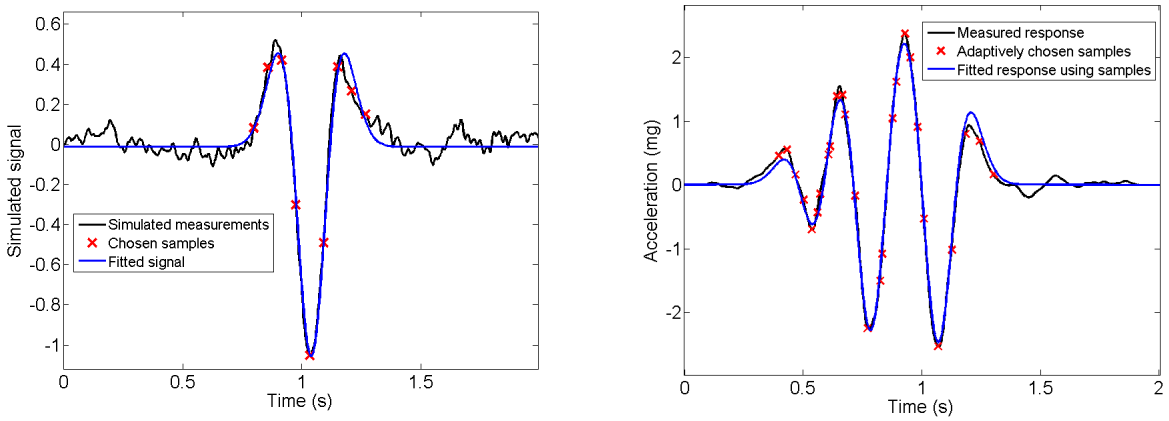
zero-crossings, we can calculate the PS as

$$PS = \quad [+3\sigma, -3\sigma] \quad = \left[+\frac{3}{2}.2\sigma, -\frac{3}{2}.2\sigma \right],$$
$$\approx \left[+\frac{3}{2}.2\hat{\sigma}, -\frac{3}{2}.2\hat{\sigma} \right] = \left[+\frac{3}{2}|t_{z+} - t_{z-}|, -\frac{3}{2}|t_{z+} - t_{z-}| \right].$$

We now select L points around each axle peak with the peak being the central sample and with a sampling interval of

$$\Delta T_L = \frac{3|t_{z+} - t_{z-}|}{L-1}. \tag{6.11}$$

To check the proposed procedure we simulated different functions m(n) using Gaussian random noise, and adaptively sampled and reconstructed the underlying signals. The proposed adaptive sampling worked well at even a low sampling rate of 9 samples/axle. Figure 6.4(a) shows an example of the chosen samples and the curve-fit obtained using these samples.



(a) (b)

Figure 6.4: Adaptive sampling (a) Simulated signal to demonstrate the adaptive sampling for a isolated axle (b) Example fit for a class-9 truck.

For the multi-axle case, we independently sample around each axle peak and use the samples corresponding to the entire truck to perform the curve fitting described in Section 5.3. Figure 6.4(b) shows an example of a class-9 truck. It should be noted that we cannot just use the peaks of the measured response for load estimation because the response of the first two axles is strongly coupled. The signal between the peaks of two strongly coupled axles is of high importance for extracting the individual axle response accurately. The adaptive sampling procedure automatically chooses more points in the overlapping region if the axles

are closer. We now present the load estimation algorithm at both the sensor level and the AP level.

6.2.2 Sensor-level algorithm

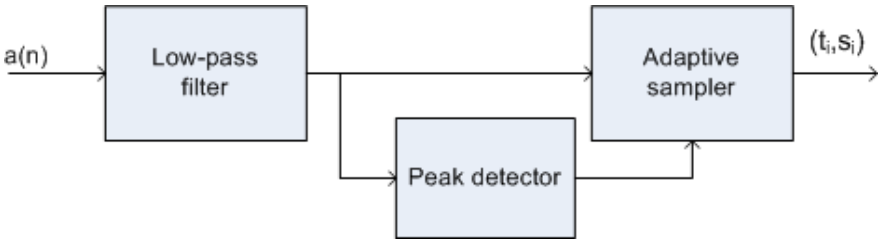


Figure 6.5: Algorithm followed by non-tire-on-top vibration sensors.

Figure 6.5 shows the block diagram of the sensor-level algorithm. If measured acceleration suggests that a truck's tire is directly on top of the sensor, the tire-on-top axle detection data is transmitted (see Section 4.2.2). For a non-tire-on-top sensor, the measured acceleration is filtered and adaptively chosen samples around each axle peak are transmitted over the air.

Raw acceleration signal is first filtered using the same cascade of moving average filters as before but $M(v)$ is calculated using average traffic speed that can be sent by the AP every 30 seconds. The negative peaks in the filtered signals are detected while enforcing the minimum threshold and minimum time separation constraint similar to the axle detection algorithm in Chapter 4. To choose the L samples, sampling period based on the zero crossings around each peak is calculated using Equation (6.11). Instead of reporting the time corresponding to each sample, we only report the time of the axle peak and the sampling period. Since the signal is sampled relative to the axle peak, the AP can calculate the times corresponding to each sample. Therefore, each sensor transmits L acceleration samples, peak time, and the sampling period ΔT_L for each detected axle.

6.2.3 AP-level algorithm

Figure 6.6 shows the block diagram for the algorithm followed by the access point. Using the detection sensor data and the tire-on-top axle data, the AP calculates the truck speed, time window of the truck for each sensor, and the timestamps of each axle as described in previous chapters. The AP then performs the curve-fitting for each sensor using the reported samples during the trucks's time window. An example is shown in Figure 6.4(b). The initial points are guessed using the axle times, truck speed, and the measured peak acceleration as described before. If a false peak is detected by the sensor, the corresponding data is ignored by the AP since the same axle will not be detected using tire-on-top sensor data.

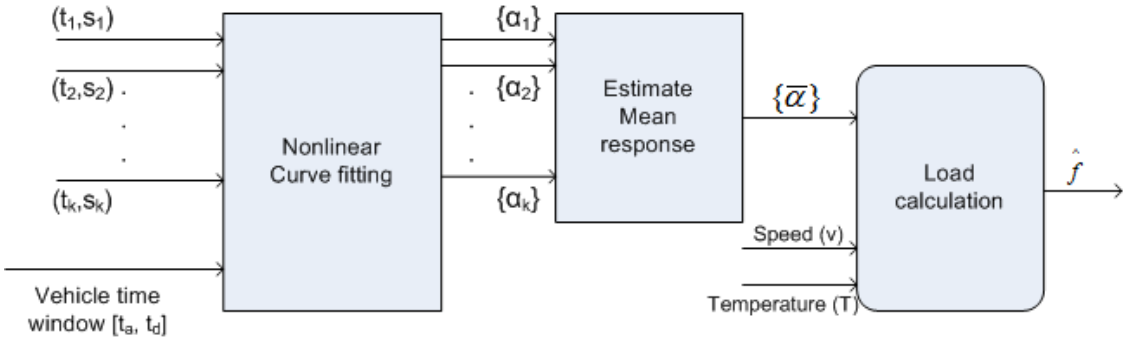


Figure 6.6: Algorithm followed by the access point.

The fitted parameter α_i is averaged across all non-tire-on-top sensors to obtain the mean response $\bar{\alpha}$ for each axle. The AP then calculates individual axle weights and the gross weight by following steps 2-5 in Remark 6.1.

6.2.4 Results

We now present the results obtained by applying the above algorithms on the raw acceleration data. In effect, this helps us simulate any degradation in system performance from the data compression. We end this section by estimating the lifetime of a sensor using the above described algorithm.

Load estimation after adaptive sampling

Using the previously described cross-validation procedure, we evaluated the accuracy of load estimates using the compressed data. Table 6.4 lists the mean LTPP error for the individual axles and gross weight using $L = 9$ samples/axle. For comparison, we provide the results from Section 6.1.2 where raw signals from all sensors are averaged first and the entire time-series of the average pavement response is used for curve-fitting. Comparing the first two columns, we see that the errors in load estimates increase significantly for Axle 1 and the total weight.

