# Data-Driven Approaches for Track Monitoring Using In-Service Trains

George Lederman

June 2016

Submitted in partial fulfillment of the requirements for
the degree of
Doctor of Philosophy
in
Advanced Infrastructure Systems

Carnegie Mellon University
Pittsburgh, PA

# Abstract

This thesis explores data-driven approaches for monitoring rail-infrastructure from the dynamic response of a train in revenue-service. Presently, track inspection is performed either visually or with dedicated track geometry cars. Collecting and analyzing vibration data from in-service trains can offer more economical and more reliable monitoring. The high frequency with which in-service trains travel each section of track means that faults can be detected sooner than with dedicated inspection vehicles, and the large number of passes over each section of track makes a data-driven approach statistically feasible.

Developing such a data-driven approach requires modeling the state of the tracks from the collected data, then detecting track anomalies as the model changes over time. Building consistent models from different passes is challenging due to the variation in the train's speed from pass to pass, the uncertainty in the train's position, and changes in the properties of the train itself.

We study two ways of modeling the state of the tracks to address these challenges: explicit models where the track profile itself is estimated, and implicit models, where features extracted from the collected data are used to imply information about the tracks. In addition, we explore change detection methods appropriate for both modeling approaches; these would allow for monitoring to occur without human supervision.

Finally, for network-level monitoring to be practical, we study how data from multiple sensors and multiple trains could be fused together. Data fusion could enable more accurate representations of the state of the tracks, and more rapid detection of track changes after they occur.

The track modeling, change detection and data fusion approaches presented in

this thesis are validated with simulations and with data collected from two instrumented trains. This collected data includes more than 500 passes through a 40km rail network over a three year period. We demonstrate that the proposed sensing, signal processing, and data analysis can detect numerous types of track anomalies and could facilitate safer, more efficient rail-infrastructure in the future.

# Acknowledgments

Thanks first and foremost to my advisors: to Jim Garrett, who made me excited about Carnegie Mellon before I had even set foot on campus and remained a strong advocate of the project even as dean, to Hae Young Noh, for her emphasis on clear communication and for fostering a collegial culture within her group, and to Jacobo Bielak, for his support, guidance and encouragement throughout the PhD.

Thanks as well to the other committee members including Mario Bergés, whose thesis was a model for this work, and to Jelena Kovačević, whose advice and guidance made this project truly interdisciplinary.

Thanks to Allen Biehler, who facilitated the collaboration with the Port Authority and first suggested that we study track-issues; to Susan Finger, who helped with my NSF application; to Christoph Mertz, who woke up at ungodly hours to help with experiments in the parking garage; and to Yoshinobu Oshima, who, by jumping in the light-rail car, demonstrated how to test its fundamental frequency. I am also grateful to the staff who facilitated the project including Jules Krishnamurthy, Stan Caldwell and Courtney Ehrlichman.

Thanks to the many helpful Port Authority employees including Bill Miller, Bruno Sinopoli, Jim Porter, Chris van Dyke, Rich Klinger, Gary Diethorn, Dave Cousins, and Bayard Galbraith. I am particularly grateful for the help of David Kramer who advocated for this research project through thick and thin and served as the project's champion within the Port Authority. His support and mentorship improved the research and ensured we addressed the most pressing issues for transit agencies.

Thanks to my colleagues, Zihao Wang, Andrew Thorsen, and, in particular, Siheng Chen. Siheng's deep knowledge of signal processing and his intense intellec-

tual curiosity made working together a pleasure.

The work was supported by the National Science Foundation through a Graduate Research Fellowship under Grant No. 0946825, by National Science Foundation awards 1130616 and 1017278, and a University Transportation Center grant (DTRT12-G-UTC11) from the US Department of Transportation.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Context

We live in an age of data. 27.9 million road-miles have been captured in Google Maps [1]. 300 hours of video are uploaded to YouTube every minute [58]. And Boeing's new 787 Dreamliner produces 500 GB of performance data every flight [24].

The amount of data collected has been enabled by low-cost data-acquisition systems and abundant data-storage. At the same time, innovations in analysis techniques, including signal processing, data mining and machine learning, have facilitated the extraction of valuable insights from this data.

Not every field has generated such large quantities of data though, nor has each benefited from data-driven insights. Infrastructure monitoring, the topic of this thesis, lags behind other fields considerably. For example, 28,000 riders use Pittsburgh's light rail everyday [4], yet the managers of the service rely on visual inspection, and the handwritten notes of the inspectors, to ensure the tracks are safe for the commuters.

There are a number of challenges which have stymied the growth of data-driven techniques to monitor more traditional industrial assets like railroad tracks, challenges which extend to much of the infrastructure domain. Rail assets are large, often spanning hundreds of kilometers, so

collecting the data can be expensive. Each railway network is unique, so developing an algorithm that can analyze many different networks is difficult. And few resources are available to monitor these assets; while as a whole, the networks serve as the backbone of commerce, individual assets are not highly valued.

One solution is to instrument a larger number of these traditional assets. And this is happening more and more. GE estimates that industrial data, which includes rail, is growing twice as fast as data in other categories [2]. The growing network of smart and connected traditional assets is known as the Industrial Internet of Things (IIoT). But to be useful, data from the IIoT needs to be analyzed. According to Cisco, only 3% of industrial data is labeled and used in a meaningful way [22].

Thus, in addition to collecting more data on these traditional assets, new techniques need to be developed to draw insight from this data. Data-driven approaches may offer a safer and more efficient way to manage traditional assets than the qualitative approaches used today. And while aging infrastructure is a daunting problem, data-driven management may be the most promising technical solution.

## 1.2 Motivation

In 2012, a freight train crossing over Mantua Creek in Paulsboro New Jersey derailed. Four tanker cars fell into a creek, one of which was punctured, releasing thousands of gallons of vinyl chloride into the sensitive wetland area. Although concerns had been raised about the bridge and the associated track, the lack of objective information made it difficult for the infrastructure managers to assess the risk.

This information gap is prevalent throughout the management of rail infrastructure, leading to potentially unsafe conditions and sub-optimal replacement of capital assets. According to the American Society of Civil Engineers, improving asset management by closing this information gap is one of the most promising techniques to improve national infrastructure. The National

Academy of Engineering (NAE) lists restoring and improving urban infrastructure as one of the 14 Grand Challenges facing humanity [50].

In this thesis, we focus on rail transit where more efficient allocation of resources could alleviate acute budgetary constraints. In rail transit, the national annual funding shortfall is $8 billion, and is expected to rise to $90 billion by 2020 [26]. Furthermore, the challenges of asset monitoring in rail transit, as one of the oldest industries, are emblematic of the broader challenges of bringing the benefits of modern data analysis to traditional domains.

## 1.3 Problem Statement

As discussed, there is relatively little data collected about the condition of railroad track. The rail industry uses a combination of visual inspection and track geometry vehicles to monitor track condition [5, 20]. For the rails themselves, rail flaw detectors are used to identify internal rail irregularities. Each of these methods has its drawbacks. Visual inspection is subjective and exposes the inspector to considerable danger. Track geometry vehicles and rail-flaw detection vehicles are are used infrequently due to their high operation and maintenance costs.

The shortcomings of these inspection techniques are highlighted by recent derailments. For example, in February 2015, 27 tanker cars filled with crude oil derailed in West Virginia. 1,100 residents were evacuated for 4 days while fire-fighters tried to extinguish the ensuing blaze. In their investigation, the National Transportation Safety Board said that the track problem which caused the derailment should have been identified in not one but two track inspections before the accident, in December 2014 and January 2015 [44].

In this accident, the inspections themselves were flawed, due to a combination of human error and inadequate data processing. However the larger problem was the limited number of inspections. Over the two month period when the track problem was evident, numerous trains passed over the tracks everyday. If data had been collected from sensors on the in-service trains, or even from sensors inside the smart phones of the crew, it is possible that the track flaw could

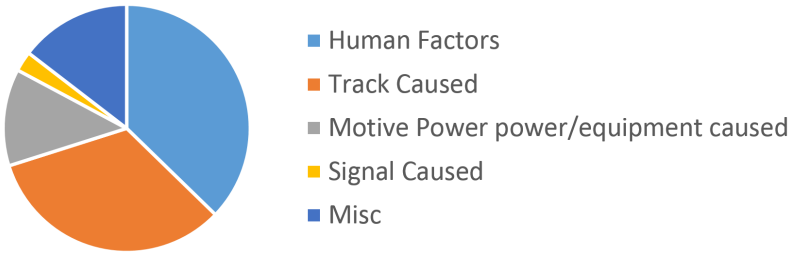



Figure 1.1: Train accidents in the US between January 2007 and February 2016. The data is from [51].

have been detected.

Of the 18,578 accidents that the US Federal Railway Administration (FRA) has documented over the last 10 years, accidents from track related problems were the second most prevalent as shown in Fig. 1.1. Further detail on the causes of these track accidents are shown in Table 1.1. While some track defects, like detail fractures, may only be apparent by using specialized rail flaw detectors, the majority of track failures, like wide gage track or broken switch points, are significant enough irregularities that they would affect the behavior of the trains passing over them.

One economical method to collect more data on the tracks, proposed by numerous researchers [7, 29, 42, 45, 52, 55, 69], would be to instrument in-service trains. In addition to collecting a larger volume of data, more frequent monitoring could help identify irregularities more rapidly after they occur. However, data collected from in-service trains would likely be of lower quality than the data collected from dedicated inspection vehicles; many operational variables, like the train's speed, could not be controlled.

As evident in the West Virginia accident, current analysis techniques miss some track irregularities already, even when using high-quality data. Using lower quality data from in-service trains further complicates the analysis. The central problem addressed in this thesis is how these large quantities of lower-quality data could be analyzed, and whether the information from these sources could improve overall track management.

| Track Failure Type | # of Incidents | Percentage |
|---|---|---|
| Wide gage | 1445 | 23.7% |
| Switch point worn/broken | 453 | 7.4% |
| Detail fracture, shelling/head crack | 435 | 7.1% |
| Transverse/compound fissue | 434 | 7.1% |
| Track alignment | 323 | 5.3% |
| Cross level track irregularity | 316 | 5.2% |
| Vertical split head | 257 | 4.2% |
| Head and web separation | 245 | 4.0% |
| Roadbed settled or soft | 202 | 3.3% |
| Catenary system defect | 180 | 2.9% |
| Broken rail base | 166 | 2.7% |
| Switch damaged or out of adjustment | 160 | 2.6% |
| Switch point gap | 146 | 2.4% |
| Other | 1346 | 22.0% |
| **Total** | **6108** | **100.0%** |

Table 1.1: Breakdown of track-caused train accidents in the US between January 2007 and February 2016. The data is from [51].

## 1.4 Ultimate Objective

The ultimate objective is to develop a system that would provide near continuous monitoring of entire rail networks by collecting large amounts of heterogeneous data from available sensors on in-service trains. One of the main functions would be to detect the location of track irregularities soon after they begin to occur. Although ideally, such a system would also provide a prediction of when the assets will fail, this would be nearly impossible with a purely data-driven approach. Instead, the system would provide the rate at which the irregularities were forming so that track managers could more effectively plan maintenance and take damage-prevention measures.

Of course, such a system could not be built overnight. Data on the tracks would have to be collected for some amount of time in order to build a baseline model of the tracks. At the same time, rail managers would have to label faults when they first occurred so that the system could learn the signatures of these irregularities. In addition, rail managers would have to start trusting the monitoring system, and this could happen perhaps most effectively during a transitional

period, where likely faults were identified by the system, then either confirmed or denied by rail managers.

Trust is a crucial element; too many early false positives might lead managers to believe the system is a waste of time, while too many false negatives might lead managers to believe the system is dangerous. To avoid these circumstances, black-box algorithms could be avoided in favor of analysis techniques with greater intuition, such as approaches which generate track parameters similar to the parameters which are used in traditional track inspections.

## 1.5 Literature Review

This thesis presents advances towards this ultimate objective, building on the previous work of numerous researchers. The idea of collecting information from in-service trains has been studied since 2006, when sensing, processing and storage hardware became cheap enough to make such monitoring economical.

Three main types of sensing technology have been proposed for in-service trains [46]: optical sensors (using lasers) [10, 66], magnetic flux sensors (also called Foucault currents or Eddy Currents) [31, 60] and inertial sensors (using accelerometers) [7, 9, 29, 34, 36, 37, 42, 43, 45, 49, 52, 55, 69]. Optical sensors are used widely on track geometry cars, however, they stop functioning if the lens becomes dirty, and thus require constant maintenance. As such, they are not appropriate for long term-monitoring from in-service vehicles [68]. Foucault current monitoring requires a magnetic coil placed close to the rail, making the coil vulnerable to objects along the track. Inertial sensors that monitor the vertical accelerations of the train have become the most popular approach. The sensors are often placed on the axle box [7, 42, 69], but can be placed anywhere on the train, even inside the cabin [49]. The challenge with using accelerometers lies in analyzing the collected data.

There have traditionally been two approaches to analyzing train-based accelerometer data: using an explicit model or an implicit model. Explicit models attempt to determine the precise

track profile, typically using *a priori* knowledge of the train and its suspension system to solve an inverse problem [49, 55]. For each pass over the tracks, the track profile is estimated; deterioration in the tracks can be detected as changes in the profile. The challenge with this approach is that solving this inverse problem can be unstable as the problem is ill-posed. Real et al. [55] filter the collected data to estimate the track profile more reliably, although their approach does not provide consistent estimates for different track sections. O'Brien et al. [49] propose a cross-entropy method to determine the track profile, yet their approach is computationally inefficient and has yet to be tested on operational data. Further details on both of these methods are provided in section 4.2. Neither method has been used for long-term track monitoring, likely because the solutions are not stable.

Implicit models derive a feature from the accelerometer data which serves as a proxy for track geometry or roughness. These models do not suffer from the instability of explicit models, but are typically less sensitive to changes in the tracks. Notable implicit models have been based on wavelets [7, 8, 9, 13], the Short Time Fourier Transform (STFT) [42], and signal standard deviation [69]. Many of these features are designed around specific track irregularities of interest; for example, Bocciolone et al. [7] use wavelets to detect track corrugations, and Molodova et al. [43] use STFT features to detect track squats. Ideally, a single feature should be capable of detecting numerous track irregularities; we explore which features can serve as a general purpose indicator of track health in Chapter 4.

Monitoring the tracks requires not only modeling the state of the tracks, but also identifying when irregularities occur. Approaches for this detection, typically known as anomaly detection or change detection algorithms [12, 47], have been studied in related fields, but have not been studied within the train-based track monitoring space. One reason is that data from many passes of the train over a particular section of track are required before anomaly detection approaches can be used, and not many studies have gathered sufficient data. Perhaps the study with the closest similarity to anomaly detection is a study by Molodova et al. [42]. In that study, rail squats are detected when the features extracted from the signal (STFT) exceed a pre-defined

threshold. This is not change detection as we will use the term throughout this thesis, because the irregularity is not detected in comparison to historical data. However, this study is mentioned as it is the closest to presenting a complete track monitoring system to the author's knowledge.

Finally it should be noted that studies thus far have concentrated on analyzing data from individual sensors, and individual train passes. To achieve the ultimate objective of network-level monitoring, data from different sensors, passes and even trains need to be combined. More detail on previous work on data fusion in related fields is provided in Chapter 5.

## 1.6 Hypothesis

Building on this past work, we begin our investigation of train-based track monitoring with a hypothesis: that information on the state of rail-infrastructure can be gleaned from vibrations felt on an in-service train; that using a data-driven framework to analyze these vibrations, entire rail-networks can be monitored without needing to manually specify information about the tracks or about the trains themselves; that the sensors can be placed anywhere on a train; and that by placing sensors on multiple trains, more continuous monitoring and more rapid detection of rail-infrastructure irregularities can be achieved.

Relative to prior studies, our hypothesis offers several contributions. We propose to build a baseline model of the tracks using a data-driven approach by looking at data from the tracks over time. While previous studies have detected particular types of track damage defined *a priori*, we propose to detect any point in the tracks which differs from the baseline track model. The proposed method can detect a variety of track defects, while doing so in an unsupervised fashion. Finally, while previous studies have analyzed data from single sensors on single trains, we present a formal technique to fuse data from multiple sources. This data fusion could facilitate more reliable network-level track monitoring.

## 1.7 Research Questions

In order to test this hypothesis we have three specific research questions (RQs).

**RQ1: How can the track state be modeled from the collected data, and in particular, how can operational variability be mitigated when building track models?**

We use the term "track state" as a proxy for track condition. Because we use a data-driven approach, we do not know the precise condition of the tracks from the data, but, by detecting a change in the data, we can say that the track has transitioned from one state to another. Thus the "track state" is a representation of the track condition derived from the collected vibration data. As mentioned in the literature review, typically, the track state is modeled either implicitly, in the form of extracted features, or explicitly, in the form of the track profile. Both models attempt to describe the track state independently of the operational conditions of the train. The train's variable speed and the uncertainty about its position are the greatest challenges in analyzing the accelerometer data; in the thesis, we propose new modeling techniques which address these challenges.

**RQ2: How rapidly and how reliably can relevant changes in rail-infrastructure be detected from an in-service train, and how can detection occur in an unsupervised fashion?**

Track network operators want inspection techniques that are low-cost and reliable, and which can detect faults soon after they occur. While our proposed method of monitoring the tracks from operational trains is low-cost by nature, this research question addresses the other two objectives, detection reliability and detection speed. In part, this question is closely related to the first research question; if the track models are perfectly accurate, detecting track changes from these models is a trivial task. However, modeling the track state is not perfect, and identifying only the most statistically significant changes is an important task for change detection. In general, there is a trade-off between either detecting changes in the track quickly, or gathering more data so that the found changes are more statistically significant. However, as will be discussed, by using certain change detection approaches, and, by combining data from multiple sources, both

the detection reliability and the detection speed can be increased.

It should be noted that for the majority of this thesis, we focus on detecting sudden changes in the tracks, i.e. changes that occur over a short period of time. These changes are the most readily detected in an unsupervised setting; however, many aspects of the framework also apply to detecting gradual changes, as is discussed in section 5.5.

**RQ3: How can a data-driven monitoring framework be scaled-up to include data from multiple trains across an entire rail-network?**

The likelihood of a single faulty sensor may be low, but this likelihood increases as more sensors are included in building the track model. Accordingly, we investigate data fusion methods to combine data from multiple sources, while weighting each source according to how accurate it has been historically. One benefit is that this data fusion helps to build more reliable track models. A second benefit is more rapid detection of track irregularities; as more trains are instrumented, the frequency with which the tracks are interrogated increases.

# Chapter 2

# Research Method

In order to test our proposed methods, we instrumented two trains operating within Pittsburgh's 40km light-rail network. This network has been pieced together over more than a century and the variety of assets in the system makes it a good test-bed. The network includes bridges, viaducts and tunnels, as well as both street running track and ballasted track. In addition, Pittsburgh's temperate climate allows for the observation of large environmental variability; we have observed temperatures lower than $-20^o$C and higher than $35^o$C. Because our test-bed was an operational system, we could not conduct experiments by damaging the tracks; instead we had to wait for changes to occur naturally in the network, then test to see if we could detect these changes.

## 2.1 Instrumentation

The trains are operated by the Port Authority of Allegheny County; the Port Authority generously helped to install and maintain the sensors for this research project. Our goal was to build a system that could be widely deployed; to that end, we used low-cost off-the-shelf components and placed most of the sensors inside the cabin where they were easy to install and were protected from the elements.

We placed sensors on two trains, each 27m long light-rail vehicles weighing 40 metric tons

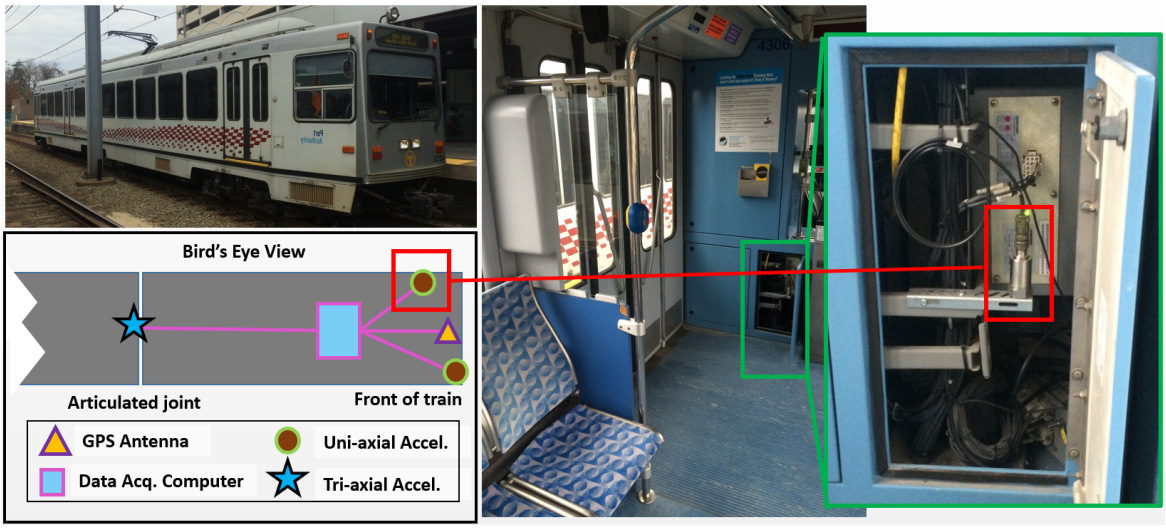



Figure 2.1: Instrumentation of Train 1. Counter clockwise from top left: a view of the exterior of the train, a schematic of the sensor locations from above, and a picture from inside the train with an inset showing details of the electrical cabinet where one of the uni-axial accelerometers is installed. The red box identifies the accelerometer and highlights which sensor from the schematic is pictured.