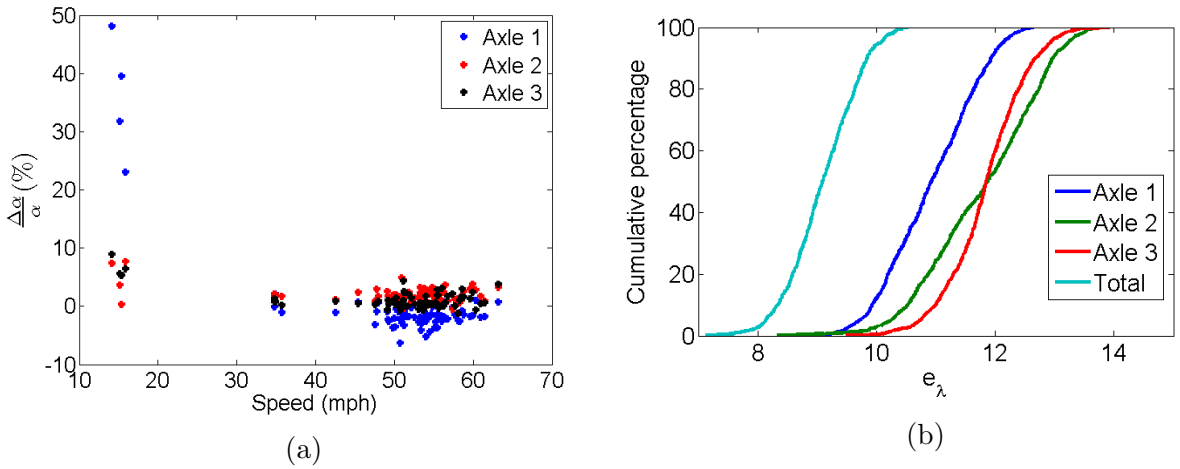
To find the cause of error increase, we analyze the change in estimated α's after compression. Figure 6.7(a) shows the percentage change in α's relative to the ones obtained using uncompressed data. It is clear that Axle 1 of the 4 calibration runs at lowest speed is affected significantly by data compression. These runs have the lowest signal-noise-ratio because of low speeds (15 mph). Since these 4 trucks are always used in the training set, we can use all the raw data for curve-fitting. All the other trucks, including the other calibration runs, use L=9 for each axle. The last column of Table 6.4 shows that the corresponding errors pass the LTPP specification for each axle and the gross weight. Compared to the case of

Table 6.4: Effect of data compression on system accuracy.

	Before compression	**After Compression**	
		L=9	L=9, except for calibration runs
Axle 1	10.84	15.05	10.93
Axle 2	12.01	12.32	11.79
Axle 3	12.22	12.03	11.84
Total	9.48	10.41	9.1

uncompressed data, the errors are slightly higher for Axle 1 but lower for the other axles and the gross weight.

Figure 6.7(b) shows the cumulative distribution for the LTPP errors for 1000 cross-validation trials. The wireless WIM errors for individual axles are below the LTPP allowed errors in every trial. The errors in gross weight are below the allowed limit for 95% of the trials.



(a) (b)

Figure 6.7: (a) Effect of data compression on load estimates. (b) Cumulative distribution of LTPP errors for 1000 random test sets.

For real deployments, there are two options to consider for low speed trucks. The first option is to specify a lower bound on speed for the wireless WIM. This is an acceptable solution since majority of the time truck speeds are higher than 15-20 mph on highways. The other option is to sample more points for lower speed trucks i.e. if the pulse span is greater than a threshold, sample more points per axle. However, it is important to realize that there is a more fundamental lower bound on speed because of the physics of pavement-vehicle interaction. A static truck causes no pavement vibrations and vibrations caused at very low speeds are below the noise floor of the sensor to be measured accurately.

Comparison with the conventional WIM

We now provide a one-on-one comparison of wireless WIM with the nearby WIM station. Due to technical difficulties, weights from this station were not available for one of our testing days (May 29, 2012). This leads to a reduced dataset for comparison, containing 59 truck runs, where 16 runs correspond to the a single calibration truck and remaining 43 runs correspond to random trucks that were stopped and individually weighed. While the weights reported for the calibration truck had low errors, the overall accuracy of the conventional WIM was below expectations. Table 6.5 compares the accuracy of both systems. The wireless WIM clearly outperforms the conventional WIM in every category. The conventional WIM meets the required LTPP accuracy levels for Axle 1 and 2 but fails for Axle 3 and the gross weight.

Table 6.5: Comparison of mean LTPP errors between our system and the nearby conventional WIM. The errors for the conventional WIM are much higher than the errors allowed by the LTPP standard.

	Wireless WIM error	Conventional WIM error	Maximum allowed error
Axle 1	10.93	10.96	20
Axle 2	11.79	14.53	15
Axle 3	11.84	24.65	15
Total	9.1	13.13	10

Estimated sensor lifetime

The conventional WIM station reports that an axle arrives at the sensors about every 4.5 seconds. Based on the time it takes to transmit a payload of 48 bytes, we calculate that duty cycle for the radio to be $(L+1)/720$ % if we send out L samples per axle. For a more conservative estimate and to account for future increase in freight traffic, we use a larger duty cycle of (L+1)/72 %. For L=9, the duty cycle is 0.139% and the expected current consumption can be calculated as

$$I_{avg} = 20\ mA \times 0.00139 + 0.37\ mA \times (1 - .00139) = 397\mu A.$$

Using a single 7200 mA battery, the sensor lifetime is slightly over 2 years. While this does not achieve our target lifetime of 4 years, we have alternatives of achieving that. We can use two batteries to double the lifetime; this will increase the sensor cost slightly and increase the sensor size. The other alternative is to use a lower current consuming accelerometer; the current consumption of the current MEMS accelerometer is about 300 μA by itself. While not available at the time of our sensor design, newer acceleromters are available in the market today that have similar noise density to our current sensor but lower current

consumption. In fact, the manufacturer of the current MEMS accelerometer (Colibrys) now has a lower current consuming alternative that can be easily added to the current circuit board. According to the datasheet, the current consumption of the newer accelerometer, Colibrys MS7002, is half of the the current accelerometer [23].

Chapter 7

Displacement Estimation

The loads reported by a WIM station are used to asses pavement condition and predict its remaining lifetime. This analysis helps a pavement design engineer to schedule timely repairs and reduce the maintenance cost. However, using cumulative information about applied loads and approximate pavement deterioration models is an indirect way to assess pavement condition [1, chap. 5]. A more direct approach is measure the pavement deflection for each applied load. In fact, the same principle is used in popular non-destructive testing methods such as the FWD and HVS. Calibrated loads are applied to the pavement and the measured deflection is used to backcalculate pavement stiffness using layered-elastic theory [44, 60]. Stiffness is a measure of strength of the pavement and can be combined with pavement roughness and cracking information to make decisions regarding road maintenance and rehabilitation.

The vibration sensors measure pavement acceleration and estimating displacement from noisy acceleration data is a challenging problem in itself. In this chapter, we present and compare three algorithms for displacement estimation. To check the accuracy of the estimates, we compare them with reference sensor data obtained during the FWD and HVS experiments. The details for the experiments are provided in Sections 3.1.1 and 3.2.1.

7.1 Algorithms for displacement estimation

We present and compare three different displacement estimation algorithms in this section. The goal is to develop and test algorithms for two potential applications:

- **Controlled experiments.** The FWD and HVS experiments fall under this category. The user can control conditions like the time, type, and magnitude of the applied loads.
- **Smart pavement monitoring system.** This is a futuristic application where pavements are monitored in real-time and preventative repair is scheduled before the appearance of any significant pavement damage. The pavement displacement correspond-

ing to each passing vehicle is measured and combined with weight information to assess pavement condition on a regular basis.

7.1.1 Algorithm 1: Double-integration

This algorithm is based on the fundamental relationship between acceleration and displacement. Pavement velocity $v(t)$ and displacement $y(t)$ can be written in terms of acceleration $a(t)$,

$$v(t) = v(0) + \int_0^t a(\tau)d\tau,$$
$$y(t) = y(0) + v(0)t + \int_0^t \int_0^\xi a(\tau)d\tau d\xi,$$

where $v(0)$, $y(0)$ depend on the initial condition of the pavement and can be set to zero if the pavement was at rest. However, simple double integration of measured acceleration leads to unexpected drifts in estimated displacement. The integration operation disproportionately amplifies the measurement errors at low frequencies leading to a significant drift in displacement. Figure 7.1(a) shows a simulated acceleration signal and a signal obtained by adding random Gaussian noise to it. Figure 7.1(b) shows the displacement estimated by double-integration. The estimate contradicts the physical principles since pavement returns to rest after the load has passed but the estimated final displacement is ≈ 50 μm and not zero. To compensate for the drift, a popular method in literature is to estimate the drift by fitting a polynomial and subtracting it from the double-integrated signal [8, 29] but we follow a slightly simpler method in Algorithm 2.

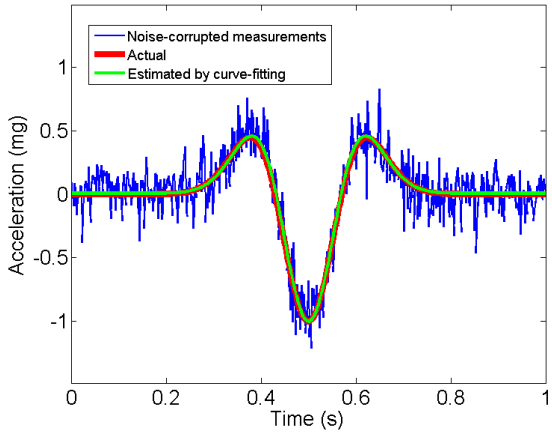
7.1.2 Algorithm 2: constrained least-squares estimation

This algorithm discretizes the relationship between acceleration and displacement, and calculates displacement using constrained least-squares estimation. The sampled acceleration $a(n)$ can be written in terms of pavement velocity $v(n)$ and displacement $y(n)$ as

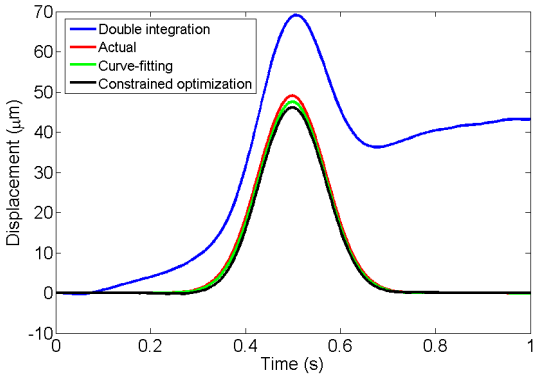
$$v(n) = \frac{y(n) - y(n-1)}{T}, \tag{7.1}$$
$$a(n) = \frac{v(n) - v(n-1)}{T} = \frac{y(n) - 2y(n-1) + y(n-2)}{T^2}. \tag{7.2}$$

Rewriting in matrix form,

$$\overbrace{\begin{bmatrix} a(n) \\ a(n-1) \\ \vdots \\ a(0) \end{bmatrix}}^{\boldsymbol{a}} = \overbrace{\frac{1}{T^2}\begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}}^{\boldsymbol{M}} \overbrace{\begin{bmatrix} y(n) \\ y(n-1) \\ \vdots \\ y(0) \end{bmatrix}}^{\boldsymbol{y}}. \tag{7.3}$$



(a)



(b)

Figure 7.1: Displacement estimation using the three algorithms. (a) Simulated acceleration measurements. Acceleration estimated by fitting a Ricker wavelet to noise-corrupted measurements is almost identical to the actual acceleration. (b) The estimated displacement using the three algorithms.

We can now solve for $\boldsymbol{y}$ by minimizing the mean-squared error between $\boldsymbol{a}$ and the measurements $\boldsymbol{a_m}$ while enforcing the initial and final conditions on the pavement state i.e.

$$\begin{aligned} \boldsymbol{y^*} &= \arg \min_{y|\boldsymbol{Cy}=\boldsymbol{d}} \|\boldsymbol{a} - \boldsymbol{a_m}\|^2, \\ &= \arg \min_{y|\boldsymbol{Cy}=\boldsymbol{d}} \|\boldsymbol{My} - \boldsymbol{a_m}\|^2 \end{aligned} \tag{7.4}$$

This is a least squares problem with linear equality constraints and can be solved using Lagrange multipliers or using standard numerical routines such as *lsqlin* in MATLAB. The initial and final conditions in our case can be set to zero by assuming that pavement is at rest before and after the load is applied. Let the rest period be defined using k samples at the beginning and m samples at the end i.e.

$$C = \begin{bmatrix} I_{k\times k} & 0_{k\times(n+1-k)} \\ 0_{m\times(n+1-m)} & I_{m\times m} \end{bmatrix},$$

$$d = [\underbrace{0\ 0\ \ldots\ 0}_{k+m}]^\top.$$

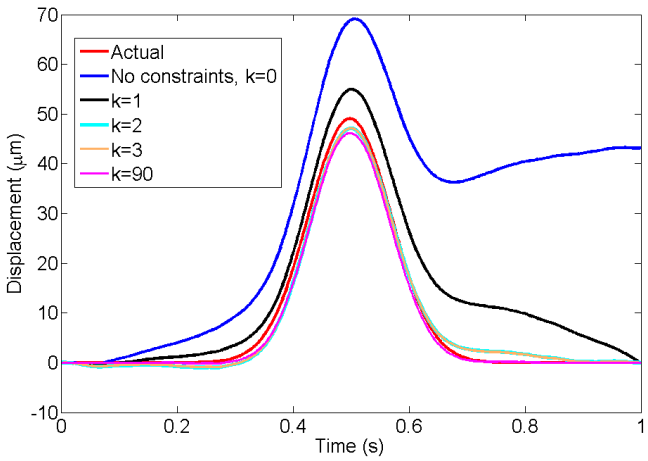


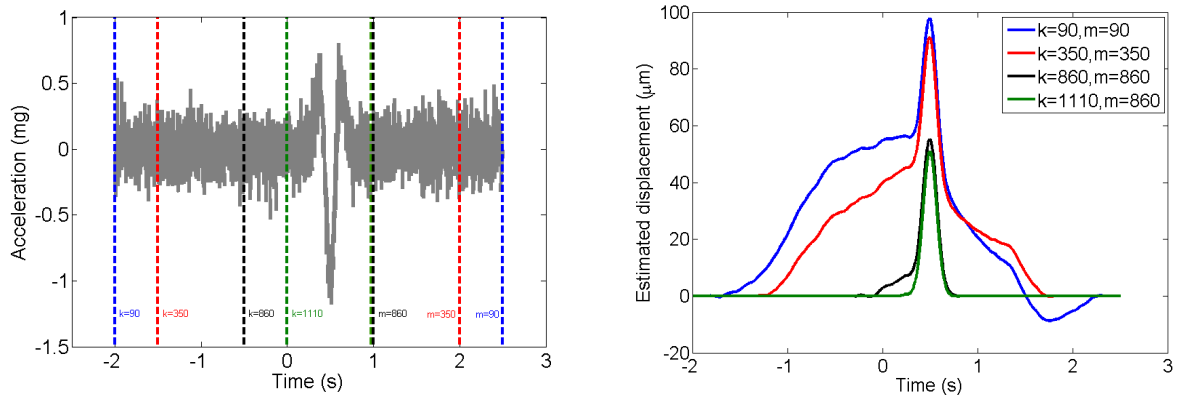
Figure 7.2: Effect of imposed constraints on displacement estimates.

Figure 7.1(b) shows that the displacement estimated using constrained least-squares is more accurate than the one estimated using double integration. Figure 7.2 shows the estimated displacement for different values of k and for convenience we chose $m = k$. The estimates change drastically as k is increased from 0 to 2 but increasing k further does not cause any significant change in the estimate. This can be explained using Equations (7.1) and (7.2). As we increase k from 0 to 3, we progressively add important constraints for pavement's "rest" state. For $k = 1$, the only final condition imposed is $y(N) = 0$ but for $k = 3$, in addition to the constraints on $y(N)$, $y(N-1)$, $y(N-2)$, we also indirectly impose the following constraints on pavement speed and acceleration,

$$
\begin{aligned}
v(N) &= \frac{y(N) - y(N-1)}{T} = 0, \\
v(N-1) &= \frac{y(N-1) - y(N-2)}{T} = 0, \\
a(N) &= \frac{y(N) - 2y(N-1) + y(N-2)}{T^2} = 0.
\end{aligned}
$$

As we increase k beyond 3, we effectively impose the above constraints at the edges of a smaller time-window (by $2\times[k-3]$) samples) and estimated displacement does not change significantly as long as the pavement starts and ends at rest during the new time-window. This algorithm is more robust to measurement errors than double-integration; however, results can be inaccurate if majority of the samples in the new time-window correspond to the pavement at rest. Figure 7.3(a) shows the time-windows for different values of k and Figure 7.3(b) shows the corresponding displacement estimates. Since the displacement estimates are much worse for larger time-windows, the smaller time-window spanning the signal

of interest must be chosen. We can estimate this window by thresholding the measured energy in the signal or by using features like zero-crossings of the measured signal (see Section 6.2.1). Alternatively, a user input can be provided to select the time-window, especially in case of controlled experiments.