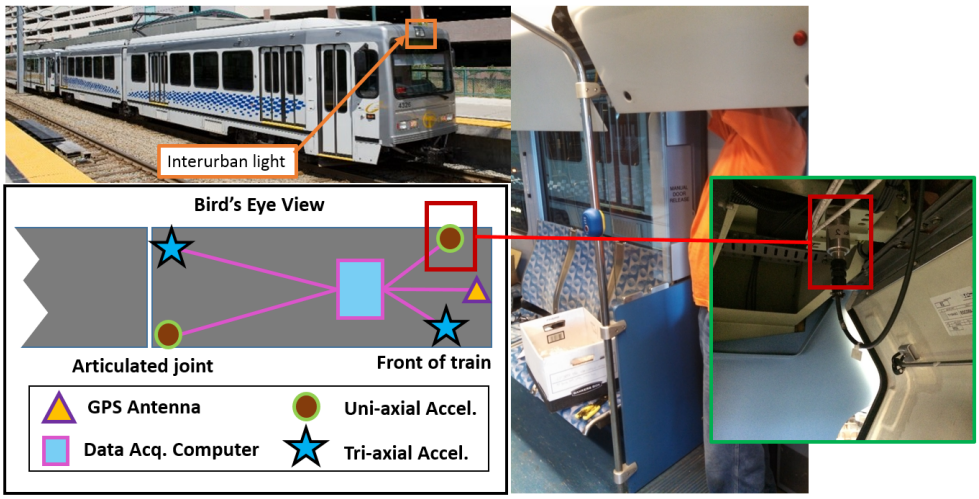



Figure 2.2: Instrumentation of Train 2. Counter clockwise from top left: a view of the exterior of the train, a schematic of the sensor locations from above, and a picture showing the installation of the sensor in one of the electrical cabinets in the ceiling of the train. An inset shows detail the accelerometer installed inside this ceiling compartment. The red box identifies the accelerometer and highlights which sensor from the schematic is pictured. In this instrumentation, the GPS antenna was installed within the interurban light enclosure, which means it had a partial view of the sky.

made by the Spanish firm Construcciones y Auxiliar de Ferrocarriles S. A. (CAF). In our first instrumentation, shown in Fig. 2.1, we placed two uni-axial accelerometers inside the cabin of the train (Vibrametrics 5102) and a tri-axial accelerometer (PCB 354C03) on the central wheel truck. This wheel truck is not powered and was selected to minimize electrical noise. We used National Instruments data acquisition hardware connected to a computer, which samples at 1.6kHz then logs the data to an external hard drive. For position, we used a BU-353 GPS antenna, logging position at 1Hz through the same data acquisition computer.

For our second instrumentation, we were able to improve upon the system from our first instrumentation. As seen in Fig. 2.2, we used more sensors than the first instrumentation with a particular focus on sensors inside the train: 2 uni-axial accelerometers (Vibrametrics 5102) and 2 tri-axial accelerometers were installed inside the train (PCB 354C03), and 2 industrial grade accelerometers (IMI 623C00) were installed on the central wheel truck. While the first system relied directly on power from the train which can occasionally go out, our second system had a built in back-up battery for uninterrupted operation. Where the first system had the GPS antenna mounted just under the roof, in the second instrumentation, the GPS was placed in the inter-urban light casing, allowing it a partial view of the sky.

Besides accelerometer data and position data, we were able to gather environmental information indirectly. To determine the environmental conditions, we used the train's GPS position to query environmental conditions such as temperature, wind and precipitation from a weather database called Forecast.io. We queried conditions when post-processing the data from historical weather models using the time stamp when the data was collected.

## 2.2 The Dataset

We collected data from the trains over a number of years; we gathered 31 months of data from Train 1 and 11 months of data from Train 2. Raw data was collected continuously from both trains, whether they were moving or in the yard. Train 1 generated roughly 10 GB of data per

| | Train 1 | | Train 2 | | |
|---|---|---|---|---|---|
| Region | Inbound | Outbound | Inbound | Outbound | **Total** |
| 1 | 226 | 363 | 51 | 82 | **722** |
| 2 | 569 | 577 | 138 | 136 | **1420** |
| 3 | 579 | 567 | 135 | 131 | **1412** |
| 4 | 288 | 292 | 33 | 34 | **647** |
| 5 | 317 | 317 | 102 | 96 | **832** |
| 6 | 440 | 425 | 116 | 110 | **1091** |
| 7 | 356 | 342 | 85 | 80 | **863** |
| 8 | 180 | 182 | 65 | 63 | **490** |
| **Total** | **2955** | **3065** | **725** | **732** | **7477** |

Table 2.1: Number of passes collected from each instrumented train through the respective regions.

day, while Train 2 generated twice that amount because it had more sensor channels.

The first step in processing the data was to separate it by geographical region. This was necessary for two reasons. First, not every train traveled the same route, and we wanted to extract data over particular regions of track where the train's path was always the same. This allowed passes from that region to be compared to one another. Second, there are a number of tunnels in the system where the GPS signal is lost; we divided some areas to ensure that in each region, the GPS signal was continuous. The geographical regions, overlaid on typical GPS traces, are shown in Fig. 2.3.

The number of passes collected from each region are shown in Table 2.1. Given that most of the network is south of downtown Pittsburgh, "Inbound" is typically northbound tracks. One oddity is that "Inbound" track north of the city actually goes away from downtown, but this is the naming convention used by the Port Authority and we follow it in this thesis. Region 1 is known as the "North Shore Connector;" it is a short section of above-ground track just north of the Allegheny River. As can been seen in the table, there appear to be more outbound passes over this region than inbound passes. This imbalance occurs because as the train emerges from the tunnel, there is a delay before the GPS can get a position lock. Occasionally, the train will reach the station (where it turns around) before the position lock occurs, in which case the inbound pass over that section of track is never registered. Another point of interest is the low number of

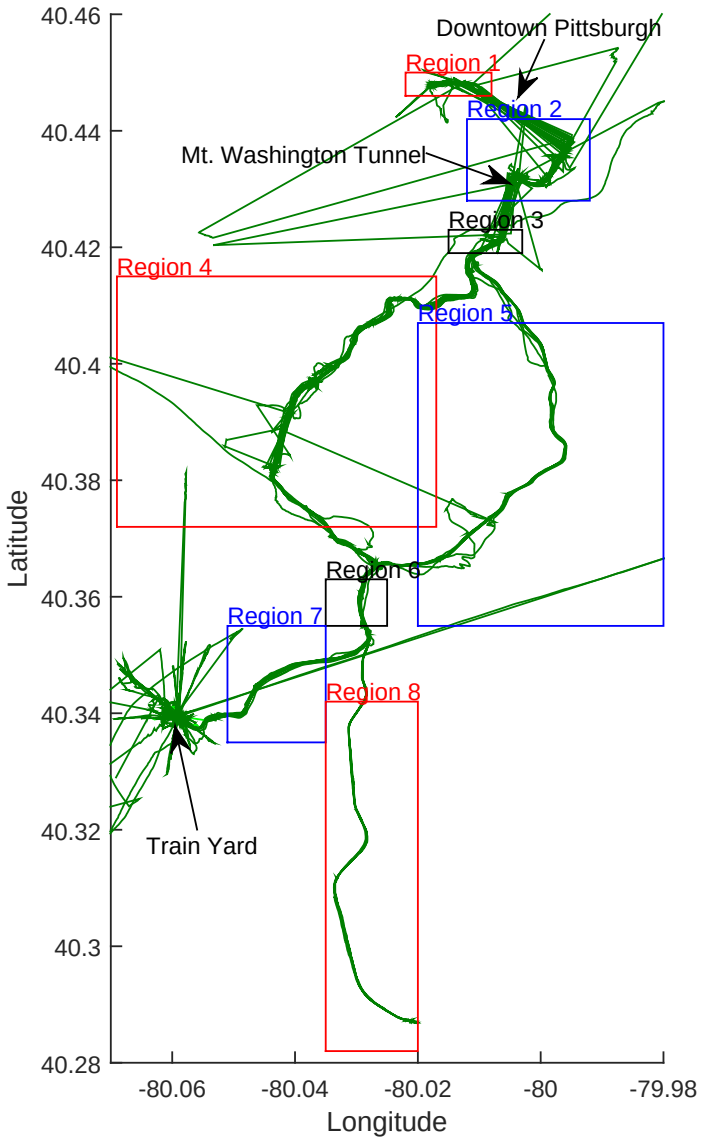



Figure 2.3: An example of the GPS trace of several passes through the network and the associated track regions used for analysis.

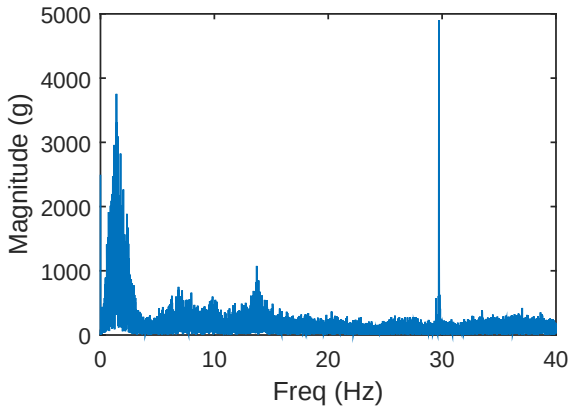



Figure 2.4: Frequency spectrum from one pass over a 4km track. Notice the sharp peak at 30Hz; this is from the train's ventilation system.

passes over Region 4, particularly from Train 2. This is because track in Region 4 was closed for a nine month rehabilitation project for much of the time that Train 2 was on-line.

## 2.3 Sources of Noise

One of the challenges in analyzing data collected from within the train is the high levels of noise. In this section we explore one of these sources of noise in detail: the train's ventilation system. We define noise here as any components of the recorded accelerometer data that do not describe the state of the tracks.

Fig. 2.4 shows a portion of the frequency spectrum of one pass of vibration data collected from Train 1. The majority of the energy is around 1.2 Hz which corresponds to the natural frequency of the train's main suspension. Because the train's main suspension is excited by rough sections of track, this part of the signal is useful for determining the track's condition. The sharp peak at 30Hz corresponds to vibrations from the train's ventilation system. This component of the signal is independent of the state of the tracks, so for our purpose, we can label it as noise. Further evidence that this 30 Hz component is from the ventilation system is shown in Fig. 2.5. Here a spectrogram of the data is shown. The train's speed varies across the pass (which spans 4km); the vibrations are low at some time points across most frequencies when the train comes

to stop. However the signal energy at 30 Hz does not depend on the train speed or interaction with the track; instead it depends on the external temperature. On a cold day, Fig 2.5a, the air conditioning is off so there is less energy at 30 Hz. On a warm day, Fig 2.5b, the air conditioner operates intermittently, so high energy is seen only occasionally at 30 Hz. Whereas on a hot day, Fig 2.5c, the air conditioner operates continuously.

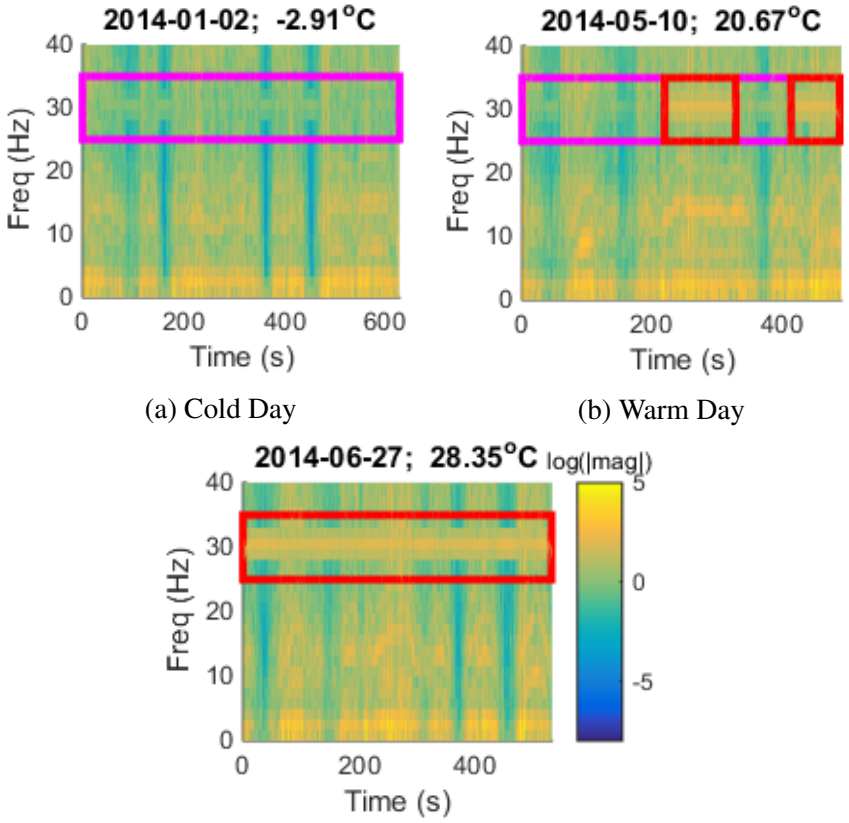
In general, we found temperature did not have a significant effect on the fundamental natural frequency of the train, but we did find it could affect sources of noise like ventilation. Ventilation turns out to be relatively insignificant compared with operational conditions such as varying train speed, however it is shown here just to illustrate some of the challenges in analyzing the collected data.

## 2.4 Validation Methodology

Ultimately, the propose of this instrumentation is to see how well we can monitor the condition of the tracks from an in-service train. However, knowing the true state of the tracks at any one time is nearly impossible, which makes evaluating different track monitoring approaches a challenge.

To circumvent this, we evaluated the efficacy of each proposed approach at identifying known track changes. In addition to the data we collected from the trains and indirectly about environmental condition, we had information about maintenance activity performed along the rail-network. This information was in the form of a weekly "track allocation report" that detailed which contractors and repair crews were allowed to be on the track and in what track sections. Using the descriptions of the planned work activities, we could infer what track changes had occurred.

While we could not perform controlled experiments on the operational rail network, we used these maintenance activities as a sort of natural experimentation. We were able to see which analysis techniques were best able to differentiate between the state of the track prior to the maintenance activity with the state of the track after the maintenance activity.




(a) Cold Day

(b) Warm Day

(c) Hot Day

Figure 2.5: Spectrograms of signals from three passes over a 4km section of track. (a) A spectrogram of a signal from a cold day ($-3^o$C); the noise level at 30Hz is low because the air conditioner compressor is off. (b) A spectrogram of a signal from a warm day ($21^o$C) where the air conditioner turns on intermittently. (c) A spectrogram of a signal from a hot day ($28^o$C) where the air conditioner compressor runs continuously. In all three figures, the time periods (vertical lines) with low signal energy are periods where the train comes to a stop. Notice that the air conditioner noise is independent of the train speed. The purple boxes indicate time periods where the air conditioner is off; the red boxes indicate times when the air conditioner is on.

Detecting rail maintenance itself could have applications, such as ensuring that contractors perform their work properly. However a more valuable application may be to detect deterioration in the track network. Just as maintenance transforms the track from a state of poor repair to a state of good repair, deterioration would be the opposite, and could be detected using a similar data-processing pipeline. In both cases, approaches which are sensitive to the track condition, but robust to the operational conditions of the train, are desired. How such approaches are developed is the subject of the next chapter.

# Chapter 3

# Implicit Models

In this chapter, we explore implicit models of the tracks. These models are based on features extracted from the raw signal, the best of which are robust to operational uncertainty. Ideally these track models are consistent so long as the track condition remains constant, and sensitive to track irregularities when they occur. We study the ability of different features and different change detection approaches to detect track irregularities from changes in these implicit models. We use both simulated data and data from our operational system to study two types of track irregularities, changes in track geometry due to tamping, and changes in the tracks themselves due a repair of a broken track.

## 3.1 Introduction

In building a data-driven monitoring system, consistent models of the track must be built from the collected data, so that when deterioration or other irregularities occur, they can be readily detected. However, the collected data varies considerably with each pass of a train over the infrastructure of interest, making consistent modeling a challenge. This occurs for two main reasons. The first is that a train's speed over a section of track differs with each pass, so methods robust to train speed must be found for comparing data between passes. This is particularly

challenging when the sensors are in the train's cabin, because the train's suspension filters the roughness from the track differently depending on speed. The second challenge is that we do not know the precise location of the train, due to GPS noise, so position uncertainty must be considered in attempting to detect track changes from the vibration signal.

We learned about these challenges from analyzing operational data, but to gain further insight into the vehicle-based monitoring problem, we began with a simulation. We modeled a simplified version of the train-track interaction using a single degree-of-freedom oscillator travelling over a rough track. This parametric simulation allowed us to explore the effects of variable speed and position uncertainty in a controlled setting before validating on the light-rail data.

Simulation has three main benefits. First, we can generate much more data than we could collect from instrumented vehicles. Although we have been collecting data from the light-rail system for several years, there are relatively few recorded maintenance events in the rail-network each year that we can use to test our approach. With simulation, we can rapidly generate hundreds of track changes. Second, we can simulate a wider range of parameters than the narrow band we have observed in our operational system. For example, while the instrumented light-rail vehicle has one fundamental natural frequency, we can simulate numerous natural frequencies to ensure that the analysis techniques we find are general. Third, in the data we have collected, we do not have the ground-truth of the train's position. Through simulation, we can study the effect of position error, and propose techniques that work well for the level of error we expect in operational systems.

While our ultimate objective is to build a complete track-monitoring system, we focus in this chapter on two main components required for automated track monitoring: meaningful feature extraction from the raw vibration signals and detection of track-changes from these features. For the first component, we examined four different features, and then used supervised classification to determine which one provides the most reliable information about the state of the track. The objective is to determine which features are most robust to the sources of uncertainty inherent in train-based monitoring. The fact that the train's speed varies between each pass (or the speed

of the travelling damped oscillator in simulation) made comparing the data from multiple passes challenging. The signals are recorded in the time domain, but given that track changes are localized in space, the spatial domain is often more useful for detecting track changes. We examined features based on the spatial-domain representation of the signal, spatial frequency domain and temporal frequency domain as well as features based on the energy in the signal. We show the discriminant power of these features on data from our parametric simulation as well as on the light-rail dataset.

The second component in our automated detection system, change detection, is a challenging task because we do not know *a priori* the type of change we are trying to detect [47]. The most closely related study of automated track anomaly detection from the vibration signal of a train is that done by Molodova et al. [42], where a system for detecting track squats (a type of rail-surface indentation) was proposed. In the study, a detection event was triggered by a vibration signal above a pre-defined threshold. We aim to build a broader detection system where anomalies are defined as changes relative to historical behavior. This ensures that areas with consistently high vibrations (like track switch gear or joints) are not labeled as anomalies, while changes, even in areas with low vibration-amplitude, are detected. For example, we will show that our method detects changes in track geometry due to tamping that can have safety implications despite their small amplitude. This historical detection technique allows for the monitoring of an entire network without manually tagging problematic areas as the method in Molodova et al. requires.

To perform change detection, we experimented with common methods like cumulative sum chart control (CUSUM) [54] and generalized likelihood ratio (GLR) [27], as well as a simplistic Haar filter [62, 64] borrowed from the field of signal processing and computer vision. We report the performance of these approaches both on our simulated data as well as on the light-rail dataset.

As a validation, we apply these feature extraction and change detection approaches to data we collected from the instrumented light-rail vehicle. We examine our ability to detect a broken track

(a localized spike in the track profile) which is then replaced, as well as tamping, a maintenance procedure which subtly corrects track geometry. In both cases, information about when and where the maintenance took place can be found in the track maintenance logs, thus detection accuracy for our system can be evaluated.

## 3.2 Track-Monitoring Method

Our proposed rail-monitoring method uses data collected from accelerometers inside a train to identify changes over time in the rail infrastructure. In thinking about how to analyze the data, our first goal was to understand how track roughness, filtered by the train's suspension, produces vibrations in the train's cabin. To do this we used the simplest possible model, a travelling damped oscillator as shown in Fig. 3.1, and conducted a simulation study as will be described in the following section. We sought to determine how best to detect changes in a section of track as the oscillator traveled repeatedly over it. This same type of model has been used previously in the track-monitoring literature [6, 55], but we extended that work to include variable train speed and differing levels of position uncertainty [40].

### 3.2.1 Feature Selection Simulation

We conducted a simulation of an oscillator travelling over a rough track to understand which features, when extracted from its dynamic response, are sensitive to track changes, but robust to speed and position uncertainty. The framework of the simulation is shown in Fig. 3.2. First we simulated a track profile and a change in that profile, for various types of track repairs. We then simulated the oscillator passing over the roughness 100 times before and after the track-repair, where each pass over the track had a unique speed profile, and extracted relevant features from the dynamic response of the oscillator. Finally, we quantified how well we could differentiate between the data before and after the repair. We repeated the steps of the flow chart in Fig.

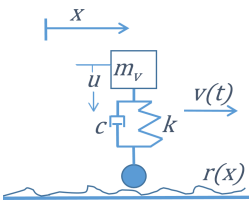



Fig. 3.1: Traveling oscillator moving over a rough track

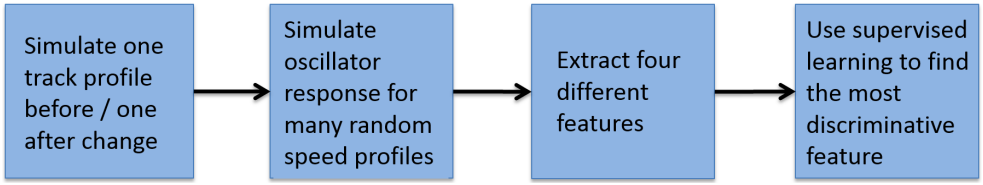



Fig. 3.2: Flow chart of the simulation. Note that this process was repeated for the three types of track changes, for a variety of damping ratios, natural frequencies, and position uncertainty levels.

3.2 for oscillators of different natural frequencies and damping ratios. The goal was not only to understand the behavior of the oscillator for parameters most similar to those of the light-rail system, but also to gain insight into how the results would vary for different rail systems so as to make the results more general.

When simulating the roughness, we generated a 1km section of track for three change types, each of which is shown in Fig. 3.3. The first track change, in Fig. 3.3a, is a toy-model of roughness used to visualize the simulation later in this chapter (greater detail is shown in Fig. 3.3b). For the second and third track changes (Fig. 3.3c and Fig. 3.3e), we simulated two types of track changes we had observed in practice, a large localized spike which is removed, simulating the replacement of a broken track (detail in Fig. 3.3d), and a smoothing of a track profile, simulating tamping (detail in Fig. 3.3f). For each of these realistic changes, we also simulated a realistic track roughness using the parameters found in the literature [14]. For the tamping change, we filtered this track roughness as per the smoothing effect of the tamping machine documented in [21]. The smoothing effect of the tamping machine is shown in Fig. 3.4; it reduces the standard deviation of the track profile from 2mm to 1.5mm over a 200m section of track. For each simulated profile, we generated 25 samples per meter for 25,000 samples per

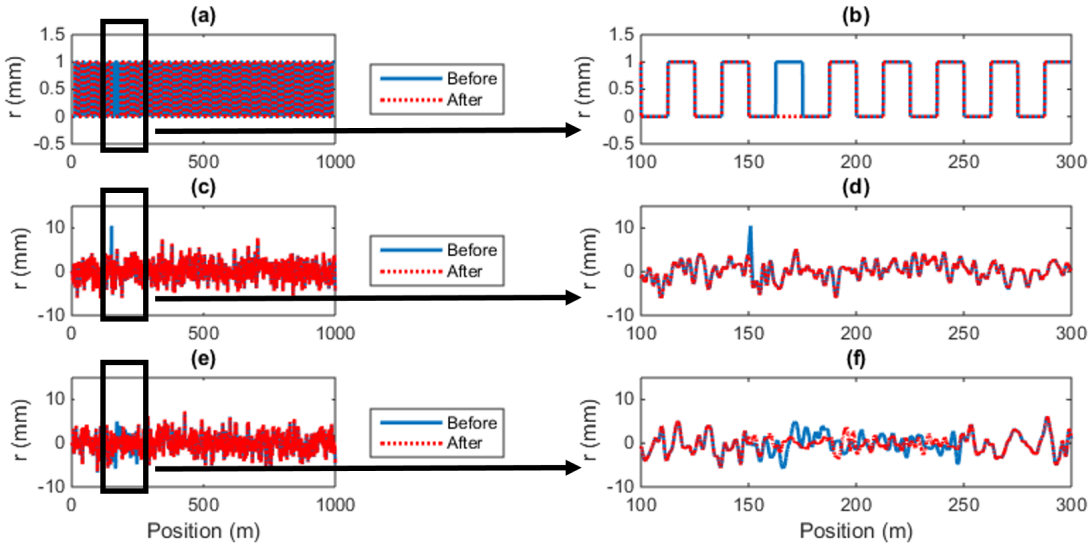



Fig. 3.3: Three types of roughness changes (left plots), with detail of each (right plots). (a) A toy-model change (used later to provide visual intuition about the simulation) and (b) the detail of the toy-model change. (c) A spike change, characteristic of a broken track before and after replacement, with a realistic track roughness and (d) the detail of the spike change at 150m. (e) The tamping change, simulated using a filter with the same smoothing effect of the tamping machine and (f) the detail of tamping which occurs between 150m and 250m.

profile; the wavelengths which most influence the fundamental frequency of train cabin (and thus our oscillator) are between 2m and 50m, so this sampling rate provided adequate resolution.

Once we generated the track roughness, the next step was to generate the response of the oscillator; two realizations of this process over the same roughness with different speed profiles are shown in Fig. 3.5. Fig. 3.5a shows the speed profiles in the time domain, while Fig. 3.5b shows them in the spatial domain (i.e. plotted against position). Note when the speed approached zero, no distance is covered, producing the scalloping effect in Fig. 3.5b, a phenomenon which is common when the train stops at a station. When generating the speed profile, we limited it to be between 0 and 15m/s (35 MPH / 55 KPH) which is the same as that of the light-rail vehicle in our deployment. The toy-model roughness profile is shown in Fig. 3.5d as a function of position, but the train experiences this profile in the time domain Fig. 3.5c. We generated the response of the oscillator in Fig. 3.5e by solving a differential equation [15],

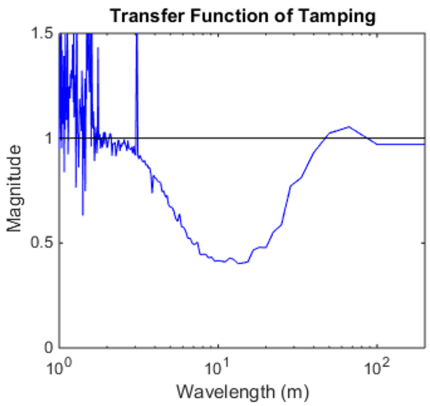



Fig. 3.4: Simulated change due to tamping. The wavelengths which are most effected by tamping are between 2m and 50m which are the wavelengths which most affect the dynamic response of the light-rail. The tamping change is based on data from Figure 3 of Esveld et. al. 1988 [21].

$$\ddot{u}(t) + 2\zeta\omega_n(\dot{u}(t) - \dot{r}(t)) + \omega_n^2(u(t) - r(t)) = 0, \tag{3.1}$$

where $\zeta$ is the damping ratio of the oscillator, $\omega_n$ is the natural frequency of the oscillator, $u$ is the displacement of the oscillator, and $r$ the track roughness as shown in Fig. 3.1. For each pass, we simulated 200 seconds of the dynamic response of the train, with a time-step of 0.01s; thus each pass had a length of 20,000 samples.

Although the bumps in the track occur at the same location in space, due to the difference in the speed profiles, the oscillator is excited in the two passes at different points in time. This posed a challenge when comparing multiple passes in the time domain, so we interpolated the signal into the spatial domain as seen in Fig. 3.5f. Given the response of the oscillator is smoother than the track roughness, when performing linear interpolation, we sample from the dynamic response at a rate of 10 samples/m. The bumps experienced by the oscillator line up more closely in the spatial domain, but the effects of varying speed are still visible. The wavelengths of the oscillators (the distance over which they occur) varies considerably in Fig. 3.5f due to the speed, even though the periods of the oscillations (their duration in time) are invariant, as can be seen in Fig. 3.5e.

The variation in Fig. 3.5 highlights one of the challenges of dealing with variable speed;

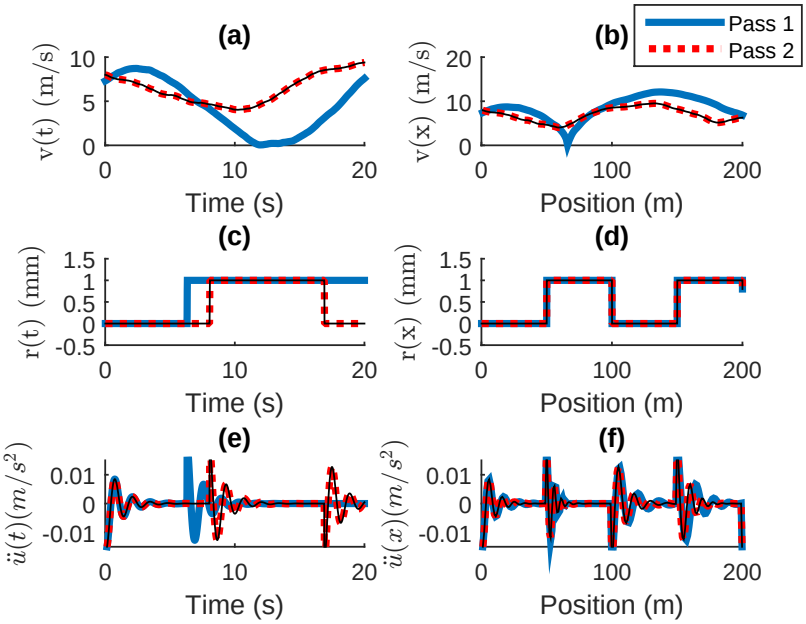



Fig. 3.5: Two passes over the toy-roughness shown for illustration: (a) the speed profiles in the time domain; (b) speed profiles in the spatial domain; (c) the roughness interpolated in time; (d) the roughness in space; (e) the acceleration of the oscillator in time; and (f) the acceleration of the oscillator in space.

a second challenge lies in not knowing the exact position of the oscillator, a phenomenon that occurs in practice due to GPS error. In order to plot the signal in the spatial domain (Fig. 3.5f), we used the position of the oscillator, $x(t)$. In practice, we only know some approximation of the position, $x_\epsilon(t) = x(t) + \epsilon(t)$, where $\epsilon$ is the error. Let us assume that this error is normally distributed with zero-mean and standard deviation, $\sigma$. We show the effect of this error with different standard deviations in Fig. 3.6.

As the train's position and the associated uncertainty are a central part of this study, the method in which the uncertainty is applied requires further comment. The position vector, $x$, is generated based on the speed profiles shown in Fig. 3.5. Because the speeds are positive, the position either increases or remains the same with each step. In order to simulate position uncertainty, we add Gaussian noise, $\epsilon_0$. From a practical standpoint, in order to interpolate the data spatially, the position vector must be monotonically increasing, so this noise poses a challenge. Physically, if the position vector does not increase monotonically, it means the train is moving backwards at some time-steps, something we know does not happen. To avoid this problem, we sort the values as follows:

$$x_\epsilon = \text{sort}(x + \epsilon_0). \tag{3.2}$$

This new position vector differs from the original position vector according to $\epsilon$, where

$$\epsilon = x - x_\epsilon. \tag{3.3}$$

This resulting error follows the normal distribution according to $\epsilon \sim \mathcal{N}(0, \sigma)$. The standard deviation of the resulting error, $\sigma$, is used to characterize the extent of the position uncertainty throughout this chapter.

While in Fig. 3.5f we showed the vibration signal for two passes in the spatial domain, in Fig. 3.6a we show 200 passes, where each horizontal line is a vibration signal from a single pass, and

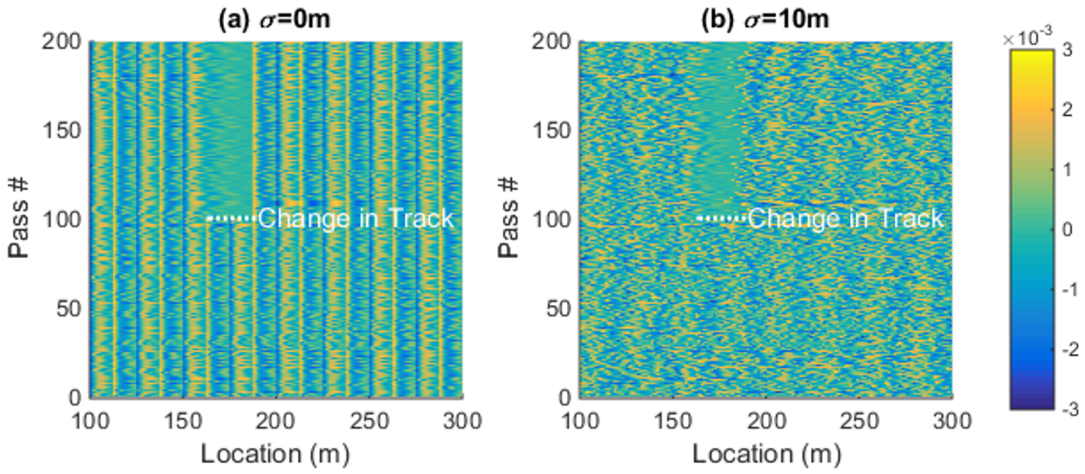



Fig. 3.6: Effect of position uncertainty. Each row of the above plots shows one pass of the oscillator over the toy-model roughness shown in Fig. 3.3. Passes 1-100 correspond to the "before" roughness, while passes 101-200 correspond to the "after" roughness. The color along each line represents the acceleration of the oscillator in $m/s^2$ but has been truncated at current bounds to show greater clarity. Note that each pass has a unique speed profile. (a) Shows the response of the oscillator in space with no position uncertainty, (b) shows the response with added uncertainty (zero mean and standard deviation of $\sigma = 10$m).

the color is indicative of the instantaneous acceleration. Note that we show 100 passes before the track change, and 100 passes after the track change from Fig. 3.3a. In Fig. 3.6b and Fig. 3.6c we show higher levels of position uncertainty by varying the standard deviation of the error, $\sigma$. As the position uncertainty grows, detecting changes in the tracks becomes harder, although detection is still possible (even visually) for this type of a trivial track change.

In Fig. 3.6 we show the vibration signal plotted spatially as a more intuitive signal representation for understanding the effect of the track on the oscillator. To quantify how well this representation portrays track condition, we consider it as a feature, and compare it with three other features, temporal-frequency, spatial-frequency and signal-energy, to see which provides the greatest discrimination of track condition. Each is described in the equations below where $F[\cdot]$ denotes the Fourier Transform, $\ddot{u}_n$ is the vector of length $p$ of collected data from the nth pass, $f_n$ is the feature vector (length $p$), and $x_\epsilon$ is the position vector with added noise $\epsilon$ (also length $p$).

$$\begin{aligned}
&\text{Temporal-frequency} && f_n = F[\ddot{u}_n] \\
&\text{Spatial-frequency} && f_n = F[\ddot{u}_n|_{x_\epsilon}] \\
&\text{Spatial-amplitude} && f_n = \ddot{u}_n|_{x_\epsilon} \\
&\text{Signal-energy} && f_n = \ddot{u}_n^2|_{x_\epsilon}
\end{aligned} \tag{3.4}$$

The motivation for having two types of frequency-based features is the prevalence of frequency-based features in the literature [11, 13, 35, 53] or features related to frequency, such as wavelets [9]. "Temporal" here means the signal in the time domain as it was originally acquired, whereas "spatial" denotes the signal has been linearly interpolated spatially given its speed profile. The signal-energy feature is similar to the spatial-amplitude, but it is squared prior to interpolation. This squaring makes a difference because it makes the feature have a non-zero mean. Prior to classification, the spatial-amplitude and signal-energy features were averaged over a moving 25m window of track. This step increased robustness to position uncertainty for both features, but can filter out some zero-mean oscillations.

To evaluate the features, we used supervised classification; the goal was to see which features allow for the clearest indication of track change. Supervised classification means that the "label" of the pass is known during a "supervised" training phase. In this case we use a binary label as to whether the change in the track has occurred. (By contrast, "unsupervised" approaches are a broad family of algorithms which do not require labels, like the change detection approaches explained later in the text.) Here we used a support vector machine classifier with a linear kernel (we choose a simple model to avoid overfitting given the relatively few passes we use in our experiments with operational data). We selected 150 passes out of the original 200 for training (the remaining 50 are for testing), and we repeated for 50 fold-cross validation. In each case, we focus on a 500m section of track around the change of interest, a procedure we also use later in this chapter for examining changes in the operational data. (This means the feature vector has a

length of 5000 samples for the spatial data, and between 2000 and 10000 samples for the time data, depending on the train's speed).

High classification accuracy means that the data is more easily separable, i.e. the feature is useful. The features were long, so we select only the 50 most discriminative indices of the feature. To accomplish this, we use the technique described in [11], where we find the mean signal for both classes from the training data, then define the most discriminative indices as those with the greatest difference between the two mean signals. In total, to explore feature selection, we simulate 180,000 passes of the oscillator over 1km of simulated track (3 types of track changes $\times$ 6 damping ratios $\times$ 5 natural frequencies $\times$ 10 levels of position uncertainty $\times$ 200 passes each). The 6 damping ratios are 0, 0.1, 0.2, 0.3, 0.4 and 0.5. The five natural frequencies are $3\pi/2, 2\pi, 5\pi/2, 3\pi$, and $7\pi/2$rad/s.

### 3.2.2 Feature Selection Simulation Results

The classification results from the simulation are shown in Fig. 3.7 and Fig. 3.8. This was a binary classification, so 50% accuracy would have meant the labels were random, while higher accuracy meant more consistency in determining the state of the track given the selected features. We report accuracy while varying the level of position uncertainty (Fig. 3.7a), the oscillator's damping ratio (Fig. 3.7b) and the oscillator's natural frequency (Fig. 3.7c). In Fig. 3.7a, we kept the damping and frequency constant at values we observe in the light-rail system ($\zeta = 0.2$; $\omega_n = 5\pi/2$rad/s) [17]. We found that temporal-frequency features provide low accuracy at all levels of position uncertainty, spatial-amplitude and spatial-frequency fall in accuracy as the position uncertainty increases, and signal-energy offer relatively high accuracy at all levels of position uncertainty. It is not surprising that temporal-frequency performed badly because it is sensitive to speed changes, which we varied between each run. Spatial-amplitude and spatial-frequency require interpolation by $x_\epsilon$, so they are sensitive to increases in the uncertainty $\epsilon$. Signal-energy also relies on interpolation by $x_\epsilon$, but is less sensitive to position error as it represents the level

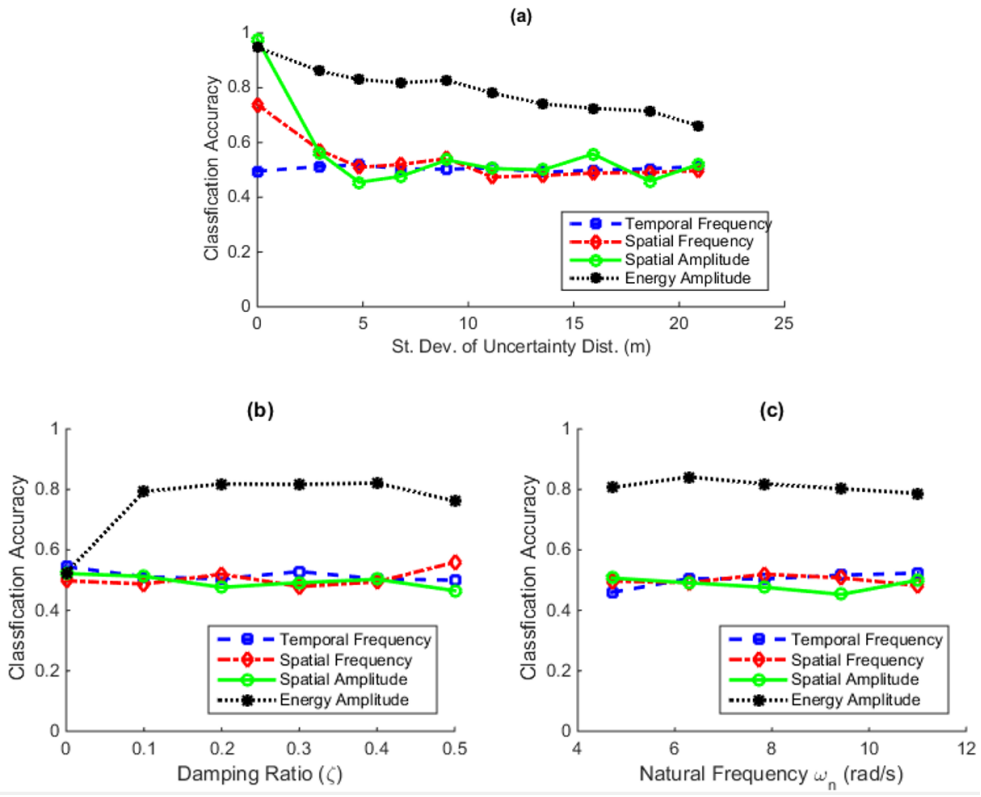



Fig. 3.7: Classification accuracy for spike change. (a) Effect of position uncertainty for oscillator with $\zeta = 0.2$ and $\omega_n = 2.5\pi$ rad/s. (b) Effect of varying damping ratio while uncertainty $\sigma = 7$m and $\omega_n = 2.5\pi$ rad/s. (c) Effect of varying natural frequency while $\zeta = 0.2$ and uncertainty $\sigma = 7$m.

of roughness or excitation caused by the track, rather than the specific state of the oscillator (it is always positive independent of the oscillator).

We also investigated the effect of varying the damping ratio (Fig. 3.7b) and the natural frequency (Fig. 3.7c) if the location uncertainty is kept constant at $\sigma = 7m$. For this level of position uncertainty, signal-energy is the only feature with strong discriminative power. When the damping ratio was zero, the oscillations kept growing over the course of a pass, so the localized track change are undetectable even from the signal-energy feature. If the damping ratio is large, i.e. $\zeta = 0.5$, the changes are so localized that they become difficult to detect given the position uncertainty. Thus, a moderate level of damping ($\zeta \approx 0.2$) appears preferable; this corresponds conveniently with the observed damping ratio of the train. For variation of the natural

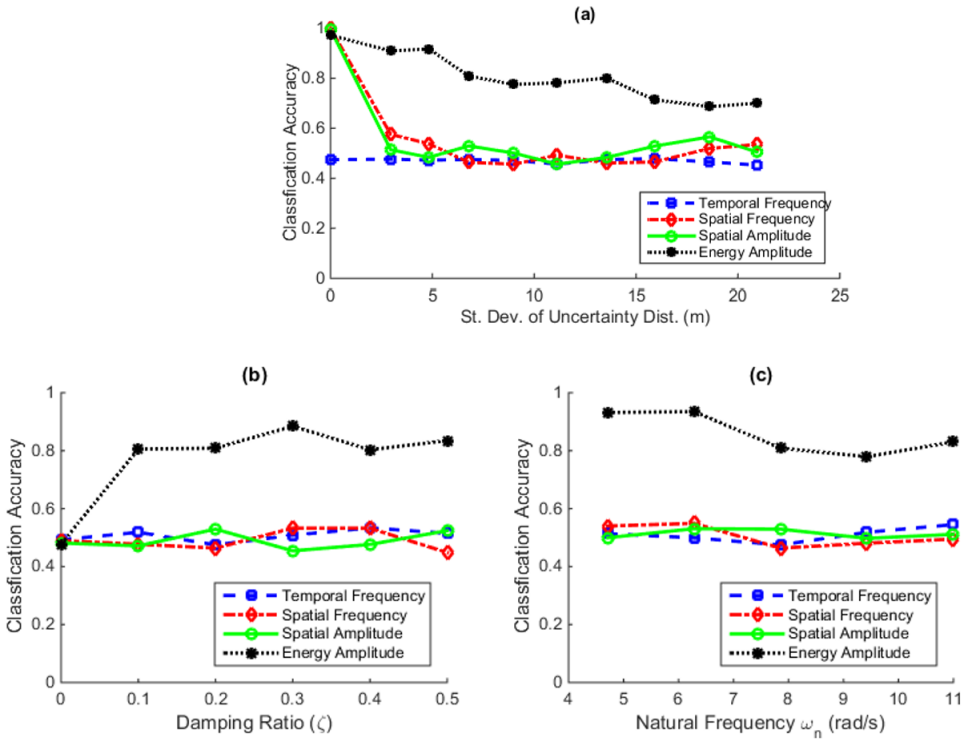



Fig. 3.8: Classification accuracy for tamping change. (a) Effect of position uncertainty for oscillator with $\zeta = 0.2$ and $\omega_n = 2.5\pi$ rad/s. (b) Effect of varying damping ratio while uncertainty $\sigma = 7$m and $\omega_n = 2.5\pi$ rad/s. (c) Effect of varying natural frequency while $\zeta = 0.2$ and uncertainty $\sigma = 7$m.