(a) (b)

Figure 7.3: Effect of time-window choice on displacement estimates. (a) Simulated acceleration measurements. The dotted-lines of matching color represent the time-windows chosen for displacement estimation. (b) The estimated displacement corresponding to each time-window. The smallest window that encompasses the loading event leads to the best estimate.

7.1.3 Algorithm 3: model-based estimation

This algorithm is motivated from our pavement-vehicle interaction model in Section 5.2. This algorithm is most suitable when the measured response can be represented using a family of functions $f(\kappa, t)$ paramterized by κ. The appropriate family of functions can be chosen based on experimental observations or based on some underlying model for pavement response. The previous two algorithms estimate the pavement displacement directly from the noise-corrupted measured acceleration. This algorithm de-noises the acceleration measurements $a_m(t)$ first and then calculates the displacement $d(t)$ by integrating clean acceleration $a(t)$ twice. Let $a_m(t) = a(t) + \epsilon(t) = f(\boldsymbol{\kappa}, t) + \epsilon(t)$, where $\epsilon(t)$ is the measurement error or noise at time t and $\boldsymbol{\kappa}$ is vector of unknown model parameters. The acceleration $a(t)$ can be estimated as follows,

$$\boldsymbol{\kappa}^* = \arg\min_{\boldsymbol{\kappa}} \|a_m(t) - f(\boldsymbol{\kappa}, t)\|^2, \tag{7.5}$$

$$a(t) = f(\boldsymbol{\kappa}^*, t), \tag{7.6}$$

$$y(t) = y(0) + \int_0^t \int_0^\xi f(\boldsymbol{\kappa}^*, \tau) d\tau d\xi.$$

Assuming that the pavement was at rest at $t = 0$ i.e. $y(0) = 0$, we have

$$y(t) = \int_0^t \int_0^\xi f(\boldsymbol{\kappa}^*, \tau) d\tau d\xi. \tag{7.7}$$

Figure 7.1 compares the estimated displacement for the three algorithms. The estimate using Algorithm 3 is the most accurate but this was partly because we used the correct function family to model pavement acceleration. We chose $f(\kappa, t)$ to be the family of Ricker wavelets (Mexican-hat functions) with the same three parameters used in Chapter 6. Algorithm 3 is also the least sensitive to the chosen time-window and can decouple the response for successive axle loads as explain in previous chapters.

7.2 Experimental results

We report the results for three different types of applied loads: the FWD, the HVS, and real truck axles.

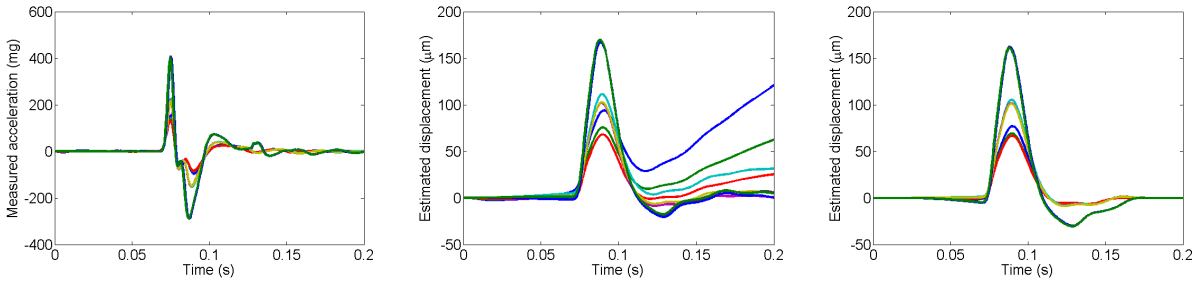
7.2.1 FWD tests

Loads are repeatedly applied at different locations relative to the sensor. The measured acceleration for different loads is shown in Figure 7.4 and also shown are the displacements estimated by Algorithm 1 and 2. As expected, Algorithm 1 suffers from low-frequency drift and is outperformed by Algorithm 2.

Figure 7.5(a) compares the estimated displacements with surface displacements measured by the FWD geophones. The estimates are highly correlated ($\rho = 0.998$) with the reference measurements. Figure 7.5(b) shows the peak estimated displacement at different locations relative to the applied load. As expected, the measured response decreases exponentially when the distance between the sensor and the load increases.

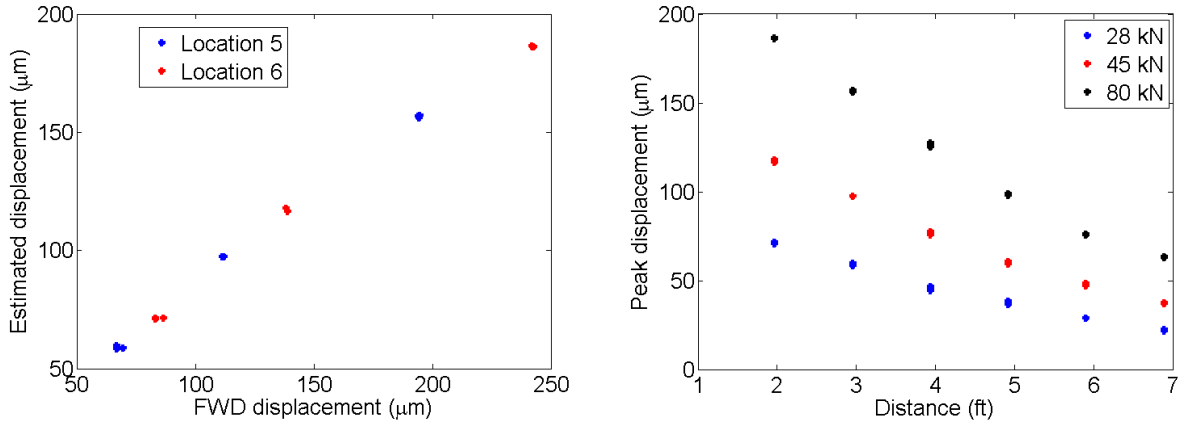
7.2.2 HVS tests

The heavy vehicle simulator was used to apply different loads to the pavement at a speed of 5.41 *mph*. Figure 7.6 shows the average pavement acceleration obtained by averaging 20 repetitions of the same load and also shown are the pavement displacements estimated



(a) (b) (c)

Figure 7.4: Comparison of displacement estimates for FWD tests. Loads of 25 kN, 45 kN and 80 kN are applied to the pavement and repeated 3 times each. (a) Measured pavement acceleration. (b) Displacement estimated using Algorithm 1. (c) Displacement estimated by Algorithm 2.



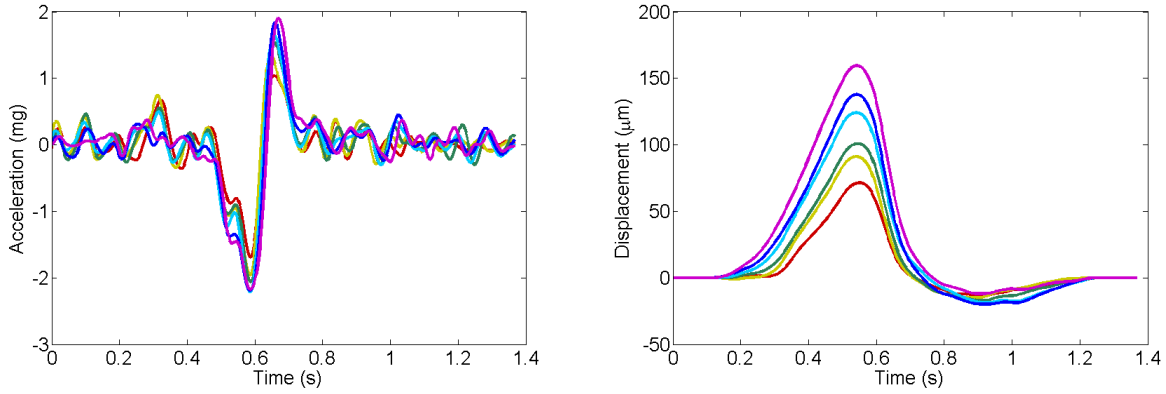
(a) (b)

Figure 7.5: Estimated peak displacement for FWD tests (a) The estimated peak displacement is highly correlated to the surface displacement measured by FWD geophones. (b) The peak displacement decreases as the distance between the sensor and the load increases.

using Algorithm 2. The shape of the response is very similar for different loads and the peak displacement is larger for heavier loads as expected.

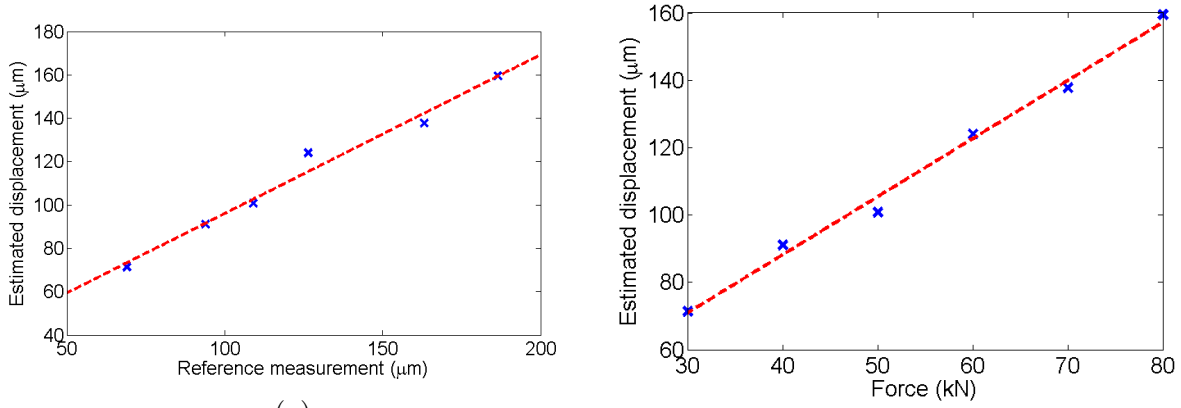
Figure 7.7(a) compares the displacement estimates with surface displacement measurements from a reference LVDT sensor and there is high correlation ($\rho = 0.99$) between the two. It should be noted that displacement measured by the vibration sensor is expected to be smaller than the surface deflection since the sensor is embedded inside the pavement. Consistent with pavement models, the estimated displacement is shown to be linearly proportional to

applied load in Figure 7.7(b).



(a) (b)

Figure 7.6: Displacement estimates for HVS tests. (a) The average pavement acceleration obtained from averaging measurements for 20 HVS runs at each load level. (b) Estimated pavement displacement for each load.



(a) (b)

Figure 7.7: Verification of displacement estimates for HVS tests. (a) Comparison of estimated peak displacement with the surface displacement measured by a LVDT sensor. (b) Estimated pavement displacement increases linearly with applied force.

7.2.3 Real truck loads on highways

We follow the same model described for load estimation and assume that

$$\boldsymbol{\tau} = [\alpha \ \mu \ \sigma]^\top,$$
$$f(\alpha, \mu, \sigma, t) = \alpha \left[1 - \frac{(t-\mu)^2}{\sigma^2}\right] e^{-\frac{(t-\mu)^2}{2\sigma^2}}.$$

The Ricker wavelet $f(\alpha, \mu\,\sigma, t)$ models the acceleration corresponding to each axle and the response due to entire truck is the sum of individual axle responses. The model parameters can be estimated using the curve-fitting procedure described in Section 5.3. The pavement displacement can then be estimated using Equation (7.7) of Algorithm 3.

Figure 7.8 compares the estimated displacement for a two-axle truck using Algorithm 2 and 3. The shape of the estimated displacements is similar but the peaks corresponding to the second axle differ by $\approx 15\%$.

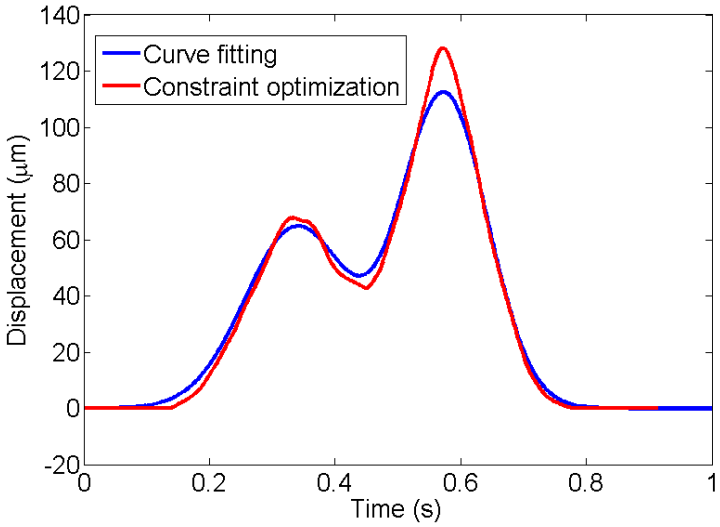


Figure 7.8: Comparison of estimated displacement for a two-axle truck using Algorithm 2 and 3.

7.3 Discussion

Algorithm 2 does not require any prior knowledge about the measured pavement response and is therefore more suitable for researching new pavement materials and structures. This algorithm is less robust to measurement noise but errors can be minimized by carefully

choosing the time-window for considered measurements. Thus, it is ideal for experiments where the time, speed, and magnitude of applied loads are controlled and monitored.

Algorithm 3 assumes a model for pavement response and error in estimated displacement can be high if the model is inaccurate. However, a more accurate model can be developed by using estimates from Algorithm 2. A major advantage of using Algorithm 3 is that the displacement estimates do not suffer from low frequency drifts as noise is removed from measured acceleration before integrating it twice. The algorithm is also more suitable long-term pavement monitoring as the assumed model can be used to develop efficient compression algorithms and reduce the amount of data sensor needs to transmit. An example of how this concept is used to extend sensor lifetime is given in Section 6.2.2.

For long-term pavement monitoring, vibration sensors can be used to report compressed acceleration data when a loaded axle is detected. The AP can average the data from multiple sensors and estimate the pavement displacement (similar to Section 6.2.3). The data can be reported to a server for future analysis.

Chapter 8

Conclusions and Future Work

This chapter concludes this dissertation by summarizing the major contributions of this work and suggesting directions for future work.

8.1 Summary of contributions

Existing weigh-in-motion (WIM) technologies are too expensive for widespread deployment and there is insufficient data for pavement management, resulting in higher maintenance costs. The goal of this project was to develop an inexpensive but accurate weigh-in-motion system that can be easily installed on new or existing roads. Current WIM systems have high costs because they use large, expensive load sensors and require special pavement construction around them. Motivated to reduce costs, we decided to use the road pavement itself as our weighing scale. Multiple wireless sensors are embedded in the pavement to sense its response to passing vehicles and the measured response is then used to estimate the magnitude of applied axle loads. The challenges involved in enabling this concept are discussed in Section 1.2.2 and the rest of the thesis describes in detail how these were overcome.

As a part of this work, we built a prototype wireless WIM system and tested it under real highway traffic to show that it meets the required accuracy standards (LTPP specification [3]). The wireless WIM also outperformed a commercial WIM station using conventional technology. It can be considered a final product once the following two tasks are completed:

- Increase sensor lifetime by doubling the battery or by upgrading the MEMS accelerometer to the latest low-power device (Section 6.2.4).
- Implement adaptive sampling procedure on the sensor nodes (Section 6.2.1), and test the resulting system under different traffic, climate, and pavement types.

We now summarize the major contributions of this work.

Wireless sensor-based platform for intelligent transportation and pavement monitoring

We built and tested a platform consisting of wireless vibration sensors, vehicle detection sensors, and a roadside access point (AP) that controls the sensors and processes incoming data from them. A camera is installed with the AP to capture and record pictures of ongoing traffic. The AP is time synchronized using Network Time Protocol (NTP) and all other network devices are synchronized to it. The platform is used to detect vehicles, estimate their speed, capture roadside traffic images, and collect pavement vibration and temperature data corresponding to each vehicle. As shown in this dissertation, the platform can be used for many new applications in intelligent transportation and pavement engineering. The wireless communication protocol, based on Sensys NanoPower protocol [31], was tested on highways with live traffic and high reliability ($\geq$ 99%) of data transmission was achieved (Section 2.3.4). The sensor firmware can be updated over the air, if needed.

Wireless vibration sensor for measuring road vibrations

Road pavements are designed to experience very small vibrations in response to heavy loads driving over them. We designed, built, and tested a new wireless sensor that can be embedded in the road to measure the transient vibrations due to different applied loads. As discussed in Chapter 2, the sensor also measures pavement temperature which is very important for modeling and load estimation. The sensor has very low measurement noise (typically $\leq$ 400 ug) and is practically immune to traffic noise (sound) that is generally picked up by MEMS accelerometers. The pavement response measured by the sensor is highly repeatable under similar conditions (0.1% - 3.4% for the most applied loads). The sensor can last more than 6 months when all raw acceleration data, sampled at a high rate of 512 Hz, is continuously reported to the AP. We showed that the lifetime can be increased for many applications by using techniques such as in-sensor signal processing and adaptive rate subsampling. Using these techniques, the sensor can last more than two years for WIM and automatic vehicle classification (AVC).