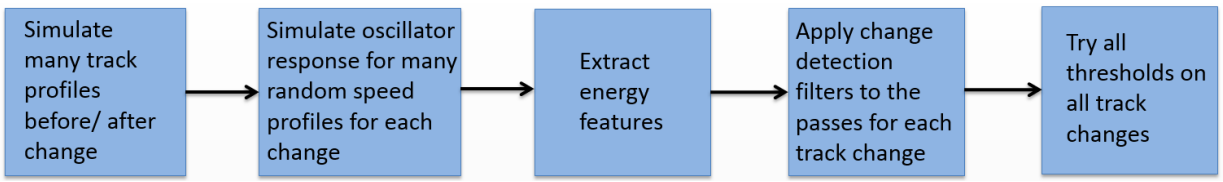



Fig. 3.9: Flow chart of change detection simulation. Note that this procedure is repeated for different track changes and different levels of position uncertainty.

frequency (Fig. 3.8c), oscillator stiffness has relatively little effect on the accuracy over the range of values considered. In the case of tamping in Fig. 3.8c, a stiffer oscillator leads to more localized changes and because tamping is effectively many small smoothing changes over a section of track, these localized changes are more difficult to detect.

Overall, the signal-energy feature outperforms the other features as it is robust to position uncertainty for both types of track changes. This is particularly convenient as the signal-energy feature could potentially be universally applied in this type of monitoring system. In industry, the most common parameter for measuring track geometry is the standard deviation of the track profile, and signal-energy of the vibration signal is related to this parameter, as both involve squaring their respective data points. The relationship between the vibration signal and the track profile, given in Eq. 3.1, is linear and time invariant, which means that the frequencies of the response are simply scaled versions of the frequencies of the roughness. The strong performance of signal-energy shows that the traditional parameters for track monitoring can inspire new features that are useful for our statistical track monitoring system.

### 3.2.3 Change Detection Simulation

Having established that among the different criteria examined, signal-energy represents the most robust feature, in this section we explore how we can achieve our second goal: to determine when a change occurs in the tracks.

We conducted a simulation to study change detection approaches by following the flow chart shown in Fig. 3.9. First we generated many passes of the train before and after a track change

(varying the speed profile as in the previous simulation study). We computed the signal-energy feature from the vibration signal, then extracted the data at a particular track location across all $N$ passes. Although a number of change-detection approaches exist, many of them are designed for finding statistical change in a scalar quantity. In this application, we compared data from different passes of a moving vehicle, so each pass was a vector. To format the data in a way that we could use these general change detection approaches, we considered the value of the signal-energy feature at a specific location on the track, $i$, over all passes, $n$, by building a matrix, $F(n, i)$ where each feature vector $f_n^T$ was a row in the matrix (the superscript $T$ indicates the transpose of the vector).

$$F(n,i) = \begin{bmatrix} f_1^T \\ f_2^T \\ \vdots \\ f_n^T \\ \vdots \\ f_N^T \end{bmatrix} \tag{3.5}$$

Using this formulation, we experimented with three different change detection filters: cumulative summation control chart (CUSUM), generalized likelihood ratio (GLR) and a Haar filter. All three change-detection methods effectively detect when there is a change in the mean-value of the feature at a particular location of track compared to that value for a window (set number) of previous passes. Details of each will be presented briefly to explain how they are applied within the two dimensional feature matrix.

1. Cumulative summation control chart (CUSUM) estimates the mean value over a window of previous passes (known as a sliding window), then compares the current (or most recent value) to the historic mean [27]. If there is a succession of passes all deviating from the mean in one direction, it will trigger a detection event because it is likely that the mean has changed. Mathematically, the mean, $\hat{\Theta}$, is estimated from historical data

over a window of $w + 1$ previous points,

$$\hat{\Theta}(n, i) = \frac{1}{w+1} \sum_{k=n-w}^{n} F(k, i). \tag{3.6}$$

We then find how the data from the current pass differs from this mean,

$$\delta(n, i) = F(k, i) - \hat{\Theta}(n, i. \tag{3.7}$$

Finally we add this difference, $\delta$, to a running sum of the differences,

$$g(n, i) = g(n - 1, i) + \delta(n, i). \tag{3.8}$$

If the data varies from the mean consistently in one direction (positive or negative) intuitively the data has changed because the mean no longer represents the data. We say a change has occurred when the magnitude of $g$ exceeds a threshold, $h$, $|g(n, i)| > h$.

2.The generalized likelihood ratio (GLR) looks for a change in the underlying distribution, and quantifies the log-likelihood that the recent data is derived from the historical distribution [27]. If the recent data appears dissimilar from the historical data, then the approach will trigger a detection event. Mathematically, let us call the first window of data $y_0$,

$$y_0(n, i) = F(n - w - 1 : n - 1, i), \tag{3.9}$$

and the second window of data $y_1$,

$$y_1(n, i) = F(n : n + w, i). \tag{3.10}$$

$y_0$ has some distribution, let us define it as $H_0$, and $y_1$ has some distribution $H_1$. We want to calculate the likelihood ($L[\cdot]$) that $y_1$ comes from the same distribution as $y_0$, $H_0$,

versus the likelihood it comes from $H_1$,

$$g(n,i) = 2log\frac{L[y_1(n,i)|H_1(n,i)]}{L[y_1(n,i)|H_0(n,i)]}. \tag{3.11}$$

We say a change has occurred when $g$ exceeds a threshold, $h$, i.e $g(n,i) > h$. In other words, beyond some threshold, the data is so much more likely to have come from a new distribution than from the historical distribution that we say a change in the data has occurred.

3. The Haar filter finds the difference between the sum of recent data and the sum of historical data, triggering a detection event if the difference is too large. Mathematically, the difference between the two windows of data of width $w+1$ can be written as

$$g(n,i) = \left| \sum_{k=n}^{n+w} F(k,i) - \sum_{k=n-w-1}^{n-1} F(k,i) \right| \tag{3.12}$$

As in previous approaches, we say a change has occurred when $g(n,i) > h$. This approach tends to find locations where a step change has occurred between the two windows of data.

Each of the approaches relies on a sliding window of data: shorter-sized windows have the potential for detecting changes more rapidly after they occur, while longer windows allow for greater statistical significance; here we chose a window size of 20 passes, which would allow our system to detect a change within a week (assuming a few passes over the track per day). The results of applying each of these filters to the simulated changes can be seen in Fig. 3.10 as will be discussed in more detail in the next section.

### 3.2.4 Change Detection Simulation Results

In this section, we discuss the performance of the three change-detection approaches described previously. Results of each approach applied to one example of each type of simulated track

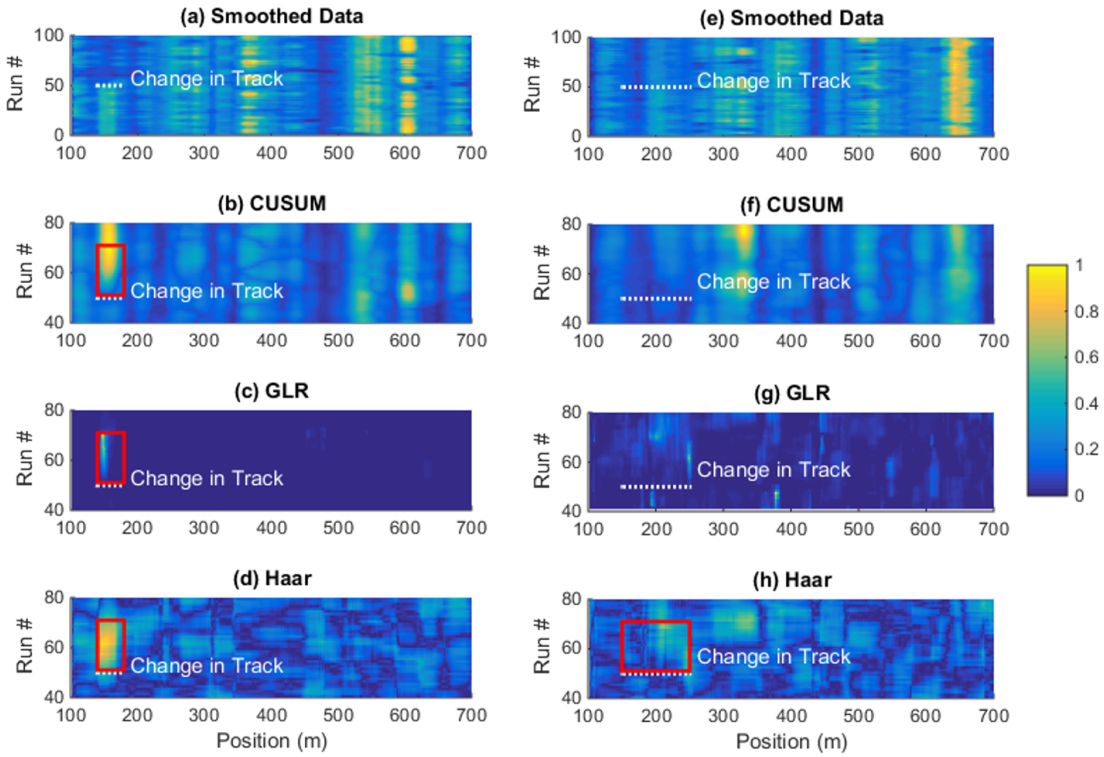



Fig. 3.10: The change detection filters applied to the data. (a) one example of a track replacement change showing the energy feature where the spike (due to a broken track) is removed at 150m, (b-d) show three change detection approaches applied to this track change, (e) shows a tamping change between 150 and 250m using the signal-energy feature, while (f-h) show the three change detection approaches applied to this data. Values in each figure (shown in color) have been normalized on [0,1]. The red-boxes indicate the true-positive events that could be detected with an ideal threshold. There are no boxes in (f) and (g) as the methods failed to detect the change. The data shown in this figure has no position uncertainty.

change are shown in Fig. 3.10. Successful change-detection filters should produce a high value after the change occurs at the same location in the track, and successful detection has been indicated with a red-box. Note that the change is not necessarily detected immediately, as it takes several passes after the change to build up statistical significance that a change has indeed occurred. All three approaches detected the track change on the left in Fig. 3.10b-d (with an ideal threshold), while only the Haar filter was able to detect the tamping change in Fig. 3.10h.

CUSUM is perhaps the most versatile of the methods, as it could detect a gradual change in the mean over time, while GLR and Haar work best for abrupt changes in the data. In our case, we assume the track change occurs over a short period of time, so the change is complete before the train passes over that section of track again. Given this assumption, GLR and Haar might be expected to perform better than CUSUM. GLR is more complex than the Haar filter as it involves not only the mean, but also the standard deviation in calculating the distribution of the data before and after the change. A change in variance could trigger a detection event, but it is unlikely in our application that high variance would be due to an infrastructure change, so GLR is less robust for this application. It is the simplicity of the Haar filter that enables it to consistently detect tamping along the section of track where it was simulated in Fig. 3.10h.

Fig. 3.9 shows just one example of each type of change, but we quantified the success of each approach over the 100 changes we simulated. We defined a true positive as a change detected within 20 passes and within 25m of a true change. A false positive was a change detected outside of this window. We report false positives and false negatives (true positives that were not detected) for a range of thresholds in Fig. 3.11a, because selecting the appropriate threshold itself can be difficult. In this chapter, we focus on which change detection approach is best suited for detecting track changes, so we report the error level assuming the ideal threshold was selected. Assuming false positives, FP, and false negatives, FN, are equally bad, we select the threshold level that will minimize the larger of the two errors, which occurs where the false negative and false positive curves intersect, as shown in Fig. 3.11a. In other words, we want to select a threshold, h, where

$$h = \underset{h}{\text{argmin}}(\max[FN, FP]). \tag{3.13}$$

In total, for understanding the trade-off in the change detection approaches, we simulated 200,000 passes (2 types of track changes $\times$ 100 track changes $\times$ 100 passes $\times$ 10 levels of uncertainty). While in the feature selection study (section 3.2.1), we used only 3 track profiles (track change, tamping change and toy-roughness) in this case we simulate 100 track profiles for track changes and 100 track profiles for tamping changes. Also, in the feature selection study (Section 3.2.1), we considered a range of natural frequencies and damping ratios. In this simulation study we use only the damping ratio and natural frequency values we have observed in our train system ($\omega_n = 2.5\pi$ rad/s, $\zeta = 0.2$).

In Fig. 3.11b the lowest error (i.e. ideal threshold) is reported for all change detection approaches on the track change simulation. While the error increased for all methods as position uncertainty increases, the Haar filter consistently performed the best. Fig. 3.11c shows the same error quantification but for the tamping change simulation, and again the Haar filters outperforms the other methods. Because tamping occurs over a larger section of track, position uncertainty matters less as we saw in the classification results shown in Fig. 3.8a, so it is consistent that the Haar filter remains relatively unaffected by position uncertainty.

## 3.3 Validation on the Light-Rail Vehicle Dataset

The previous sections provide a general understanding of which features and detection filters work well to detect changes in a rough track from the dynamic response of a travelling oscillator. In this section, we investigate whether the same findings hold true for our test-system deployed on a light-rail vehicle described previously in section 2.

Our simulations aimed to model the fundamental natural frequency of the train; a comparison between the simulated data and the collected data is shown in Fig. 3.12. While there is more

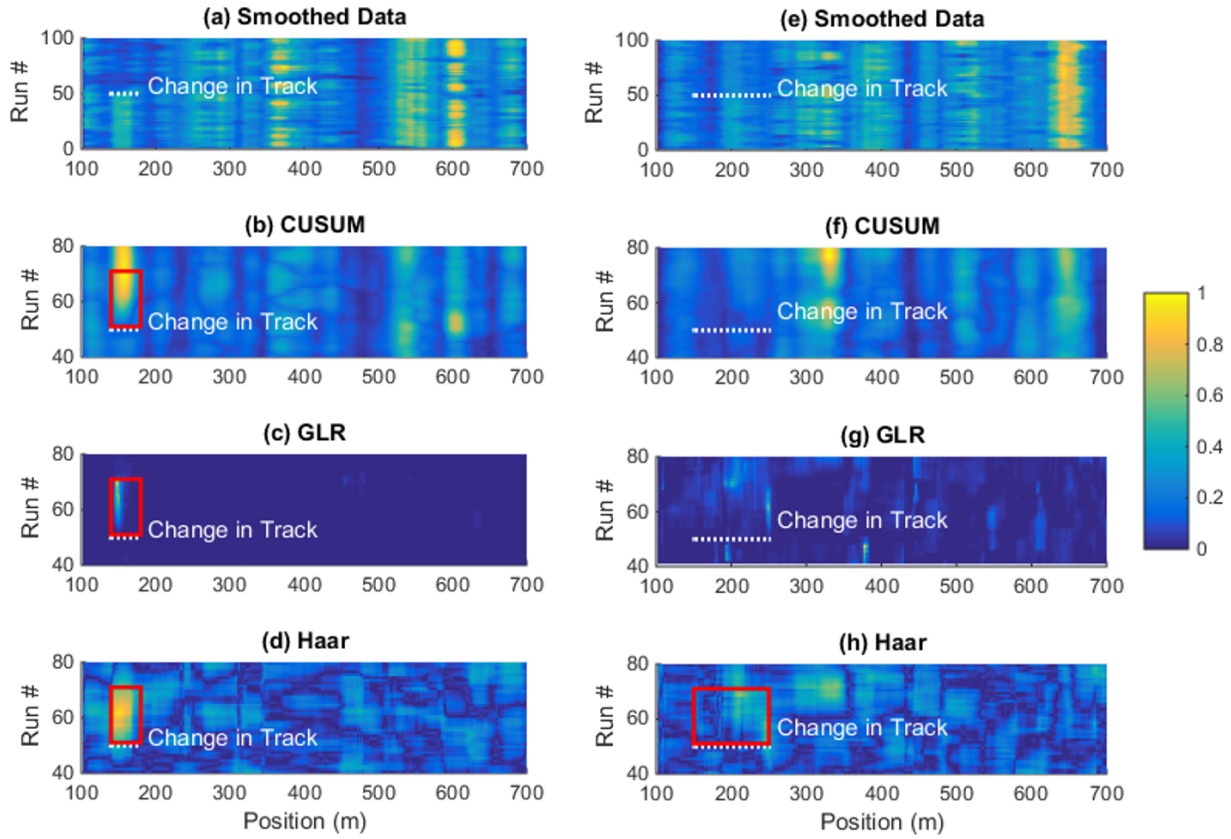



Fig. 3.11: Change detection results from the simulation. (a) A typical plot of false negatives (FN) and false positives (FP) as the threshold is varied, shown with data for the 100 examples of this type. In this case, the plot is shown for CUSUM with no position uncertainty, and where the two lines cross, there is 43% error of both types. The data has been normalized on [0,1] so the threshold spans the whole range. (b) The minimum error for all three approaches and all position uncertainty levels for the spike change. (c) The minimum error for the tamping change.

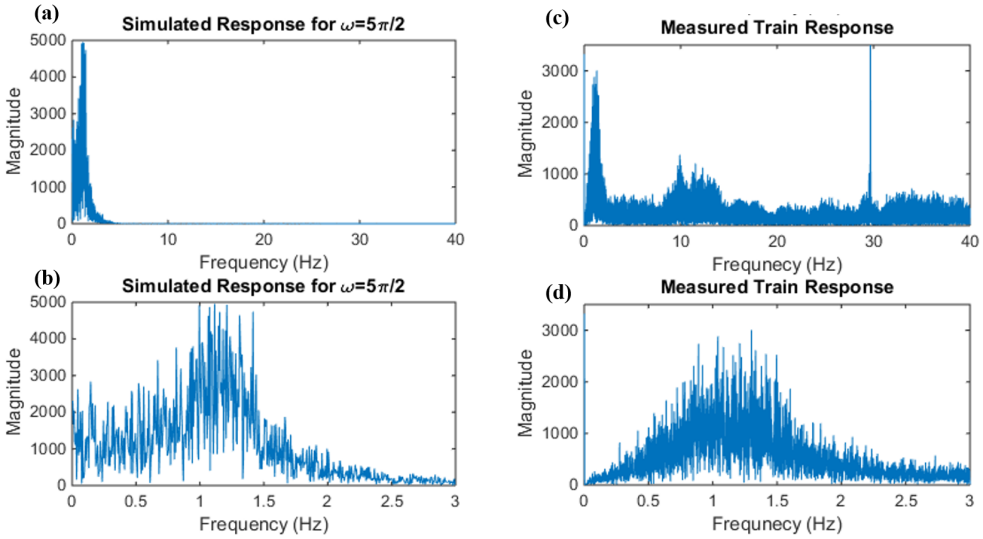



Fig. 3.12: Frequency response of the oscillator (in simulation) and operational train. (a-b) Response of the oscillator at two different scales. (c-d) Response of the train at two different scales. In (c), the measured response has a strong narrow peak at 30Hz, which is a resonance frequency of the 60 Hz electricity used inside the cabin. In general, the dominant response of the train is around 1.25 Hz, which is the same as the fundamental natural frequency of the oscillator we have used. Note that the amplitude of the lowest frequencies of the measured response (0 - 0.5 Hz) are reduced by the sensor; it is likely that the values below 0.5 Hz should be higher, but the accelerometer, which is a shear piezo-electric, has low sensitivity at such low frequencies.

noise at high frequencies in the measured data (Fig. 3.12c), both the oscillator and train have a natural frequency of around 1.25 Hz. Our simple oscillator does not attempt to model all the complexities of the train system. For example, the air-conditioning system is one of the biggest sources of noise within the train cabin. However, the train's primary response to the track roughness does appear similar to that of a single-degree-of-freedom oscillator.

### 3.3.1 Track Change in the Light-Rail Dataset

In September 2014, the owner of the light-rail system replaced the track in an old road-crossing. We use this known maintenance activity, where faulty track was replaced with good track, to test our signal analysis pipeline. Fig. 3.13a and b show two characteristic passes of the train over the 1km section of track. In the second pass the train stops at two stations, the first at 200m and the second at 520m, whereas in the first pass it only stops at one of the stations (at 200m). The train

stops when there are passengers to pick up or let off, which poses a challenge when comparing data between passes. One hundred passes over this section of track are shown in Fig. 3.13c, where the color indicates the value of accelerations recorded. Although difficult to see in Fig. 3.13, there is a high frequency vibration event when crossing a road at 220m, an event which no longer occurs after the 50th pass, when the repair is done. It is much easier to see the change in Fig. 3.14 because signal-energy is a better indicator of track state. As the train crosses the road at 220m, high signal-energy can be seen in Fig. 3.14a. This spike is absent in Fig. 3.14b or after pass #50 in Fig. 3.14c due to the repair.

Note as well that the first station stop in Fig. 3.13(a) and Fig. 3.14(a) occurs just after 200m, whereas in (b) it occurs just before 200m. This is likely due to the orientation of the train. The GPS is at one end of the train, and the train always stops at the same location within the station. If the orientation of the car changes, the position can differ by 27m, the length of the car. Due to the sensor's location on the train, it will interrogate the tracks near the stations at different speeds. A sensor at the front of the train will travel quickly over tracks at the beginning of the station, and slowly over tracks towards the end of the station; the reverse is true for a sensor at the back of the train, adding to the difficulty of comparing data directly between different runs.