System modeling and application-specific data processing solutions

We used the above platform for solving three problems: AVC, WIM, and displacement estimation. These are formally described in Problem 1.2, 1.3, and 1.4.

The number of axles and the inter-axle spacings of a vehicle can be used to accurately classify it [33]. In Chapter 4, we present our solution for automatic vehicle classification via axle detection. Our model assumes that each axle causes a burst of energy in measured vibrations. A data processing algorithm is presented which uses acceleration data from multiple sensors to detect individual axles of each passing vehicle. The solution was tested using 53 random trucks on a highway, and axles for all the trucks were successfully detected. This algorithm forms the basis of the energy-efficient axle detection algorithm.

In Chapter 5 and 6, we present our solution to the weigh-in-motion problem. In Chapter 5, we propose a new model that relates pavement acceleration to moving axle loads. It is based on the Euler beam model for pavements [52] and we confirm important results of the model in Observations 5.1- 5.5. The model can be summarized by the following equation (see Section 5.2 for details):

$$F = \beta \frac{\alpha}{v^2} \tau(T). \tag{8.1}$$

The parameter α is the peak acceleration caused due to each individual axle. However, the individual response due to closely-spaced axles needs to be decoupled and estimated from the measured response. To accomplish this, we propose a parametric model for the measured response and estimate the model parameters by minimizing the error between the measured and the modeled response. The initial parameters must be chosen carefully in order to ensure that solution to this nonlinear optimization problem corresponds to the global minimum error. We choose these initial parameters based on signal features and vehicle speed. The estimated α's are combined with vehicle speed and pavement temperature to calculate individual axle loads. To test the model and the load estimation algorithm, we used 75 randomly chosen trucks that were stopped and weighed individually at a weigh station. The above linear model fits the data with a R^2-value of 0.99, as shown in Figure 5.9(a).

In Chapter 6, we make improvements to our model by optimizing the effect of vehicle speed on pavement response and choosing a different error measure (LTPP error) for parameter estimation. To benchmark the accuracy of wireless WIM, we used random sub-sampling cross-validation. In each of the 1000 cross-validation trials, 25 truck runs with known axle weights are used to calibrate the system and remaining 50 trucks are used for determining predictive accuracy. The accuracy levels passed the LTPP specifications and errors were found to be less than 12.5% for individual axles and 9.5% for total weight.

In Chapter 7, we present the problem of displacement estimation from noisy acceleration measurements. Two new algorithms were developed for different uses: Algorithm 2 for controlled experiments (eg. FWD, HVS), and Algorithm 3 for real-time pavement monitoring. Algorithm 2 estimates displacement by discretizing the relationship between acceleration and displacement, and solving a constrained least-squares problem. This algorithm is more suitable for controlled experiments where initial and final pavement conditions are easy to obtain. Algorithm 3 should be used when the measured response can be represented by a family of functions with unknown parameters. For example, the measured displacement due to each loaded axle is assumed to be a Gaussian function with unknown amplitude and variance. Just like the axle weights, pavement displacement can be recorded in real-time for each and every passing truck. The estimated displacements are highly correlated to measurements from reference instruments, see Section 7.2 for details.

Energy-efficient algorithms for different applications

Enabling the above solutions using wireless sensors requires that the algorithms reduce the computational and data transmission load on each sensor in order to extend its lifetime. We use techniques like in-sensor feature extraction, adaptive rate subsampling, and distributed computation to reduce the power consumption of sensor nodes at the expense of the access point.

In Section 4.2.2, we present an in-sensor feature extraction algorithm that reports the amplitude and timestamp for each detected axle peak. The AP combines data from multiple sensors to estimate the axle count and axle spacings for each passing vehicle. Thus, in comparison to the *raw data* mode, the amount of data transmitted is reduced by more than 1000 times. This results in huge power savings and extending the battery life from months to years.

In Chapter 6, we use a similar in-sensor feature extraction algorithm to detect each individual axle, and use adaptive rate sampling around each peak to reduce the amount of data transmitted for load estimation. The AP reconstructs the measured signal for each sensor from these strategically chosen samples, and combines measurements from multiple sensors to estimate axle weights and gross weight of the truck. The details can be found in Section 6.2.3. Using these techniques, the sensor lifetime increased from 6 months to 2 years without any degradation to system accuracy. The error (LTPP) in estimated weight is typically less than 12% for individual axles and 9.1% for the total. The wireless WIM proposed here outperformed a nearby in-service WIM station whose errors were as high as 24% for individual axles and 13% for gross weight.

The same algorithm for wireless WIM also estimates in real-time the pavement displacement corresponding to each on-going axle and truck. This information can be combined with weight information to estimate the strength of the pavement [2]. Thus, unlike conventional WIM systems, the wireless WIM can be used as a tool for regular assessment of pavement condition.

Other contributions

An analytical solution was derived for determining the optimal scaling coefficients to combine noisy measurements from multiple arrays in Chapter 6. The solution is used for calibration of $\boldsymbol{\beta}$ by minimizing the LTPP error, a system accuracy measure for WIM systems. The LTPP error includes both the bias and the variance terms of system error and the result could be useful in many other applications.

8.2 Future Work

We tested the prototype system for AVC, WIM, and displacement estimation but more engineering work is required before it can considered a commercial product. We suggest some possible directions to extend the current work.

Automatic Vehicle Classification

The AVC system presented in this dissertation has a lifetime of only two years. However, majority of current consumption is due to the low noise MEMS accelerometer that we are using. In Section 4.2, we increased the density of sensors across the width of the lane and made sure that at least one of them will be a tire-on-top sensor. The amplitude measured by tire-on-top sensors is much higher and does not require such low noise accelerometers. The analog MEMS sensor on the board was replaced by a low power consuming digital accelerometer and is being tested by Sensys Networks in a commercial setting. The system provides a wireless alternative to current technologies; an alternative that is very accurate, easy-to-install, and cheaper than existing technologies.

Wireless Weigh-In-Motion

While the prototype system outperformed a commercial WIM station in terms of accuracy, it needs improvement in other areas. The sensor lifetime, currently 2 years for this application, needs to be at least doubled. As suggested in Chapter 6, this can be done easily by doubling the sensor battery. An alternative is to replace the MEMS accelerometer with one that has a similar noise floor but a lower current consumption [23].

The wireless WIM algorithm, presented in Section 6.2, was tested by offline application of algorithm on the raw sensor data. The algorithm needs to be implemented and tested on sensor nodes. Some other unanswered questions are:

- How does the system perform for other classes of vehicles? Is the calibration parameter β different for different classes? If yes, this can make the calibration procedure more complex and expensive as more vehicles will be required.
- How does the system accuracy change with pavement deterioration? The answer to this question determines how often calibration is required for the system. We tested the system on 4 different days, spanning from February to September, and observed no significant change in accuracy.
- Can we improve the accuracy further by adding additional arrays of vibration sensors? We showed in Figure 6.2(a) that increasing sensor arrays from 1 to 4 progressively reduces the errors in load estimates. Some European WIM specifications require the errors to be as low as 5%, and adding sensor arrays is one possible way of achieving this.

- Can we quantify the effect of vehicle wander (driving off the lane center) on load estimates? This requires a method to accurately determine the location of the wheels relative to the sensors. If possible, this can also help reduce the weight errors.

The wireless WIM technology presented here has the potential to replace the current WIM systems at less than one-tenth of cost. The system can also be used as a Prepass WIM near static weigh stations. Prepass stations estimate the weight of trucks in motion and direct them to the weigh station if they are *likely* to be overweight. If truckers can be incentivized to carry wireless tags for automatic identification, more intelligence can be added to the system. For example, anytime a truck with a tag is flagged for static weighing, the data obtained from the weigh station can be used to automatically re-calibrate the wireless WIM. Finally, as discussed below, the wireless WIM can provide important pavement monitoring data that existing technologies are currently incapable of.

Displacement Estimation

The system presented here has many applications in pavement engineering and opens up new research avenues. The vibration sensor, combined with the algorithms, can be easily embedded in new or existing pavements and used as a wireless displacement sensor. Algorithm 2 is ideal for controlled research experiments and can be used to test new materials and structures for pavements. Once a material is characterized, the modeled response can be incorporated in Algorithm 3 to estimate pavement deflection in real-time for each passing vehicle.

Monitoring pavement displacements and loads simultaneously provides much more valuable data than just weight information. Each time a truck with known weight drives over the system, the displacement measured by different sensors can be used to backcalculate the pavement stiffness [2]. This provides a much cheaper way of assessing current pavement condition than using the typical FWD test. Further testing needs to be done in order to determine the accuracy of estimated stiffness. Finally, the sensor data can possibly be combined with damage detection algorithms to detect pavement damage at an early stage.

Road to smart pavements

The concept of smart pavements is a futuristic concept where small wireless sensors are used to monitor pavements and detect structural damage at an early stage, thereby reducing the maintenance costs and extending pavement lifetime. Currently, pavement instrumentation for condition monitoring is done on a localized and short term basis, mainly for research purposes. The current technology does not allow for continuous long-term monitoring because of many limitations.

The wireless sensors described in this work can be used for this purpose but some improvements need to be made. We built and tested sensors that are 2.5 $inch^3$ in size and cost a

few hundred dollars. As power consumption is reduced, smaller battery can be used and sensor size can be reduced to 1 $inch^3$. As cost of sensors comes down (or using a cheaper alternative), the density of these sensors on roads can be increased for monitoring the entire road network. This can enable pavements to monitor their own structural health and request repair or assistance when any damage is detected.

Other applications

The platform can be used in many other applications. We list a few interesting ones here:

- **Rail WIM:** Just like roads, vibrations sensors can be attached to rail ties in order to measure vibrations on the rail track when a train goes by. These vibrations can be used to detect and weigh individual train cars. If possible, the system can provide an alternative to current rail WIM systems that are much more expensive.
- **Footstep detection in buildings:** We used the vibration sensors for detecting footstep traffic in buildings. Each footstep is an impulsive load applied to floor and the corresponding vibrations are used for detecting each step. The algorithm used is very similar to the axle detection algorithm. We found that coverage of a single vibration sensor can be up to 5-10 feet, except for ground floors that have a stronger base and produce almost no vibrations to footsteps. As a bonus, the sensor also measure room temperature. Multiple sensors in a building can be used to estimate which areas are populated and this information can be used to control the lighting, HVAC etc to minimize the energy footprint. Another application is intrusion detection and data reported from sensors can be incorporated into the building's security system.
- **Structural health monitoring (SHM):** SHM involves continuous measurement of a structure's response, extraction of damage-sensitive features from these measurements, and statistical analysis of these features to determine the current state of system health. The underlying concept is similar to *smart pavements* [7] but has been used for other civil infrastructure like bridges and buildings, for example in [36, 35, 17].
- **Wireless seismic monitoring:** The high resolution vibration sensor is a natural candidate for seismic monitoring. It can be used to detect even the lowest magnitude earthquakes. The AP can analyze the data reported from sensors and trigger alarms when interesting seismic events happen. The fact that the sensors are wireless makes it very cheap and convenient to scale the sensor network and monitor a large area. Seismic monitoring is also used for oil exploration as discussed in [39].

Bibliography

[1] Guide AASHTO. *Guide for Design of Pavement Structures*, 1993.

[2] Federal Highway Administration. Instrumentation for flexible pavements-field performance of selected sensors. Technical Report FHWA-RD-91-094, US Department of Transportation, June 1992.

[3] Federal Highway Administration. Long-term pavement performance program protocol for calibrating traffic data collection equipment. Technical report, US Department of Transportation, 1998.

[4] Federal Highway Administration. Long-term pavement performance information management system: Pavement performance database user reference guide. Technical Report FHWA-RD-03-088, US Department of Transportation, 2003.

[5] Federal Highway Administration. Federal size regulations for commercial motor vehicles, 2004.

[6] Federal Highway Administration. Long-term pavement performance program: Accomplishments and benefits 1989-2009. Technical Report FHWA-HRT-10-072, US Department of Transportation, 2010.

[7] Federal Highway Administration. Smart pavement monitoring system. Technical Report FHWA-HRT-12-072, US Department of Transportation, 2013.

[8] M. Arraigada, M. Partl, S. Angelone, and F. Martinez. Evaluation of accelerometers to determine pavement deflections under traffic loads. *Materials and Structures*, 42:779–790, 2009.

[9] Ravneet Bajwa. Weigh-in-motion system using a mems accelerometer. Technical report, EECS Department, University of California, Berkeley, 2009.

[10] Edward S Barber. Calculation of maximum pavement temperatures from weather reports. *Highway Research Board Bulletin*, 1957.

[11] I.A. Barbour and W.H. Newton. Multiple-sensor weigh-in-motion. In *First European conference on weigh-in-motion of road vehicles*, 1995.

[12] Transportation Research Board. Guide for mechanistic-empirical design of new and rehabilitated pavement structures. Technical Report 1-37A, National Research Council, March 2004.

[13] Transportation Research Board. Pavement lessons learned from the aasho road test and performance of the interstate highway system. Technical Report E-C118, National Research Council, July 2007.

[14] Piotr Burnos, Janusz Gajda, Piotr Piwowar, Ryszard Sroka, Marek Stencel, and Tadeusz Żegleń. Accurate weighing of moving vehicles. *Metrology and Measurement Systems*, 14(4):507–516, 2007.

[15] R. Bushman and A.J. Pratt. Weigh in motion technology–economics and performance. In *Presentation on the North American Travel Monitoring Exhibition and Conference (NATMEC). Charlotte, North Carolina*, 1998.

[16] D. Cebon. *Handbook of Vehicle-Road Interaction*. Swets and Zeitlinger Publishers, 1999.

[17] M. Ceriotti, L. Mottola, G.P. Picco, A.L. Murphy, S. Guna, M. Corra, M. Pozzi, D. Zonta, and P. Zanon. Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment. In *Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*, pages 277–288. IEEE, 2009.

[18] M Cherniakov, RSA Raja Abdullah, P Jancovic, and M Salous. Forward scattering micro sensor for vehicle classification. In *Radar Conference, 2005 IEEE International*, pages 184–189. IEEE, 2005.

[19] Sing Yiu Cheung, Sinem Coleri, Baris Dundar, Sumitra Ganesh, Chin-Woo Tan, and Pravin Varaiya. Traffic measurement and vehicle classification with single magnetic sensor. *Transportation Research Record: Journal of the Transportation Research Board*, 1917:173–181, 2006.

[20] Coibrys, Inc, accelero.us@colibrys.com. *MEMS Capacitive Accelerometers Product Description MS9000.D*, 2010.

[21] Erdem Coleri and John T Harvey. Evaluation of laboratory, construction, and performance variability by bootstrapping and monte carlo methods for rutting performance prediction of heavy vehicle simulator test sections. *Journal of Transportation Engineering*, 137(12):897–906, 2011.

[22] Erdem Coleri, Bor-Wen Tsai, and Carl L Monismith. Pavement rutting performance prediction by integrated weibull approach. *Transportation Research Record: Journal of the Transportation Research Board*, 2087(1):120–130, 2008.

[23] Colibrys, Neuchâtel, Switzerland. *MS7002 Datasheet*, 2013.

[24] Colibrys, Inc, Neuchâtel, Switzerland. *MEMS Capacitive Accelerometers Datasheet MS9000.D*, 2010.

[25] Barry J Dempsey and Marshall R Thompson. A heat transfer model for evaluating frost action and temperature-related effects in multilayered pavement systems. *Highway Research Record*, 1970.

[26] M.F. Duarte and Y. Hen Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, 2004.

[27] Federal Highway Administration, U.S. Department of Transportation. *Traffic Monitoring Guide*, May 2001.

[28] Michael C. Fehler. *Seismic Wave Propagation and Scattering in the Heterogenous Earth*. Springer, 2nd edition, 2009.

[29] Mayrai Gindy, Hani H Nassif, and Jana Velde. Bridge displacement estimates from measured acceleration records. *Transportation Research Record: Journal of the Transportation Research Board*, 2028(-1):136–145, 2007.

[30] S. Gupte, O. Masoud, R.F.K. Martin, and N.P. Papanikolopoulos. Detection and classification of vehicles. *Intelligent Transportation Systems, IEEE Transactions on*, 3(1):37–47, 2002.

[31] A. Haoui, R. Kavaler, and P. Varaiya. Wireless magnetic sensors for traffic surveillance. *Transportation Research C*, 16(3):294–306, 2008.