As in the simulation, we extracted different features from the data and used classification to determine which features allow the infrastructure change to be detected most readily. In a binary classification between the two states, we achieve 91% accuracy in Fig. 3.15a, drawing data from a 500m section of track shown in Fig. 3.15c. The second best feature appears to be the temporal-frequency feature. However, as we saw in the simulation, this feature would not be expected to detect track changes given the train's variable speed. It is far more likely that the feature is detecting temperature differences between the two classes as the data was collected over a year, and Pittsburgh is a temperate region with significant temperature variability. Temperature affects the data in a number of ways: for example, we have observed higher noise levels inside the train cabin during the summer months due to the vibrations of the air conditioning system. More subtle effects include slight stiffening of the steel in cold weather, which affects train

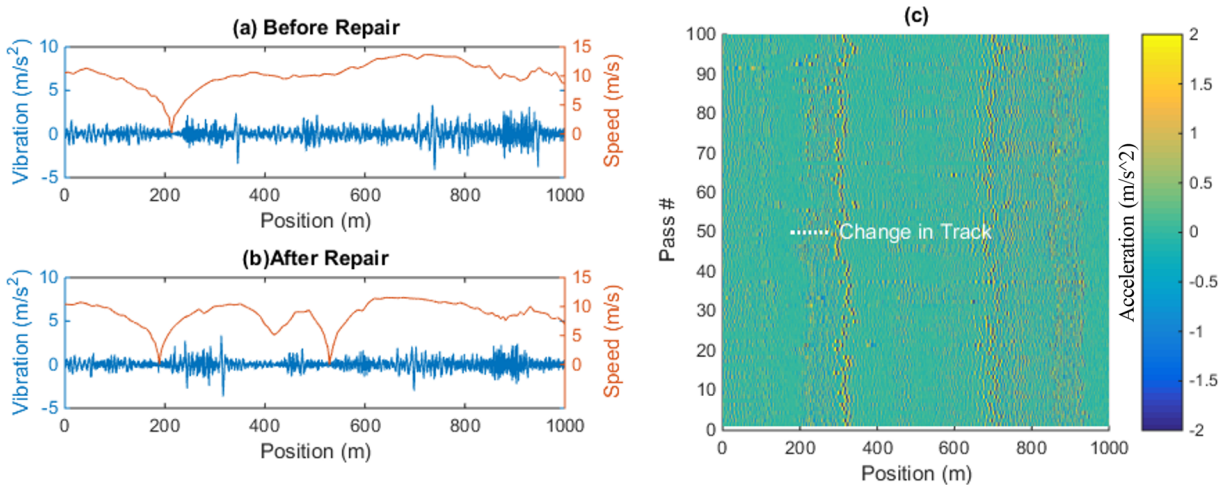



Fig. 3.13: Spatial signal. (a) A pass before repair showing both the train speed and vertical vibrations from a sensor inside the train. (b) A pass after repair. (c) 50 passes before and 50 passes after the repair, where each pass is a horizontal line and the color indicates the instantaneous acceleration. With the spatial-amplitude feature, the track change is nearly impossible to see.

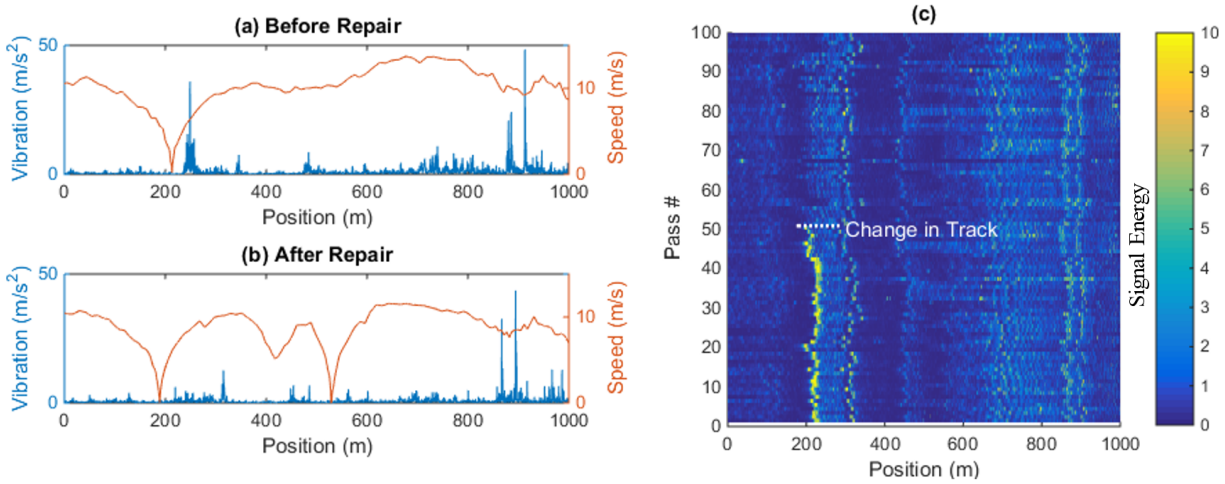



Fig. 3.14: Signal-energy. (a) A pass before repair showing both the train speed and vertical vibrations from a sensor inside the train. (b) A pass after repair. (c) 50 passes before and 50 passes after the repair, where each pass is a horizontal line and the color indicates the magnitude of the signal-energy feature. With the signal-energy feature, the track change is clearly visible.

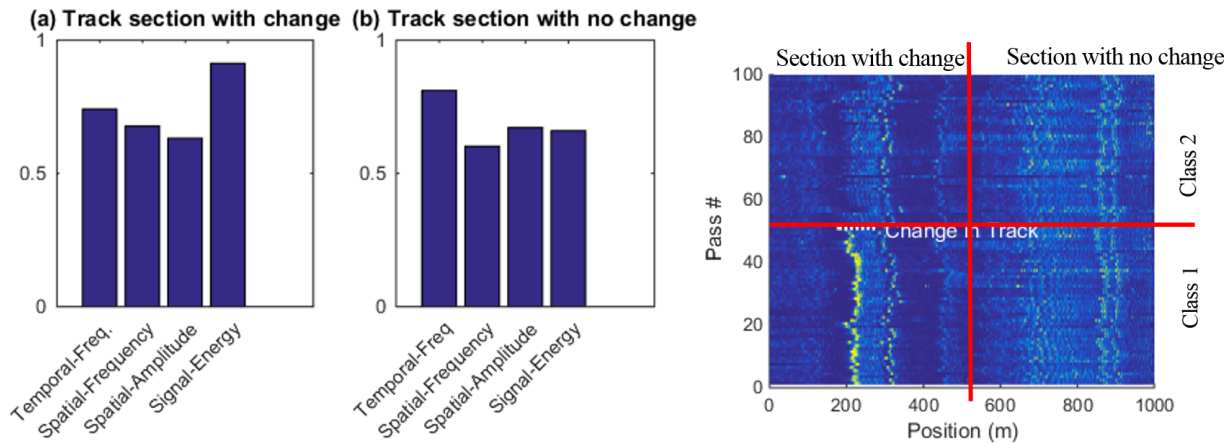



Fig. 3.15: Classification accuracy of a 500m section (a) of a track where rails were replaced, and (b) of a track nearby where no work was done. (c) Shows where the data for the classification was drawn from and the two classes used in the binary classification. High classification accuracy means the classes are separable, and 50% accuracy means the classes are not separable leading to random classification. Signal-energy is sensitive to track changes because it achieves 91% accuracy when there is a track change, and is close to random when there are no track changes, meaning it is not classifying based on environmental factors.

components like the wheels. That temperature impacts the signal is supported by the results shown in Fig. 3.15b where classification results are presented for an adjacent 500m section of track where no work was done. Classification using the frequency-feature allows for 78% accuracy despite the fact that the infrastructure has not changed, whereas classification using the signal-energy feature allows for 67% accuracy (close to random) meaning it is more robust to environmental conditions. Again we use 75% of the data for training, 25% for testing and 50-fold cross validation as in the classification in the simulation study.

### 3.3.2 Tamping Change in the Light-Rail Dataset

Tamping is an important maintenance procedure used to improve track geometry. The tamping machine measures the profile of the track, then adjusts the ballast below the track to produce a smoother, safer ride. One future goal could be to use our data to optimize tamping schedules. At this early stage though, we are most interested in identifying which features are sensitive

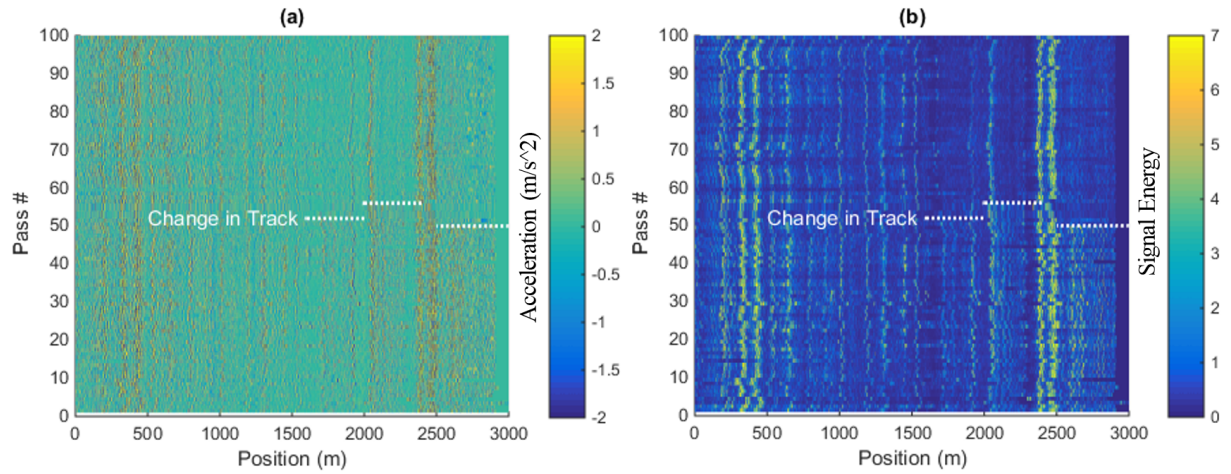



Fig. 3.16: Region of track where tamping occurs with (a) showing the spatial-amplitude feature and (b) showing the signal-energy feature. The tamping maintenance was done three times on three separate days due to the limitations on how much work the tamping machine can do per day. Note that the peaks between 2400 and 2500m (and between 400 and 500m) are due to switchgear in the tracks.

to tamping, and which detection approach works best for this feature. In the summer of 2014, three 500m sections of track were tamped as shown in Fig. 3.16, starting at 1500m, 2000m and 2500m. Unlike the track change in the previous section, the change due to tamping is subtle, but occurs over a much larger section of track. Notice how in the regions with no tamping, the signal is relatively consistent over all 100 passes, while in the areas with tamping, there is more signal-energy (i.e. more bumpy ride) before tamping, and less signal-energy (i.e. smoother ride) after tamping.

We tested all four features in a binary classification to discriminate between before and after a tamping event; we found that the signal-energy feature is the best for detecting this change. The classes in the classification and the region from which the data was drawn are shown in Fig. 3.17c. Signal-energy achieves 90% classification accuracy (shown in Fig. 3.17a) with a change, and $< 60\%$ accuracy (effectively random) in Fig. 3.17b when there is no change. This means that the signal-energy feature is classifying due to the change in the track and not an environmental variable. Frequency and spatial-frequency features on the other hand report 70% accuracy if there is a change (as in Fig. 3.17a), and 70% with no change (as in Fig. 3.17b). We can assume

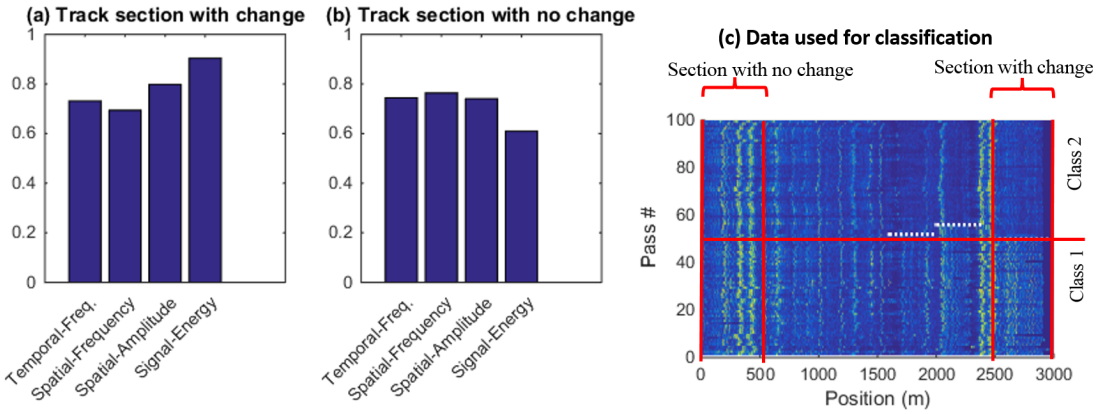



Fig. 3.17: Classification accuracy of a 500m section (a) of a track where tamping work was done, and (b) of a track nearby where no work was done. As in Fig. 3.15, the high accuracy of the signal-energy feature where there is track work shows it is sensitive to infrastructure changes, and the low accuracy (almost random) where no work has been done shows it is robust to environmental variables. (c) We show the data used for the classification in relation to Fig. 3.16b, both in terms of which 500m sections of track were used, and how the two classes in the binary classification were defined.

the discriminative power stems from temperature changes and not from infrastructure changes. Note that for consistency, we only classified data from a 500m section of track as in the previous section, but the tamping occurs over a larger region. Classification accuracy can be slightly increased by considering a larger section of track.

### 3.3.3 Detecting Change in the Light-Rail Dataset

The ultimate goal of the project is to automatically detect changes in the infrastructure. As such, change detection is a vital step. Although we have collected hundreds of passes through the rail-network, we still have relatively few known infrastructure changes. Thus, we do not have sufficient data to rigorously test different threshold levels as we did in the simulation. Instead, we applied the change detection methods on the operational data and present the results with a basic threshold applied in Fig. 3.18. This figure mirrors the results in Fig. 3.10, where the raw data and all three approaches are shown. Fig. 3.18a shows the signal-energy feature for the track replacement change, and Fig.e 3.18b for the tamping changes. Fig. 3.18 c-h show the results

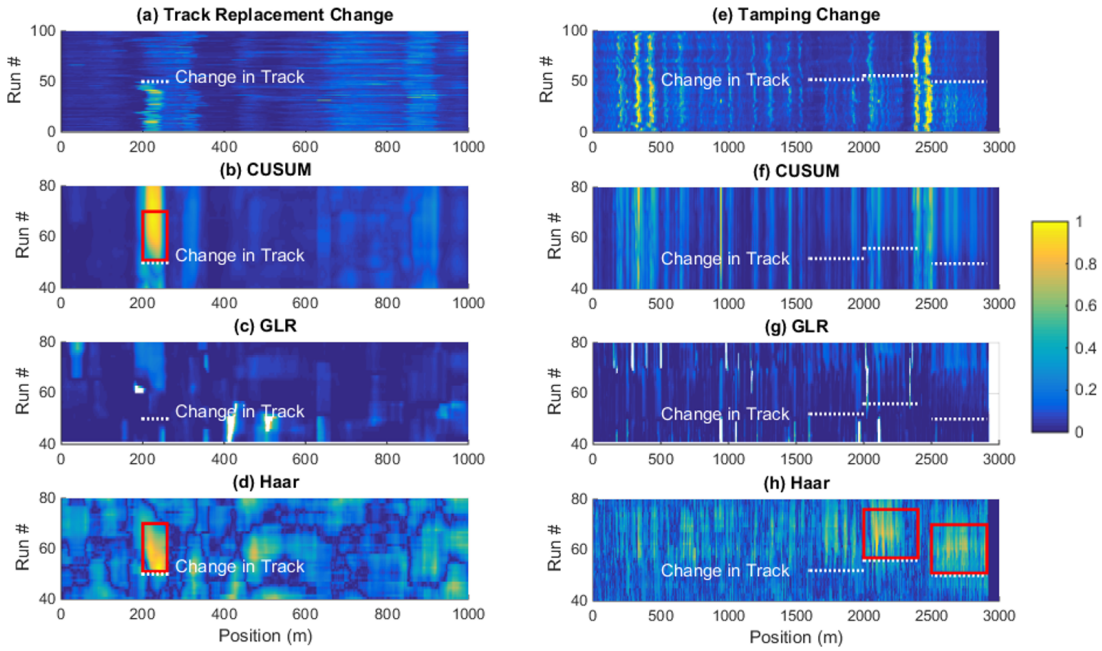



Fig. 3.18: Change detection on the light-rail dataset for the track replacement change (a)-(d) and tamping change (e)-(h). Panels (a) and (e) show the signal-energy feature while (b)-(d) and (f)-(h) show the respective change detection techniques. The red-boxes indicate true-positive changes that could be detected with an appropriate threshold.

of applying the three change detection approaches to each of these two changes, in which the red-boxes indicate the change-detection approach was successful.

For both types of track changes on the light-rail dataset, the Haar filter performed the best, reliably detecting both track and tamping changes (Fig. 3.18d and h) as predicted by our simulation results. For the track-type change, CUSUM and Haar filter (Fig. 3.18b and Fig. 3.18d respectively) could detect a change with zero error given the correct threshold. For the tamping change, CUSUM and GLR fail to detect the change at all. CUSUM fails because the variability in the sections with high energy (like the switchgear at 2500m) are larger than the change from tamping itself, which occurs on low-energy sections of track. It is unclear why the GLR method has false-positives, although it could be due to the GLR's sensitivity to changes in variance as discussed earlier. It is important to note that the Haar filter succeeds in our original goal: detecting changes relative to historical behavior, rather than simply detecting areas of the track that produce large vibrations. The high values around 2500m in Fig. 3.18a due to switchgear do not

affect the detection in Fig. 3.18h.

## 3.4 Discussion

To build meaningful implicit models, new features must be extracted from the vibration signal that are able to characterize the rails despite the filtering from the train's suspension, the train's changing speed, and the noise in the GPS location. We tested four features, temporal-frequency, spatial-frequency, spatial-domain and signal-energy on simulated data and found signal-energy to be the best feature for detecting both track changes and tamping changes. We then tested the same features on the operational data from the light-rail system, and showed that signal-energy was the most sensitive to infrastructure changes and the least sensitive to environmental variability. Both in our simulations and in the operational data, we found that frequency-based features do not work well, despite their widespread use in many structural health-monitoring studies.

In addition to feature extraction, we studied a variety of unsupervised change detection approaches, including CUSUM, GLR, and the Haar filter. We found that the Haar filter outperformed the other approaches on both the simulated data and operational data, as it was particularly robust to position uncertainty.

From the results of this chapter, it appears that implicit modeling could be an effective tool for train-based track monitoring. We were able to detect changes on an operational system using just a single sensor on a train in revenue service.

## 3.5 Future Work

While a systematic method for comparing different features and change detection methods has been presented, only four types of features are tested and three types of change detection methods. Future work should look at additional features, like wavelets, and additional change de-

tection approaches, like the KL divergence. And while we studied the applicability of these approaches for two types of track changes, a much larger collection of track changes would be useful. Some ideas on how larger data-sets could be collected are detailed in the Future Work section at the end of thesis.

Finally, this chapter has presented statistical approaches that largely ignore the underlying physics of the problem; as will be discussed in the next chapter, hybrid approaches that combine statistics with a physics-based approach could provide additional insight.

# Chapter 4

# Explicit Models

An alternative to the implicit modeling presented in the previous chapter is modeling the track profile itself, what we call explicit modeling. Determining this profile requires solving an inverse problem. The collected data describes the dynamic response of the train, which is caused in part by the track profile itself. However, this inverse problem is ill-posed. In this chapter we propose a novel analysis technique for solving for the track profile by exploiting the sparsity inherent in train-vibration data. This sparsity is based on the observation that large vertical train vibrations typically involve the excitation of the train's main suspension due to track joints, switchgear, or other discrete hardware. Rather than trying to model the entire rail profile, in this chapter we examine a sparse approach to solving an inverse problem where (1) the roughness is constrained to a discrete and limited set of "bumps"; and (2) the train system is idealized as a damped oscillator that models the train's main suspension. We use an expectation maximization (EM) approach to iteratively solve for the track profile and the train system properties, using orthogonal matching pursuit (OMP) to find the sparse approximation within each step. By enforcing sparsity, the inverse problem is well posed and the train's position can be found relative to the sparse bumps, thus reducing the uncertainty in the GPS data. We validate the sparse approach on two sections of track monitored from an operational train over a 16 month period of time, one where track changes did not occur during this period and another where changes did occur. We show

that this approach can not only detect when track changes occur, but also offers insight into the type of such changes.

## 4.1 Introduction

In this chapter, we propose solving the inverse problem to find the track profile, but we constrain the problem to make it more stable. The constraint comes from the observation that the train's suspension is typically activated by a few large bumps in the track. Thus we aim to find these discrete "bumps" and do so by enforcing sparsity in the estimated track profile. The method is effectively a hybrid of the implicit model and explicit model mentioned earlier; although we solve an inverse problem, due to the sparse constraint, the found roughness is an abstraction of the actual roughness, similar to a feature in an implicit model.

Enforcing sparsity in the track has a number of benefits [23, 48, 65, 71]: (1) the problem is constrained so some properties of the train system can be found without making the problem ill-posed, (2) the discrete bump locations can be used to locate the train, overcoming GPS error, and (3) the size of the bumps are useful low-dimensional features for detecting the significance of changes in the track.

(1) We characterize the train system while constraining it to the physics of the problem. We require that the transfer function correspond to a simple damped oscillator. When enforcing this condition, the parameters found relate to the stiffness and damping ratio of the main suspension, between the wheel truck and the train chassis. This makes physical sense because when a large bump in the tracks excites the train, the largest displacement is in the primary suspension, and we require that the approximate roughness model only large bumps. Unlike previous methods in which the parameters of the train must be known *a priori*, our approach solves for the train properties in the process of solving for the track profile.

(2) We locate the train using a GPS antenna, but due to overhead interference and other

factors, the position-error can exceed 10m. This level of error makes it challenging to compare data between passes. Train localization has been studied in the literature, primarily for collision avoidance [56]. Some researchers have proposed using track features to help localize trains in this context [29, 30]. For monitoring purposes, precise localization is of paramount importance. Enforcing sparsity facilitates localization because the position of the train can be found relative to the sparse bumps.

(3) Finally, the size of the bumps can be used to determine whether the tracks have changed or deteriorated. If a complete rail profile were to be calculated, the high dimensionality of the data (proportional to track length) would make robust change detection more challenging; the low-dimensionality of bump height as a feature simplifies change detection, as will be shown in Section 4.4.

We show the application of this sparse approach on data collected from the same light-rail vehicle as in Chapter 3, over a 16-month period. We explore the consistency of the method over time, both in terms of identifying the parameters of the train, and the track profile. Finally, we study how consistently the proposed approach can identify the same bumps in the track, both when the tracks remain unchanged over a period of time, and when the tracks change.

## 4.2 Problem Description

As the train travels along the tracks, the suspension filters the roughness of the track. Let $u$ be the vertical displacement of the train, and $r$ be the vertical track profile. If the train suspension is modeled as linear[1], the relationship between the acceleration of the train $\ddot{u}$ and the track profile $r$ can be written as

$$\ddot{u} = h * r. \tag{4.1}$$

[1]The train's suspension is not purely linear; for example, the interface between the wheel and the rail is often modeled as a non-linear Hertzian spring, but for the vast majority of analyses, this slight non-linearity can be approximated as linear [6].

Here, $h$ is the impulse acceleration response of the train system and $*$ is the convolution operator. $\ddot{u}$, $r$, and $h$ are all $p$-dimensional vectors corresponding to $p$ points in time. If $h$ and $r$ are known, we refer to this as the forward formulation for evaluating $\ddot{u}$. On the other hand, if $h$ and $\ddot{u}$ are known but $r$ is not, Eq. 4.1 represents an inverse problem in which the roughness profile can be found through deconvolution. In the frequency domain (denoted by the hat symbol), this is written as:

$$\hat{r} = \frac{\hat{\ddot{u}}}{\hat{h}}. \tag{4.2}$$

Most studies that try to model the track profile explicitly use this inverse formulation.

Real et al. [55] estimate the transfer function $\hat{h}$ by modeling the train as a two degree-of-freedom oscillator. Directly solving Eq. (4.2) can be unstable, as the transfer function $\hat{h}$ of an oscillator approaches zero for high frequencies, while the measured data $\hat{\ddot{u}}$ can have non-zero values at high frequencies. This potentially leads to amplification of high-frequency data. Real et al. address this instability by bandpass filtering the measured data. The challenge is how to define an appropriate bandpass filter; in their paper, they fit a filter using one section of track, but that filter appears not to work well on other track sections. That the filter is not generalizable could be explained by the fact that the train system is more complicated than the two-degree-of-freedom model they use. Thus the true system response has some non-zero values at high frequency, and the filter is an attempt to compensate for an oversimplified model for $\hat{h}$.

O'Brien et al. [49] use a different approach to avoid the instability inherent in dividing by the transfer function. They use the cross-entropy method to generate a family of roughness profiles, then use the forward formulation in Eq. (4.1) to calculate the response that the roughness would generate. They then repeat the process, each time attempting to minimize the mean squared error between the generated response and the measured response. This iterative process is computationally expensive, requiring several hours of processing to analyze a simulated 20m section of track. Furthermore, this approach has not yet been tested on operational data, and its sensitivity to noise has not yet been studied.

In our approach, we aim to preserve the computational efficiency of directly solving the inverse problem, while constraining the problem sufficiently so that a stable solution is found even if the exact transfer function is not known. Without knowledge of the precise transfer function, we cannot find the precise track profile, but as a trade-off, we aim to find only the location and magnitude of the "bumps" or irregularities along the track. As will be described in the following section, we achieve this by enforcing sparsity.

## 4.3 Algorithm

We want to find the train's transfer function and the track roughness that best approximate the measured vertical accelerometer data. This can be written as

$$\min_{h,r} \quad ||\hat{\ddot{u}} - \hat{H}\hat{r}||_2 \tag{4.3}$$

where $||\cdot||_2$ is the $\ell_2$ norm, $\hat{\ddot{u}}$ and $\hat{r}$ are $p$-dimensional vectors corresponding to the $p$ observations, and $\hat{H}$ is a $p \times p$ diagonal matrix of the transfer function,

$$\hat{H} = \begin{bmatrix} \hat{h}[1] & 0 & \dots & 0 \\ 0 & \hat{h}[2] & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{h}[p] \end{bmatrix}. \tag{4.4}$$

Without constraints, this problem is ill-posed and has trivial solutions: for example, $\hat{H}$ could be the identity matrix and $\hat{r} = \hat{\ddot{u}}$, which would lead to a perfect solution. But nothing would be learned either about the nature of the track roughness or about the train system. Instead we constrain the track profile to a set of $n$ discrete bumps, and the transfer function to model a single degree of freedom oscillator as seen in Fig. 4.1. It is important to note that the train is a far more complex system than the single degree-of-freedom oscillator used in our formulation; we

do not attempt to accurately represent the complete behavior of the train. Instead, by adding physics-based constraints, we aim to imbue some meaning into the optimization problem and to learn about one component of the train.

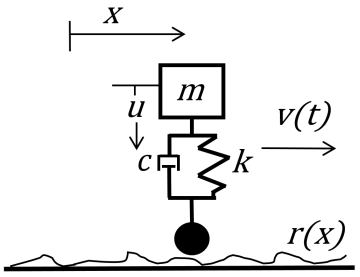



Fig. 4.1: A representation of the train as a single-degree of-freedom oscillator

The vertical motion of the oscillator is governed by spring stiffness, $k$, damping, $c$, vehicle mass, $m$, and the roughness experienced by the train $r'(t)$,

$$m\ddot{u}(t) + c\dot{u}(t) + ku(t) = c\dot{r}'(t) + kr'(t). \tag{4.5}$$

Here $r'(t)$ can be found by interpolating $r(x)$ the actual roughness profile (in the spatial domain) according to the train's position,

$$r'(t) = r(x)|_{x(t)}. \tag{4.6}$$

In the frequency domain, Eq. (4.5) can be written as,

$$-m\omega^2\hat{u}(\omega) + ci\omega\hat{u}(\omega) + k\hat{u}(\omega) = ci\omega\hat{r'}(\omega) + kr'(\omega), \tag{4.7}$$

which can be rearranged to define the transfer function, $\hat{h}$,

$$\hat{h}(\omega) = \frac{\hat{\ddot{u}}(\omega)}{\hat{r}'(\omega)} = (i\omega)^2\frac{i\omega c + k}{-\omega^2 m + i\omega c + k}, \tag{4.8}$$

where $\omega$ is frequency and $i = \sqrt{-1}$. To find a discrete representation of $\hat{h}$, we evaluate Eq. (4.8)

at discrete frequencies, $\omega[p]$, producing discrete values of the transfer function, $\hat{h}[p]$. Returning to Eq. (4.3), we can now write the physical constraints we will enforce,

$$\begin{aligned} \min_{h,r} \quad & ||\hat{\ddot{u}} - \hat{H}\hat{r}||_2, \\ \text{subject to} \quad & ||r||_0 \leq n \\ & \hat{h}[p] = (i\omega[p])^2 \frac{i\omega[p]c + k}{-\omega[p]^2 m + i\omega[p]c + k}, \end{aligned} \tag{4.9}$$

where $n$ is the number of sparse bumps and $||\cdot||_0$ is the $\ell_0$ norm.

In minimizing Eq. (4.9), we find a sparse version of the track profile due to the $\ell_0$ constraint which limits the roughness profile, $r$, to $n$ non-zero values. In addition, the properties of the identified damped oscillator approximate those of the train's main suspension.

Directly solving this minimization problem would be computationally expensive; instead of simultaneously solving for the optimal transfer function and roughness, we solved for each iteratively. We used an Expectation Maximization approach [16] to first solve for the transfer function then the roughness, repeating until convergence.

As the first step, we create a dictionary of possible transfer functions. Each transfer function is a $p \times 1$ vector, found by selecting values for stiffness, damping and mass then solving the equation from the second constraint shown in Eq. (4.9). Here, the discrete frequency values, $\omega[p]$, correspond to the discrete frequencies of $\hat{\ddot{u}}$. We place all vectors into a dictionary matrix $\hat{D}_h$ , then solve

$$\begin{aligned} \min_{\alpha} \quad & ||\hat{\ddot{u}} - \hat{R}\hat{D}_h\alpha||_2, \\ \text{subject to} \quad & ||\alpha||_0 \leq 1, \end{aligned} \tag{4.10}$$

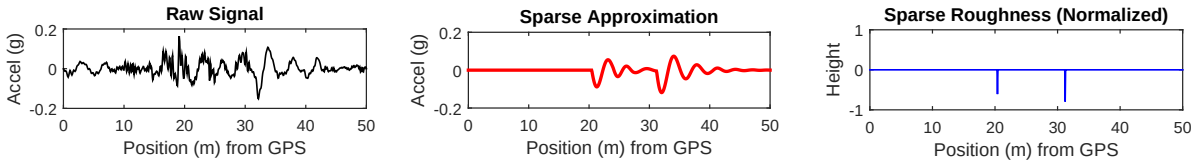
where $\hat{R}$ is a diagonal matrix of the roughness profile in the frequency domain and $\alpha$ is a vector

indicating the selected transfer function from the dictionary. If this is the first iteration, an arbitrary roughness can be used initially; for this roughness, we used $n$ evenly spaced bumps. If this is not the first iteration, the roughness found in the second step is used. Given the $\ell_0$ constraint, directly solving the problem is $NP$-hard [25], so we use orthogonal matching pursuit (OMP [38]), a computationally efficient way to select the optimal $\alpha$. In this case $\alpha$ has one non-zero element; the location of this non-zero element within the vector corresponds to the location of the best transfer function within the dictionary, $\hat{D}_h$, which is the transfer function that minimizes the $\ell_2$ norm. The value of the non-zero element is found using least squares minimization. We then update the transfer function using $\hat{h} = \hat{D}_h \alpha$.

As the second step, we create a dictionary of the possible roughnesses. We want to allow the solution to have a discrete bump at any point along the signal, so our dictionary is a collection of all the possible vectors with a single non-zero value. The result is the identity matrix. Because the calculation is done in the frequency domain, the dictionary is the Fourier transform, $\mathcal{F}$, of the identity matrix, $I$: $\hat{D}_r = \mathcal{F}(I)$, which is known as the discrete Fourier transform matrix [63]. We compute

$$\begin{aligned} \min_{\beta} \quad & ||\hat{\ddot{u}} - \hat{H}\hat{D}_r\beta||_2, \\ \text{subject to} \quad & ||\beta||_0 \leq n, \end{aligned} \tag{4.11}$$

where $\beta$ is the vector indicating the selected roughness bumps from the dictionary and $\hat{H}$ is the diagonalization of the transfer function, $\hat{h}$, found in the first step. We solve this problem using OMP as before [38]. In this method, first the best bump (i.e. the bump that minimizes the $\ell_2$ norm) is selected, then the second best bump is selected, and the magnitude of each adjusted using least squares minimization. This process is repeated up to the $n$ allowed bumps. While the first bump is guaranteed to be optimal, the combination of bumps is not necessarily optimal. Given the results we have achieved, this approach appears adequate for the task, and, as discussed




(a) (b) (c)

Fig. 4.2: Application of the sparse approach to raw signals, showing the found sparse bumps in (c). The sparse approximation shown in (b) can be thought of as the convolution of the train's transfer function with these sparse bumps.

previously, finding the optimal solution is not practical as the problem is $NP$-hard. The trade-off of not necessarily selecting the optimal combination of bumps is that the process is fast computationally.

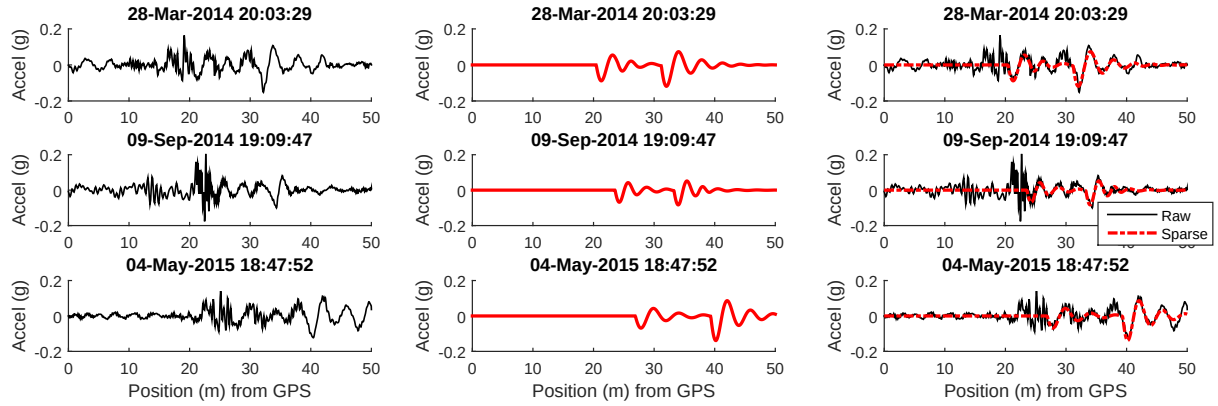
Having determined $\beta$, we update the roughness as $\hat{r} = \hat{D}_r\beta$. We then return to the first step and repeat the process iteratively until convergence. Note that this process is expected to converge because in each step the $\ell_2$ norm error either decreases or stays the same.

## 4.4 Validation on Operational Data

We apply our sparse approach to a large field data set collected from the in-service light-rail vehicle described in Chapter 2. In this section, we show that the sparse approach can effectively analyze noisy data and detect changes in rail infrastructure.

### 4.4.1 Application of the Sparse Approach to Operational Data

Data from one pass of the train traversing a 50m section of track is shown at the top of Fig. 4.2. The vibration signal is plotted as a function of the train's position according to GPS. The signal is approximated using a sparse roughness profile with two bumps (middle) leading to the approximation at bottom. Three signals, approximated in a similar fashion, are shown in Fig. 4.3. Each signal has the same distinctive pattern but the signals are not well aligned to one another due to GPS error. Again, we apply the sparse approach to each signal, limiting the solution to




(a) (b) (c)

Fig. 4.3: Raw signal and sparse approximations for three passes of the train over a 50m section of track. (a) Raw signal from an accelerometer inside the cabin. (b) Sparse approximation of the signal. (c) Raw signal and sparse approximation overlaid.

two bumps, then iterating to convergence. Here two bumps are chosen based on the empirical observation that the train over this length of track experiences large amplitude excitation twice; if $n = 1$, one or the other of the two bumps is selected; if $n > 2$, the solution is less consistent with the additional bump modeling different parts of the signal with each pass. The sparse approximations of the signals are shown in Fig. 4.3b. Note the similarity between the sparse signals; this is because the train travels over the same bumps with each pass, so it has a similar response. The sparse approximation effectively denoises the signal to find only the response to these most significant bumps.

The sparse solution overlaid on the raw signal (Figure 4.3c) lends insight into which part of the signal the sparse solution approximates most closely. Note that the sparse solution does not approximate the raw signal when the train is first excited, but rather has a sort of lag and approximates the latter part of the response to this excitation, which more closely resembles free vibration. When the train is first excited by a rough patch of track, the signal is noisy and depends on the shape of the roughness. Afterwards, the train's response more closely follows the response of a damped oscillator, so the signal depends on the parameters of the train. As the sparse solution tries to decompose the signal into track roughness and the train's response, it approximates the

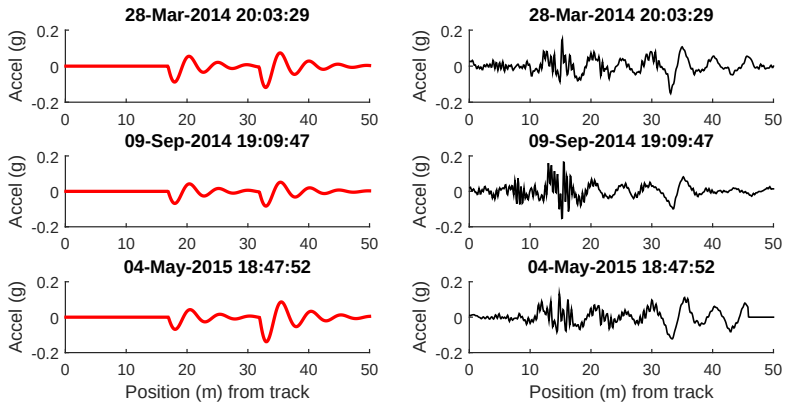


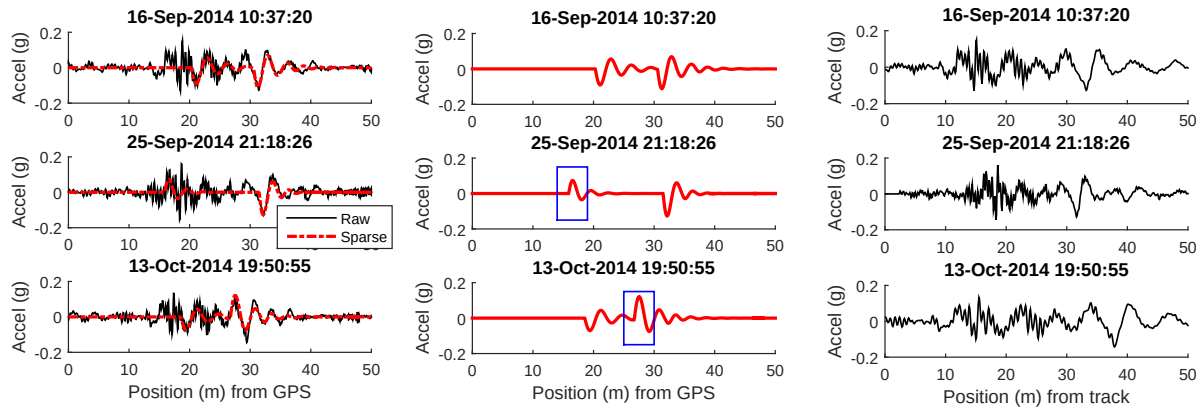

Fig. 4.4: Aligned sparse and raw signal. Using the discrete bumps found through the sparse approach, (a) the sparse approximation and (b) the raw signal can be aligned.

sections when the train's vertical movement most closely resembles free-vibration.

In essence, our sparse approach uses the latter part of a vibration event to characterize the train's properties, where the train is the system of interest. Although to our knowledge this is the first time this concept has been used in vehicle-based monitoring, the concept has parallels in other domains. In seismology, the last part of a ground-motion signal, known as a coda wave, can be used to characterize geological structure, which can be thought of as a system in that context [3].

Returning to the monitoring problem, the location of the identified bumps in the track can be used to align the signals as shown in Fig. 4.4. This is accomplished by piecewise linear interpolation of the data so that the discrete bumps (shown in Fig. 4.2c) occur at the same location for each pass. For these three passes, each looks similar after alignment meaning the sparse approach has selected the same two bumps in the track from the signal (these are the bumps which minimize error). However, it is not guaranteed that the same two bumps will be selected from every signal; if different bumps are selected, alignment based on the bumps does not have a physical meaning. Such is the case with two of the three passes shown in Fig. 4.5.

As discussed earlier, the sparse approach tends to model the train's free vibration response after an excitation event. However, if the excitation itself happens to be similar to the free vibration




(a) (b) (c)

Fig. 4.5: An example of the sparse approach finding inconsistent bumps. (a) The raw signal and the sparse approximation overlaid. The sparse approximation in the first pass (16-Sep) matches the pattern in Figs. 4.3 and 4.4. The second and third passes each follow their own patten. (b) The sparse approximation with boxes to highlight where the sparse approximation differs. In Figs. 4.3 and 4.4 the sparse damped oscillations start downward; the two boxed oscillations start upward. (c) Alignment of the raw signal using the sparse approximation. Because the selected bumps are inconsistent, alignment due to track features is no better than the original GPS alignment.

response (i.e. the excitation appears to be the first oscillation in a series of damped oscillations), then the selected sparse bumps are not consistent with the bumps found in similar signals. In Fig. 4.5, the sparse approximation of the 16-Sep signal matches the sparse approximation in Figures 4.3 and 4.4. For the 25-Sep signal, the first of the two sparse bumps fits the excitation rather than just the free vibration component. A similar phenomenon happens for the second bump in the 13-Oct signal. In Figures 4.3 and 4.4, the excitation is upward (positive), but the free-vibration response begins as the train accelerates downward (negative). The anomalous sparse vibrations in Fig. 4.5 appear upwards, as is highlighted by the blue boxes in Fig. 4.5b. The sparse approximation could differ either because of an issue with the approximation itself, for example, approximating the noise in the signal, or because of a variation in the train response, for example, the initial condition of the train before hitting a bump could affect its response. In either case, if inconsistent bumps are chosen by the sparse method, using these bumps for alignment leads to misaligned signals in Fig. 4.5c.

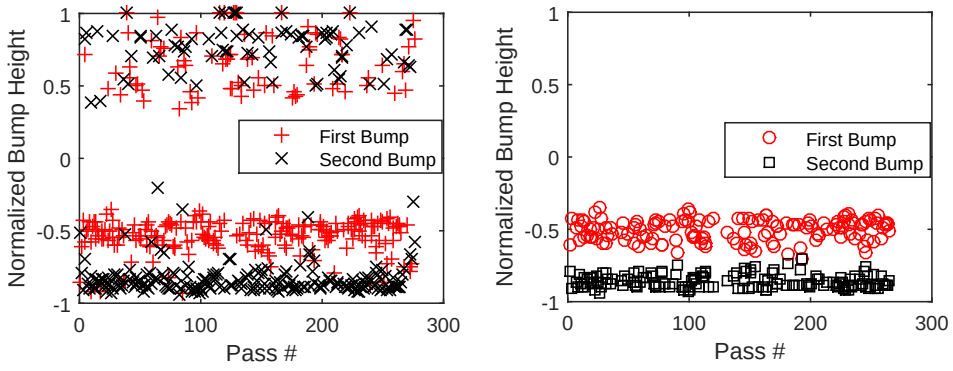
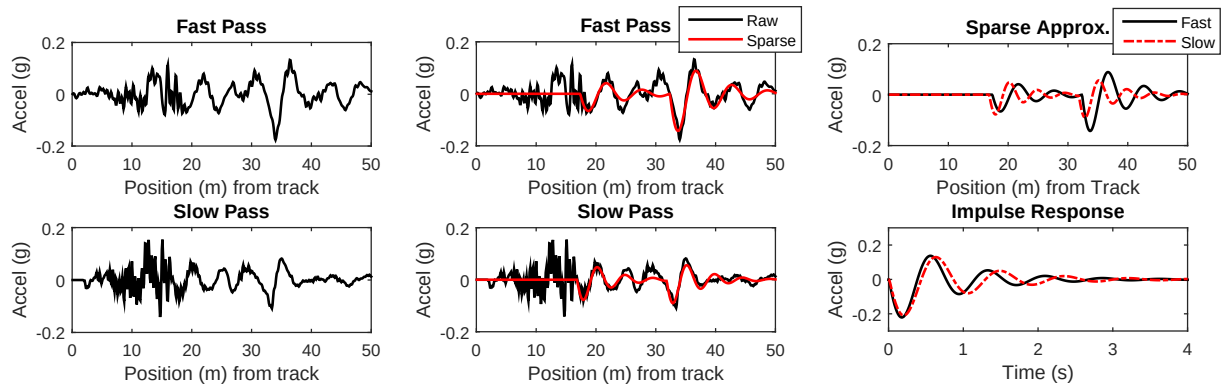



Fig. 4.6: Cleaning the data using the found bump heights. (a) The bump heights for the 267 passes over the 50m section of track. (b) Selected passes in which the first bump is negative and the second bump is more negative than the first.