[32] Charles Harlow and Shiquan Peng. Automatic vehicle classification system with range sensors. *Transportation Research Part C: Emerging Technologies*, 9(4):231 – 247, 2001.

[33] Robert I. Stevens John H. Wyman, Gary A. Braley. Evaluation of fhwa vehicle classification categories. Technical report, Federal Highway Administration, 1984.

[34] Holger Karl and Andreas Willig. *Protocols and architectures for wireless sensor networks*. Wiley-Interscience, 2007.

[35] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, and Martin Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *IPSN '07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 254–263, New York, NY, USA, 2007. ACM.

[36] Anne S Kiremidjian. Time series based structural damage detection algorithm using gaussian mixtures modeling. *Journal of dynamic systems, measurement, and control*, 129:285, 2007.

[37] Ara N. Knaian. A wireless sensor network for smart roadbeds and intelligent transportation systems. Master's thesis, Massachusetts Institute of Technology, 2000.

[38] Karric Kwong, Robert Kavaler, Ram Rajagopal, and Pravin Varaiya. Arterial travel time estimation based on vehicle re-identification using wireless magnetic sensors. *Transportation Research Part C: Emerging Technologies*, 17(6):586–606, 2009.

[39] Jerome Laine and Denis Mougenot. Benefits of mems based seismic accelerometers for oil exploration. In *Solid-State Sensors, Actuators and Microsystems Conference, 2007. TRANSDUCERS 2007. International*, pages 1473–1477. IEEE, 2007.

[40] M. Li and Y. Liu. Underground structure monitoring with wireless sensor networks. In *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 69–78. ACM, 2007.

[41] CR Liu, L Guo, J Li, and X Chen. Weigh-in-motion (wim) sensor based on em resonant measurements. In *Antennas and Propagation Society International Symposium, 2007 IEEE*, pages 561–564. IEEE, 2007.

[42] Yean-Jye Lu, Yuen-Hung Hsu, and Xavier Maldague. Vehicle classification using infrared image analysis. *Journal of transportation engineering*, 118(2):223–240, 1992.

[43] Jerome P Lynch and Kenneth J Loh. A summary review of wireless sensors and sensor networks for structural health monitoring. *Shock and Vibration Digest*, 38(2):91–130, 2006.

[44] Robert L Lytton. Backcalculation of pavement layer properties. *Nondestructive testing of pavements and backcalculation of moduli, ASTM STP*, 1026:7–38, 1989.

[45] Ramesh B Malla, Amlan Sen, and Norman W Garrick. A special fiber optic sensor for measuring wheel loads of vehicles on highways. *Sensors*, 8(4):2551–2568, 2008.

[46] CI Merzbacher, AD Kersey, and EJ Friebele. Fiber optic sensors in concrete structures: a review. *Smart materials and structures*, 5:196, 1996.

[47] A.Y. Nooralahiyan, M. Dougherty, D. McKeown, and H.R. Kirby. A field trial of acoustic signature analysis for vehicle classification. *Transportation Research Part C: Emerging Technologies*, 5(3-4):165–177, 1997.

[48] Texas Department of Transportation. *Traffic Recorder Instruction Manual*. Texas DOT, February 2012. Appendix A.

[49] Dong-Yeob Park, Neeraj Buch, and Karim Chatti. Effective layer temperature prediction model and temperature correction via falling weight deflectometer deflections. *Transportation Research Record: Journal of the Transportation Research Board*, 1764(1):97–111, 2001.

[50] D. Parl, N. Buch, and K. Chatti. Effective layer temperature prediction model and temperature correction via falling weight deflectometer deflections. *Transportation Research Record: Journal of the Transportation Research Board*, 1764:97–111, 2001.

[51] Jorge Alberto Prozzi. *Modeling pavement performance by combining field and experimental data*. PhD thesis, University of California, Berkeley, 2001.

[52] Ram Rajagopal. *Large Monitoring Systems: Data Analysis, Deployment and Design*. PhD thesis, University of California, Berkeley, 2009.

[53] H. Sawant, Jindong Tan, and Qingyan Yang. A sensor networked approach for intelligent transportation systems. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 2, pages 1796 – 1801 vol.2, 2004.

[54] Boutros Sebaaly, Trevor G. Davis, and Michael S. Mamlouk. Dynamics of falling weight deflectometer. *Journal of Transportation Engineering*, 111(6):618–632, 1985.

[55] Sensys Networks, Inc, 2650 9th Street, Berkeley CA 94710. *The Sensys Wireless Vehicle Detection System*, 1.1 edition.

[56] Silicon Designs, Inc. *Model 1221 Low Noise Analog Accelerometer*, 2010.

[57] S. W. Smith. *The scientist and engineer's guide to digital signal processing*. California Technical Publishing, San Diego, CA, 1997.

[58] JB Sousa, JA Deacon, S Weissman, JT Harvey, CL Monismith, RB Leahy, G Paulsen, and JS Coplantz. Permanent deformation response of asphalt-aggregate mixes. *Rep. No. SHRP-A*, 415, 1994.

[59] Andrei Nikolaevich Tikhonov. Regularization of incorrectly posed problems. In *Soviet Math. Dokl*, volume 4, pages 1624–1627, 1963.

[60] Fuming Wang and Robert L Lytton. System identification method for backcalculating pavement layer properties. *Transportation Research Record*, 1993.

Appendix A

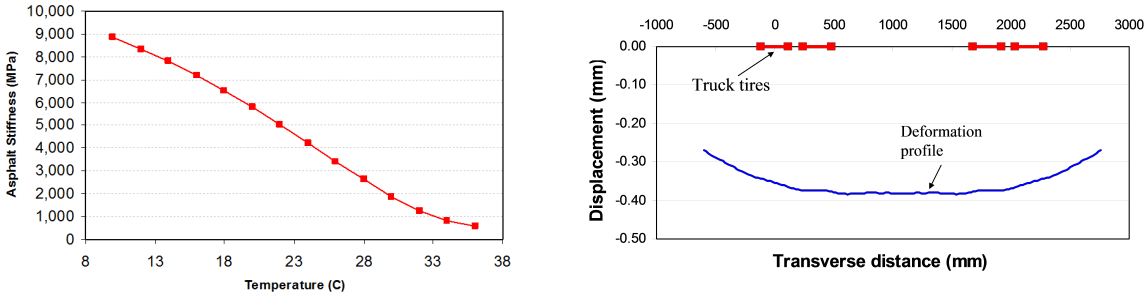
Temperature Compensation

Asphalt concrete is a composite engineering material which consists of aggregates, air-voids and asphalt mastic. Due to the viscoelastic nature of asphalt mastic, composite asphalt concrete mixtures show a viscoelastic response when subjected to traffic loads. Temperature dependent behavior of asphalt concrete layer needs to be modeled in pavement engineering to determine the changes in pavement response with temperature [10, 22, 25, 49, 58]. Since the vibration sensors were embedded in asphalt concrete layer of the pavement, a temperature compensation model for this layer is developed using the following steps:

1. Determine the pavement layer thicknesses for the section where vibration sensors were installed.
2. Determine the typical unbound layers' stiffnesses from previous falling weight deflectometer (FWD) test results [21] and assume a typical concrete layer stiffness (35 GPa).
3. Determine the asphalt layer stiffnesses for asphalt materials used in California for temperatures ranging from 10°C to 40°C. The change in stiffness with temperature for one of the asphalt material types is given in Figure A.1(a).
4. Create a layered elastic theory (LET) model for a typical dual-tandem axle (8 tires) on the pavement with structural properties determined in steps 1 to 3.
5. Using the developed LET model, determine the deformations directly under the center of the dual tandem axle for all asphalt materials in the given temperature range. Deformation profile for one of the asphalt material types at 24°C is given in Figure A.1(b).
6. Normalize the deformation versus temperature curves by dividing all deformations by the deformation at 25°C (reference temperature). This leads to a unique temperature compensation function $\tau(T)$ for each asphalt type.

7. Determine the asphalt material type by using $\tau(T)$ for each material to estimate the weight of calibration trucks. The $\tau(T)$ for the correct asphalt material is assumed to minimize errors in load estimates.

Alternatively, a data-driven temperature compensation model can be developed by measuring pavement response for the same applied load at different temperatures. For example, a calibration truck can be driven over the system multiple times on a day with a temperature range of $\geq 10°C$.



(a) (b)

Figure A.1: Temperature compensation for asphalt layer. (a) Change in asphalt stiffness with temperature. (b) Deformation profile for one of the asphalt material types at 24°C.