Fortunately, this inconsistency is easy to detect, and we can use the size and direction of the found sparse bumps. Fig. 4.6a shows the bump heights for all 267 passes recorded between February 2014 and June 2015. Note that while solving the inverse problem, the transfer function could be very small and the bumps very large, or *vice versa*, so when comparing values between solutions, we normalize the bump heights by setting the length of the found roughness vector to 1 ($\sum r^2 = 1$). While the majority of signals have negative values for both of the bump heights, there are exceptions which follow the patterns explained in Fig. 4.5. Using a simple threshold, we can select only the passes with a negative first bump, and a second bump that is more negative than the first. Doing so leaves us with the 145 passes shown in Fig. 4.6b. Note here, the track state appears to be consistent over time; this is discussed in detail at the end of this section.

One benefit of using an explicit model, such as the sparse approach, is that the train's variable speed is handled automatically by the way in which the problem is formulated. The success of the sparse approach in approximating signals from passes of different speeds is shown in Fig. 4.7. In Fig. 4.7a, two passes are shown, one where the train's average speed is 7.8 m/s and another where the average speed is 5.4 m/s. Note that the slow pass appears to have more high-frequency content. This is because the data is sampled at a constant rate in time (1.6kHz), so slower passes have more samples per meter, and thus the appearance of more high frequency



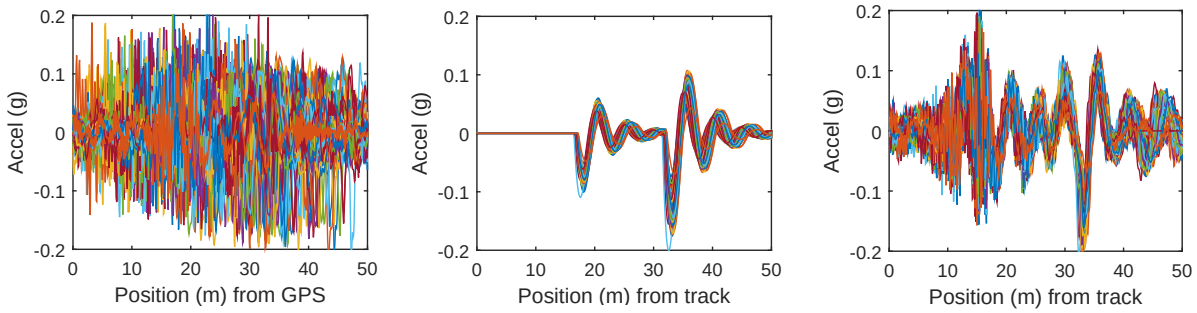

(a) (b) (c)

Fig. 4.7: The sparse approach applied to signals of different speeds, one where the train is moving at 7.8m/s ("Fast Pass") and one at 5.4 m/s ("Slow Pass"). (a) The raw signals of each pass. (b) The raw signals with the sparse approximation overlaid. (c)(top) The sparse approximation for the two speeds overlaid. (c)(bottom) The impulse response of the system used for the sparse approximation. This Fig. shows that the sparse approximation correctly handles variable speed. When two sparse approximations for different speeds are shown versus position, the oscillations are different because they conform to the observed data. However the impulse response for the two approximation are similar in the time domain, showing that the system of the train does not change much with speed (as expected).

noise when plotted spatially. Fig. 4.7b shows the sparse approach applied to the two passes; in both cases, the sparse approximations model similar components of the raw signals.

Fig. 4.7c showcases the flexibility of the sparse approach in modeling different speeds. When the train hits a bump in the tracks, the duration of each oscillation is relatively constant and depends largely on the natural frequency and damping of the suspension. This physical intuition is clearly seen in the bottom part of Fig. 4.7c where the impulse responses of the systems found in analyzing the data are similar. However, in the spatial domain, the track distance covered during each oscillation depends on the train's speed, so the difference seen in the top panel of Fig. 4.7c is expected.

The success of the sparse method in making sense of noisy data is evident in Fig. 4.8. The 145 passes which conform to the pattern of interest are shown in Fig. 4.8a according to their recorded GPS position. As can be seen, this raw signal looks unintelligible. The sparse approximation is then found for each, and aligned according to the position of the bumps as shown in Fig. 4.8b.




(a) (b) (c)

Fig. 4.8: The sparse approach's ability to make sense of noisy data. (a) The raw signal for the 145 passes which conform to the patten in Fig. 4.6b. (b) The sparse approximation for these 145 passes, aligned according to the bumps. (c) The original data from (a) now aligned according to the sparse bumps.

This sparse approximation reveals underlying meaning in the noisy signals: the train gets excited by two bumps along the track as it travels. Finally, we can use the information from the bumps to align the original data, which is shown in Fig. 4.8c.

In this particular section of track, the pattern we observe (two negative bumps) appears to occur throughout the monitoring period. This can be seen in Fig. 4.6, where the pass indices are arranged chronologically, and the 145 selected indices are evenly distributed throughout the 267 total indices. We can confirm that for this particular 50m section of track, no maintenance work was performed or requested by track inspectors according to the maintenance logs. Thus we assume that the consistency of the pattern over the 16-month period indicates that the tracks did not change significantly. A section of track that did change will be the focus of the next section, which can be detected through inconsistent patterns in bump height (as will be shown in Fig. 4.12).

### 4.4.2 Detecting Track Change with the Sparse Approach

In this section, we attempt to detect a change in track geometry due to tamping using the sparse approach. In this example, as mentioned previously, we are detecting maintenance work because it can be verified against work logs. However, just as our approach can detect the change from


Fig. 4.9: The tracks and bridge of interest. The lower photo shows the entire span of the bridge, while the photos above are aerial shots showing the transition between the concrete deck and the ballasted track.

a state of mis-repair to a state of good-repair, it could equally detect degradation from a state of good-repair.

In September of 2014, maintenance workers tamped the ballasted tracks on both sides of a bridge structure shown in Fig. 4.9. The vibration data from this change was analyzed previously in Chapter 3 using the energy in the vibration signal [36], and representative results are reproduced in Fig. 4.10. Unlike the sparse approach, which models the track state explicitly, the signal-energy approach models the track implicitly. Although there is less energy in the signal after the tracks are tamped (i.e. the train ride is smoother), it is unclear exactly what happened. Furthermore, because the track itself is not modeled, it is more challenging to align data given the position uncertainty from the GPS signal.

We next apply the sparse approach to the vibration data collected in the instrumented car from February 2014 to June 2015 around the bridge of interest. Four example passes of the raw signal are shown in Fig. 4.11a. The tamping activities occurred in late July 2014; two passes prior to tamping and two passes after tamping are shown. Fig. 4.11b shows the sparse approximation, which clearly exhibits two changes in the signal. While in the first two passes, the oscillation due to the first bump is positive, in the second two passes, it is negative. The second change can be seen towards the end of the signal: the first two passes have an additional excitation after the

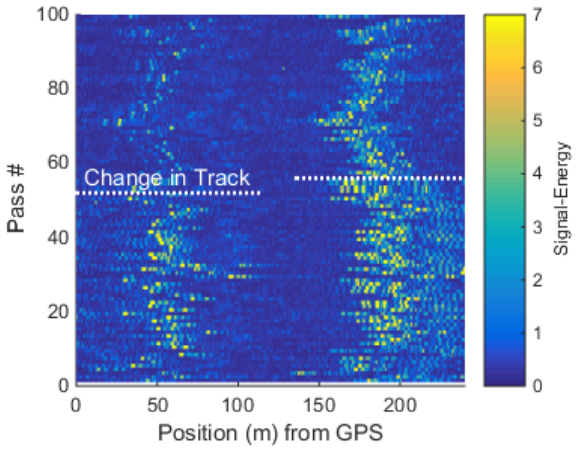
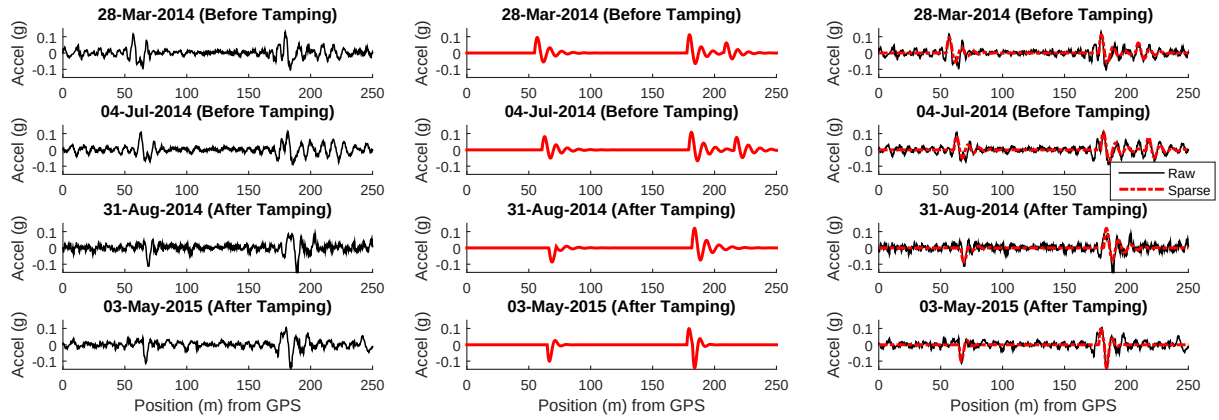



Fig. 4.10: The change due to tamping shown using a signal-energy feature [36]. Each horizontal line corresponds to one pass of the train over this section of track, with the color indicating the signal-energy at that location. The high energy point around 60m is the track joint where the bridge starts and the high point at 180m is the track joint where the bridge ends. After pass 50, the ballasted track to the left of the bridge is tamped. After pass 55, the ballasted track to the right of the bridge is tamped. Note the position here is from the GPS signal, so consecutive passes are not well aligned.

second major bump, while the second two do not.

We can use the height of sparse bumps as a measure of the track height. As shown in Fig. 4.12, if we look only at the height of the first bump, it changes dramatically after the tamping has occurred. If the goal was to classify the state of the track, (i.e. pre-tamping versus post-tamping) this could be done simply using the sign ($\pm$) of the first bump. In this case, of the first 59 passes, 57 would be correctly identified as pre-tamping (96% accuracy) and of the 108 remaining passes, 98 would be identified as post-tamping (90% accuracy). Overall, this would lead to an accuracy of 92% using just the height of one bump and a hard threshold; this is higher than the accuracy found earlier for classifying the state of the track using signal-energy. In that case (detailed in Chapter 3) we found 90% classification accuracy using data from a 500m section of track and a more sophisticated support vector machine classifier (SVM).

One remarkable aspect of the sparse approach is its ability to overcome the position uncertainty from GPS error. With the implicit approach, a high-dimensional feature was used from each pass: the signal energy at each point along the track. Longer feature vectors are required be-




(a) (b) (c)

Fig. 4.11: The sparse approach applied to the bridge data. (a) The raw signal for two passes before tamping and two passes after tamping. Notice the train's response to the track joint at the beginning of the bridge around 60m and at the end of the bridge around 180m as was also seen in Fig. 4.10. (b) The sparse approximations for these passes. Notice that the first oscillation is positive for the first two passes and negative for the second two. (c) Comparison of the raw signal and the sparse approximation.

cause the position is not known exactly. Using the bump height found with the sparse approach, in this case a one-dimensional feature, consistently models the same point in the signal despite the GPS error. Because of the simplicity of the feature, a threshold can be used successfully for the classification.

The real power of the sparse approach, however, is not simply its ability to achieve high classification accuracy; it is its ability to provide insight into how the track has changed. The change from positive oscillations to negative oscillations as the train enters the bridge means that the relative heights of the track around the joint have changed. Before the tamping, the train would accelerate upward as it crossed the joint, meaning the track on the bridge side of the joint was likely higher than the track on the ballasted side. After the tamping, the train would accelerate downwards meaning the track on the ballasted side is likely higher now than the track on the bridge side. This type of information could be useful to inspectors. If this data were observed outside of a period of maintenance, it could signify, for example, that the bridge had settled. Such is the benefit of building an explicit model: the bumps, although sparse, directly

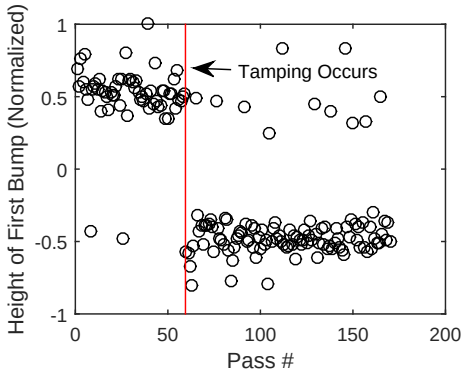



Fig. 4.12: The height of the first bump for each pass over this section of track. Tamping occurs after the 59th pass. The mean of the bump heights changes from 0.5 to -0.5. This can be used as a feature to detect the change.

relate to the shape of the track that most excites the train.

## 4.5 Discussion

In this chapter we have proposed a novel sparse approach for analyzing vibration data collected from an operational train. The approach uses an iterative method to find an approximation for the track roughness and for the properties of the train. By enforcing sparsity in modeling the train system, we solve for the parameters of the train's main suspension. By enforcing sparsity in the track profile, we learn where in the tracks the train is most excited.

Applying this approach to operational train data, we show that the sparse approach can find consistent patterns in the train's response, effectively denoising the data. The identified sparse track roughness is invariant to train speed, and the location of the bumps can be used to determine the train's location. In this regard, the sparse approach helps to overcome error in the GPS signal. Furthermore, the magnitude and location of the found bumps are strong indicators of the state of the tracks, which we have shown both in cases where the track did not change, and where there was a change. In the example shown in this chapter, the change detected is due to maintenance work so the findings can be validated against work logs, although the methods could equally be applied to detect damage.

## 4.6 Future Work

Throughout this chapter, the number of "bumps" was chosen empirically for each section of track. A formal method for selecting the appropriate number of "bumps" would make it possible to apply this method to all track sections within a network. In an effort to present such a formal method, we studied how the number of allowed "bumps" affects the reconstruction error of the sparse approximation. While minimizing the reconstruction error has been used for selecting the optimal level of sparsity in other applications, we did not find it was suitable here. The main criterion in selecting the number of bumps empirically was to look at several passes of the train over a section of track, and to choose the number of significant track bumps that tend to occur with each pass. By so doing, we were able to achieve a consistent sparse approximation between passes. As the number of bumps increases, the reconstruction error decreases, but the consistency of the solution decreases as well because fewer constraints on the sparse solution means more freedom in how it approximates the signal. Future work could consider how to select the number of bumps which yields the most consistent sparse approximation while simultaneously minimizing the reconstruction error.

In addition, the train system was constrained to be a single degree of freedom oscillator, or equivalently, by considering a single mode of vibration of the vehicle. This could be relaxed to include more complex characterizations of the train. The challenge here is two-fold: first, while multi-degree of freedom train systems could be used, the different degrees of freedom are coupled, so solving for each could be computationally expensive. Second, and perhaps more challenging, we were unable to demonstrate that the found natural frequency of the train on a particular pass (the impulse response found through the sparse approximation) corresponded to changes in the train itself. Prior to using more complex train models, the validity of the current model would have to be further confirmed. We expected that by characterizing the train as an oscillator, we would see lower-frequency oscillations when the train is heavy at rush-hour, for example, or when the suspension is particularly cold. However, we were never able to attribute

the changes in the found fundamental frequency of the train to known changes in the train itself. This was partly due to a lack of information about the train for each pass, and perhaps changes due to passenger loading or temperature levels are not significant. However, given the limited information we had, these were the only changes we thought to test. The found fundamental frequency of the train appeared to be almost random when plotted against either temperature of time of day (where time-of-day indicates rush hour). Further study could confirm whether this variability is due to inaccuracy in our sparse method or insufficient data on the condition of the train.

Finally, this chapter does not address the challenge of combining data collected from multiple vehicles, although by solving an inverse problem, this task is straightforward. The sparse approach effectively decomposes the signal into track components and vehicle components. The track components extracted from different trains should be equivalent. Combining data between multiple trains when using an implicit model is more challenging. This is topic is studied in the next chapter.

# Chapter 5

# Data Fusion

This chapter presents a data fusion approach for enabling data-driven rail-infrastructure monitoring from multiple in-service trains using an implicit model of the tracks. While in Chapter 3, the features required to build an implicit model are studied, no formal approach was presented for combining data between multiple sensors, passes, or trains. In this chapter, we present a feature agnostic approach to fuse together this information to build a more reliable track model. We use a two-step approach that first minimizes GPS error through data alignment, then fuses the data with a novel adaptive Kalman filter. We show the efficacy of this approach both through simulations and by using one year of data from the two trains we instrumented. As will be shown, the proposed data fusion approach allows for more continuous and more robust data-driven monitoring than by analyzing data from any one train alone.

## 5.1 Introduction

Most researchers investigating train-based track monitoring have studied how to increase inspection reliability by developing new features to build more reliable implicit track models. Rather than propose a new feature, in this chapter, we introduce a novel data fusion approach that can take as input the features currently used in the monitoring community. As will be shown, our

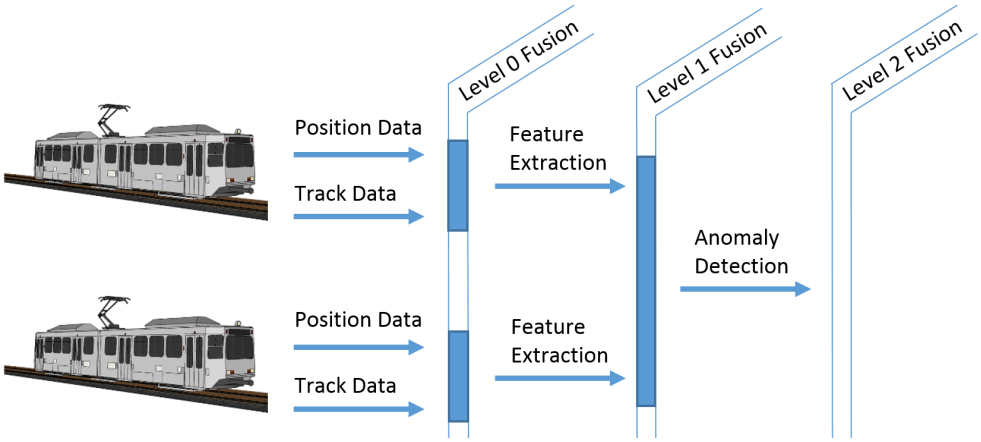



Fig. 5.1: Proposed processing pipeline. The proposed data fusion approach in this paper is a Level 1 Fusion method. Level 0 Fusion is the combination raw data, Level 1, features, and, Level 2, decisions. Although Level 2 Fusion is not required for the proposed pipeline, it is used in other studies [19], and is thus included for completeness.

fusion approach not only enables more continuous monitoring by leveraging data from multiple trains, but also, by combining multiple sensors, increasing the overall reliability of inspection from in-service trains.

Before describing our proposed method, we briefly discuss prior work on data fusion in the vehicle-based infrastructure monitoring space. To do so in a structured manner, we introduce the most prevalent data fusion model, the JDL Model (Joint Directors of Laboratories of the US Department of Defense), which has been developed and refined for defense applications since the mid-1980s [18, 28, 57, 59, 70]. In this model, data fusion is categorized into distinct "levels" within a data-processing pipeline. Level 0 Fusion is "Sub-object Data Association and Estimation," Level 1 Fusion is "Object Refinement," and Level 2 Fusion is "Situation Refinement." In our proposed data processing pipeline, shown in Fig. 5.1, Level 0 Fusion means combining raw data from different sensors, for example, fusing position data with data about the track condition. Level 1 Fusion means combining object level data, in this case, features extracted from multiple passes over the same track, where the track section is the "object" of interest. Level 2 Fusion is combining situation level information, in this case, the detection outcome from anomaly detection. In our proposed pipeline, all the data are fused at Level 1, so no Level 2 Fusion is

required.

We can now reexamine studies on train-based track monitoring, discussed earlier in this thesis, in light of of these data fusion categories. One of the earliest studies in train-based track monitoring was conducted by Bocciolone et al. [7]. They developed a system to detect track corrugations in the Milan Metro by looking at the wavelet transform of data acquired from accelerometers mounted on the axle of a passing train. In their study, they performed Level 0 Fusion by combining accelerometer and position data; they then analyzed the accelerometer data in the wavelength domain which is position dependent. Their study does not go beyond feature extraction; they do not combine data from different passes over the track, different accelerometers, or different trains.

The same year, Weston et al. [69] published a study on monitoring track alignment using sensors placed on a train in the Tyne and Wear Metro. They propose a technique to extract the standard deviation of the track profile, which is a common parameter of track geometry used in traditional track geometry car inspection. In terms of data fusion, Weston et al. go beyond the work by Bocciolone et al. by presenting results from multiple passes over the same track separated by several months. They note regions in which changes in the track appear to have occurred, but they do not present formal Level 1 Fusion approaches or anomaly detection techniques.

While higher-level data fusion approaches have not been studied for train-based infrastructure monitoring, they have been employed in related fields. Eriksson et al. [19] propose a method for pothole detection that uses vibration data collected from cell-phones in taxi-cabs. In their study, potholes are defined as events which exceed a predefined threshold; several detection events from individual vehicles must occur in the same vicinity before a pothole is detected by the overall system. Essentially, data from each individual pass is analyzed independently all the way through anomaly detection; if a pothole is detected, this detection serves as a "vote" that a pothole has occurred at that location. Voting represents one type of Level 2 Fusion, as it occurs after anomaly detection.

This voting technique works well in cases where a specific type of damage is of interest. For example, Molodova et al. [43] study rail-squats and define this type of damage as any point in the tracks where a particular feature exceeds a threshold set *a priori*. However, in many cases it is more desirable not to specify a threshold or the type of damage of interest *a priori*. In these cases, a data-driven approach can be used to build a baseline estimate of the typical track features, then detect changes when deviations from this baseline occur. This allows a wider variety of damage types to be detected and ensures that sections of track with complex geometry do not unnecessarily trigger false alarms.

However, building a baseline model from multiple passes and multiple vehicles requires a Level 1 Fusion approach; neither the Level 2 voting approach, nor the Level 0 combination of raw data proposed previously, fill this need. There are three challenges in combining data from in-service trains at Level 1. First, the trains do not necessarily pass over the tracks at regular intervals; thus the pass data is asynchronous. Second, position data, typically from GPS, has variable levels of accuracy, and data from different passes are often not well aligned spatially. Third, individual track sensors or entire train monitoring systems can malfunction or exhibit high noise levels. However, both the sensors and the systems can be reset, so the reliability of the data changes with each pass.

To address these challenges, we propose a novel fusion approach. The method is based on the Kalman filter, which can handle asynchronous data. In addition, we propose an alignment technique as a pre-processing step to mitigate GPS error. Finally, in order to combine data from as many senors as possible while ensuring that malfunctioning sensors do not degrade the overall output, we estimate the sensor reliability, then weight each sensor accordingly.

We validate our approach on both simulated data and data collected from two in-service trains in Pittsburgh, PA. For the simulation study, the ground truth is known so the performance of the data fusion approach can be readily evaluated. For the operational data, the precise state of the tracks is not known at any moment; in this case, we investigate whether known changes in the tracks can be more readily detected after fusing the data versus analyzing the data from

individual sensors independently.

## 5.2 Algorithm

The goal of our data fusion approach is to estimate the state of the tracks using data from multiple trains. We perform Level 1 Fusion [28, 59], combining features extracted from the raw signal of multiple sensors. In this application, the estimated track state is a feature representation of the state of the tracks; if the fusion is successful, the resultant feature representation is more accurate and consistent than if it were extracted from any individual sensor alone. As our proposed technique is based on the Kalman filter, we first provide a brief overview of the filter and its benefits.

The Kalman filter provides an estimate of the current state of a system by combining current and past data collected from that system. It is one of the most popular data fusion techniques because it is computationally efficient, works in on-line applications, and can process data collected asynchronously. For a linear process where the noise has a mean of zero, the estimate is optimal [32, 67].

As an example, let $\mathbf{x}_k$ be a vector describing the state of a linear system at time-step $k$, which relates to the previous state according to

$$\mathbf{x}_k = a\mathbf{x}_{k-1}. \tag{5.1}$$

Our goal is to estimate the state at each time-step from a set of noisy observations acquired from $j$ sensors, $\mathbf{z}_{k,j}$ (also a vector),

$$\mathbf{z}_{k,j} = \mathbf{x}_k + \epsilon_j, \tag{5.2}$$

where $\epsilon_j$ is some zero-mean noise specific to the sensor.

The Kalman filter can be used to find an optimal estimate of the system from these observations, assuming $a$, a parameter of the linear process, is known. The steps for one iteration of

**Algorithm 1** Kalman filter

This algorithm is repeated for each time-step $k$.

**Predict**

1. Estimate the state where $a$ is some scalar constant describing the linear process of interest.

$$\hat{\mathbf{x}}_k = a\hat{\mathbf{x}}_{k-1} \tag{5.3}$$

2. Estimate the error variance of the prediction, $p_k$.

$$p_k = ap_{k-1} \tag{5.4}$$

**Update**

3. Update the Kalman gain, $k_{k,j}$, for each of the $j$ sensors. Here $r_j$ is the variance for each of the $j$ sensors; in the basic Kalman filter, the sensor variance is assumed to be known *a priori*.

$$k_{k,j} = \frac{p_k}{p_k + r_j} \tag{5.5}$$

4. Update the state estimate, $\hat{\mathbf{x}}_k$, based on the current measurement data, $\mathbf{z}_{k,j}$, from the $j$ sensors.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \sum_j k_{k,j}(\mathbf{z}_{k,j} - \hat{\mathbf{x}}_{k-1}) \tag{5.6}$$

5. Update the prediction variance, $p_k$.

$$p_k = \left(1 - \sum_j k_{k,j}\right) p_k \tag{5.7}$$

the most basic Kalman filter are shown in Algorithm 1; these steps are repeated each time new information is collected. Throughout this paper, we will use bold letters to denote vectors, and hat symbols $\hat{\cdot}$ to denote values that are only estimates of the true values. First the estimate of the state of the system, $\hat{\mathbf{x}}_k$, is predicted from the predicted state at the previous time-step, $\hat{\mathbf{x}}_{k-1}$, multiplied by the known constant of the linear process, $a$, as shown in Eq. (5.3). The prediction error variance, $p_k$, provides an estimate of how much the prediction from the previous step can be trusted; this too is updated in Eq. (5.4).

While Eqs. (5.3) and (5.4) provide an *a priori* estimate, Eqs. (5.5-5.7) provide an *a posteriori* estimate, and thus are categorized under a new heading, "Update" [67]. The first step under this heading is to estimate the Kalman Gain for each of the $j$ sensors, $k_{k,j}$, that controls the trade-off between trusting the estimate from the last time-step versus trusting the newly observed data. This "trust" level depends on both the prediction error variance, $p_k$, and the sensor variance, $r_j$. Eq. (5.6) then provides the updated estimate, including the newly observed data, $\mathbf{z}_{k,j}$. Finally, the *a posteriori* estimate of the prediction error, $p_k$, is calculated in Eq. (5.7).

For the case of track monitoring, we are interested in estimating the state of the tracks with each newly collected pass of the train over them; thus $k$ is used to refer not to time-steps but to the pass numbers. Although the passes can be thought of as time-steps, we avoid using the word "time" as it could be confused with the time-domain data collected within each pass.

As discussed in section 5.1, there are three basic challenges in fusing data about the state of the tracks: the data is collected asynchronously; the data can be misaligned spatially; and individual sensors can be noisy or malfunctioning.

To address the first challenge, we select a Kalman filter as the basic data fusion technique. The filter can provide a new prediction of the state of the tracks with data collected from each new pass of the train. This is the core of the proposed data fusion approach we present in Algorithm 2.

To address the second challenge, we align observed data from each pass over the tracks so that they fit more closely with the previous estimates of the feature representation for the tracks. The

**Algorithm 2** Proposed Fusion

This algorithm is repeated with each new $k^{\text{th}}$ pass over the tracks of interest.

1. Align Data I - find the optimal offset, $\hat{m}_{k,j}$, between the measured data at the $k^{\text{th}}$ pass from the $j^{\text{th}}$ sensor, $\mathbf{z}_{k,j}$, and the estimated state at the previous pass, $\hat{\mathbf{x}}_{k-1}$, by finding the value of $\hat{m}_{k,j}$ that maximizes the cross correlation,

$$\underset{\hat{m}_{k,j}}{\text{argmax}} \left( \sum_{n=1}^{N} \hat{\mathbf{x}}_{k-1}[n] \mathbf{z}_{k,j}[n + \hat{m}_{k,j}] \right). \tag{5.8}$$

2. Align Data II - determine the estimate of the correctly aligned data, $\hat{\mathbf{z}}_{k,j}$,

$$\hat{\mathbf{z}}_{k,j}[n] = \mathbf{z}_{k,j}[n + \hat{m}_{k,j}]. \tag{5.9}$$

For values of $n$ where $\mathbf{z}_{k,j}[n + \hat{m}_{k,j}]$ is undefined (i.e. $n + \hat{m}_{k,j} > N$), set $\hat{\mathbf{z}}_{k,j}[n] = 0$.

3. Kalman filter I - estimate the sensor variance, $r_{k,j}$,

$$r_{k,j} = (\hat{\mathbf{z}}_{k,j} - \hat{\mathbf{x}}_{k-1})^T (\hat{\mathbf{z}}_{k,j} - \hat{\mathbf{x}}_{k-1}). \tag{5.10}$$

4. Kalman filter II - calculate the Kalman Gain, $k_{k,j}$. Here $p_{k-1}$ is the prediction error from the previous step,

$$k_{k,j} = \frac{p_{k-1}}{p_{k-1} + r_{k,j}}. \tag{5.11}$$

5. Kalman filter III - estimate the current state,

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1} + \sum_j k_{k,j} (\hat{\mathbf{z}}_{k,j} - \hat{\mathbf{x}}_{k-1}). \tag{5.12}$$

6. Kalman filter IV - update the prediction error,

$$p_k = \left( 1 - \sum_j k_{k,j} \right) p_{k-1}. \tag{5.13}$$

If $p_k < p_{min}$, $p_k = p_{min}$.

misalignment occurs due to position error from the GPS such as multipath reception, a particular problem when the GPS receiver does not have a clear view of the sky. This error tends to be relatively constant for a given pass over a short section of track, because the multipath error is constant for a particular configuration of satellites relative to the train [33]. To align the observed data with the estimate of the state of the tracks, we determine the offset using cross correlation in Eq. (5.8), then provide an estimate of the properly aligned observed data, $\hat{\mathbf{z}}_{k,j}$, in Eq. (5.9).

To address the third challenge, we estimate sensor variance for each sensor and each pass over a given section of track. Typically the error level of a sensor is constant for each pass. For example, if the sensor is malfunctioning, the collected signal is not useful over the entire pass. Conversely, if the signal appears similar to previous readings over the first half of a section of track, the second half of the signal also tends to match historical data. Thus, we calculate the sensor variance, $r_{k,j}$, for each pass according to the variance between the observed signal over the entire pass and the estimate of the track state over the entire length of the track section, as shown in Eq. (5.10). Because we calculate the variance with each pass, we make the Kalman filter "adaptive." The majority of adaptive Kalman filters proposed in the literature also present novel methods for estimating sensor variance [41, 61], so the present innovation follows a trend in the literature.

Eqs. (5.11-5.13) in our proposed algorithm follow closely from Eqs. (5.5-5.7) in the standard Kalman filter. Eqs. (5.3) and (5.4) from the standard Kalman filter have not been reproduced because in our application, the state of the tracks is predicted to remain the same between individual passes, so $a = 1$. Thus, we simply use $\hat{\mathbf{x}}_{k-1}$ in Eqs. (5.11-5.13) rather than $\hat{\mathbf{x}}_k$ as is used in Eqs. (5.5-5.7).

One concern in using a Kalman filter is that it is designed to estimate a system in steady-state; this too must be modified for our application. While the state of the tracks is estimated to remain constant, it could change at any time. Because of this, the prediction error should never be zero, as this would mean that the Kalman filter would trust only its prediction and ignore the newly observed data. In the standard Kalman filter, the prediction error $p_k$ approaches zero

asymptotically; to avoid this, we set a minimum level of $p_k$ which we refer to as $p_{min}$ as the last step in Algorithm 2. The value of $p_{min}$ is the only parameter which must be set in our proposed data fusion method. As will be discussed in the next section, determining the optimal value for $p_{min}$ is an important component of implementing our data-fusion approach. If $p_{min}$ is too small, the model updates slowly after a change in the tracks; if $p_{min}$ is too large, the filter is sensitive to noise in newly collected data.

## 5.3 Data Fusion Applied to Simulated Data

To test our proposed data fusion approach, we first apply it to simulated data. As mentioned previously, our data-driven approach could be applied to fuse a wide variety of features; one of the benefits of testing the approach in simulation is that more generic data can be generated and tested to show the generality of the approach. In addition, in simulation, the ground truth is known, so the performance of the approach can be readily quantified. This is important, not only for judging the effectiveness of the approach, but also for parameter selection.

### 5.3.1 Data Generation

We simulate some feature vector over the length of the track in two different states as shown on the left of Fig. 5.2; this is a feature representation of the ground truth state, $\mathbf{x}_k$. The change between State 1 and State 2 symbolizes some sort of track deterioration or maintenance that causes the tracks themselves to change.

The right side of Fig. 5.2 shows two examples of the observed features from the train. These are simulated by adding a random offset $m_{k,j}$ to the true feature state, $\mathbf{x}_k$, along with some noise $\epsilon_j$ scaled by a value $c_k$ which is constant for each pass,

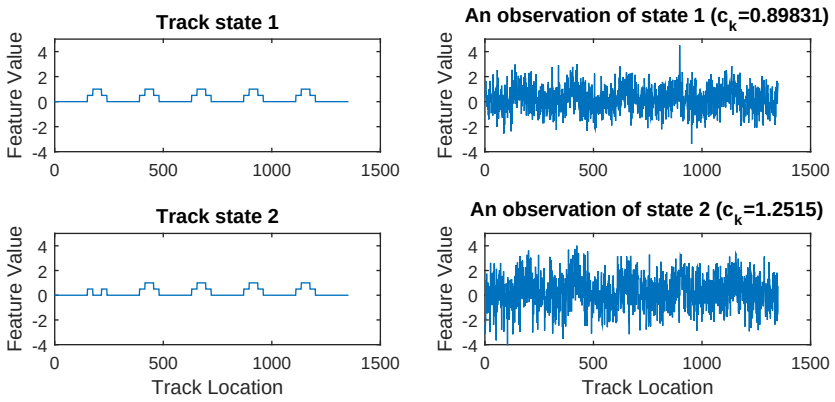
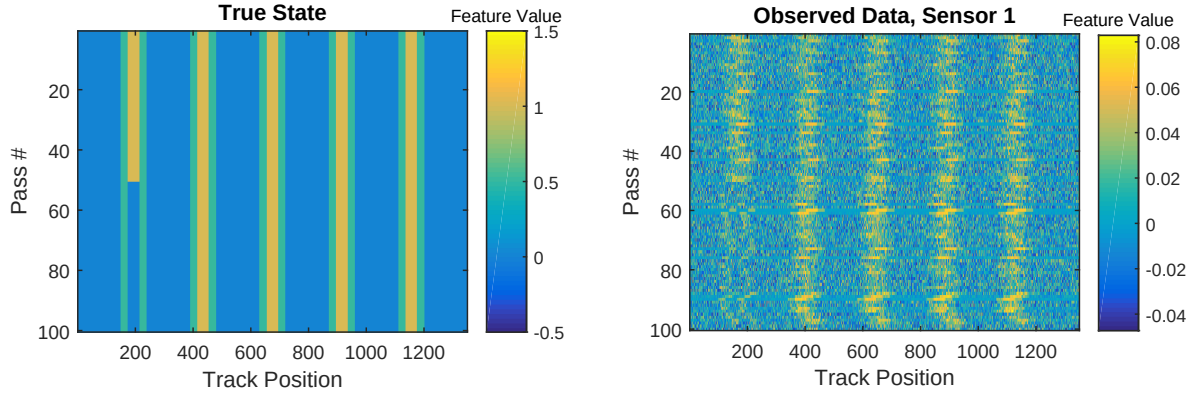



Fig. 5.2: True feature representation of track state and observed feature representation of track state. At left, the track state is shown for two states, which might represent before and after a change at location 250. At right, an example of the observed data for one pass over the tracks in each state.




(a) (b)

Fig. 5.3: The true state of the tracks and the observed state of the tracks, in terms of extracted features. (a) The track states over 100 passes. Each horizontal line shows one of the two track states from Fig. 5.2, with the first 50 pass showing State 1, and the second 50 passes showing State 2. (b) The data observed about the state of the system from a passing train. It is normalized such that the length of each vector is one ($\sum_n \mathbf{z}_{k,j}^2[n] = 1$).

$$\mathbf{z}_{k,j}[n] = \mathbf{x}_k[n + m_{k,j}] + c_k \epsilon_j[n],$$

where

$$\begin{aligned} m_{k,j} &\sim \mathcal{U}(0, 50) \\ c_k &\sim \mathcal{U}(0, 2) \\ \epsilon_j[n] &\sim \mathcal{N}(0, 1). \end{aligned} \tag{5.14}$$

Here, $\mathcal{U}(a, b)$ indicates a uniform distribution between $a$ and $b$, while $\mathcal{N}(\mu, \sigma)$ indicates a normal distribution with mean $\mu$ and standard deviation $\sigma$. Note that these random constants represent the challenges mentioned in Section 5.2: $m_{k,j}$ is the offset representing GPS noise, while $c_k$ scales the noise in each pass, which indicates whether the observed data is accurate or noisy. Due to the low signal-to-noise ratio (less than 0.1 for some passes), determining the state of the tracks from a single pass provides an inaccurate estimate. However, in the case of train-based monitoring, particularly if several trains are instrumented, numerous passes can be collected, and the data from each pass fused to achieve a better estimate.

Fig. 5.3 shows 100 simulated passes over the track section of interest, assuming that for the first 50 passes, the track is in State 1, and for the second 50 passes, the track is in State 2. In Fig. 5.3a, the true state, $\mathbf{x}_k$, is shown for $k$ passes where $k = [1, 2, ..., 100]$, while in Fig 5.3b, the observed data for one of the $j$ sensors is shown ($\mathbf{z}_{k,j}$). In this simulation, we assume we have two sensors ($j = [1, 2]$) on a single train, so we assume $c_k$ is the same in both cases. This makes the fusion more challenging, because if one sensor is noisy, the other sensor is also noisy, so the estimate for that pass must rely more heavily on the estimate from the previous passes. Note that the observed data have been normalized such that the length of the vector is equal to 1: this is beneficial because noisy passes tend to have larger amplitudes (both in this simulation and in the data we have collected from our instrumented trains) so normalization helps by reducing the magnitude of these noisy passes.

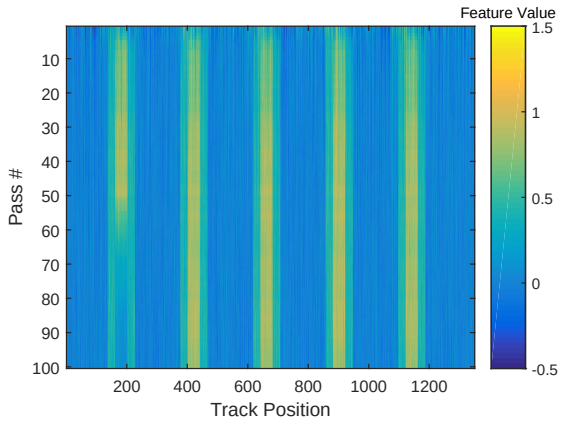
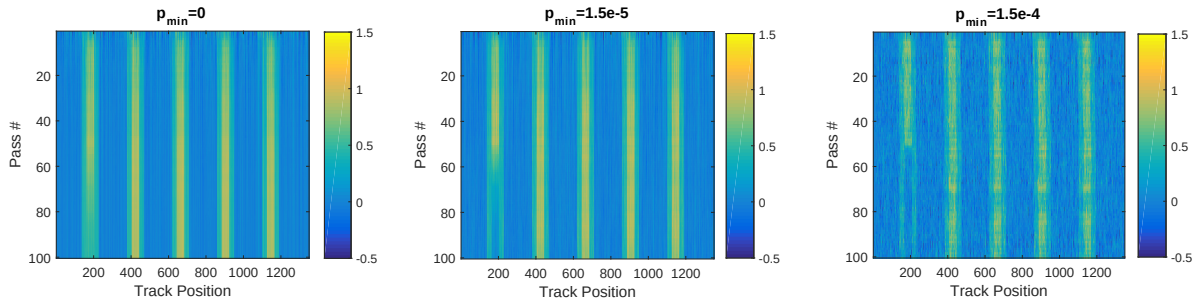



Fig. 5.4: Estimate of the state of the track using the proposed approach. Qualitatively, this estimate appears to be successful as it is close to the true state shown in Fig. 5.3a.

### 5.3.2 Data Fusion

We can now apply the approach presented in Algorithm 2 on the simulated data shown in Fig. 5.3b. The output, the estimate of the state of the tracks, is shown in Fig. 5.4. This figure bears remarkable similarity to the true state of the tracks shown in Fig. 5.3a, meaning the fusion approach is largely successful. Notice that for the first several passes, the estimate is not good, but by around the $20^{\text{th}}$ pass, the filter produces an accurate estimate of the state. Immediately after the change from State 1 to State 2 at pass #50, the filter is slow to adapt to the change. The trade-off between adapting quickly to changes versus making the most accurate estimate of the track if it is in steady-state, is controlled by the chosen value of $p_{min}$ in our proposed Kalman filter. In this example we have chosen $p_{min} = 1.5 \times 10^{-5}$; other values can be seen in Fig. 5.5. When $p_{min}$ is zero, the filter approaches a steady-state where it no longer considers new inputs; when $p_{min}$ is large, it never trusts its estimate too strongly, so it is susceptible to newly observed data that is potentially noisy. Because the ground truth feature state of the tracks is available for this example, we can determine the optimal value of $p_{min}$, that is, the value of $p_{min}$ which minimizes the error between the estimated feature state and the true feature state. We show




(a) (b) (c)

Fig. 5.5: Outputs of the data-fusion pipeline with varying values of $p_{min}$. For (a) $p_{min} = 0$, for (b) $p_{min} = 10^{-5}$ and for (c) $p_{min} = 10^{-4}$.

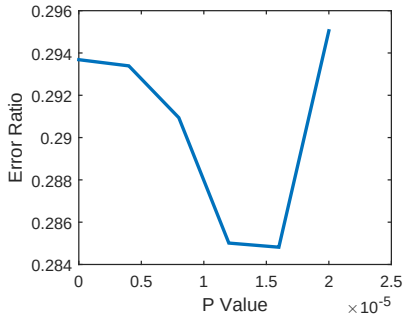
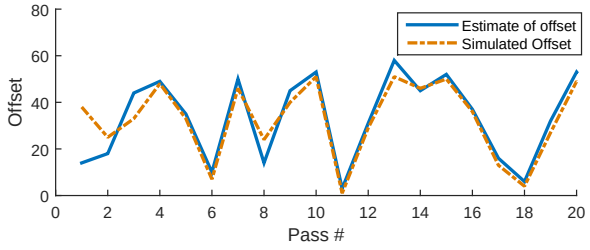



Fig. 5.6: Effect of varying the value of $p_{min}$ on the error of the data fusion. Here, an error ratio of 1 is as bad as the observed data itself, while an error ratio of zero means that the fusion has perfectly reconstructed the ground-truth.

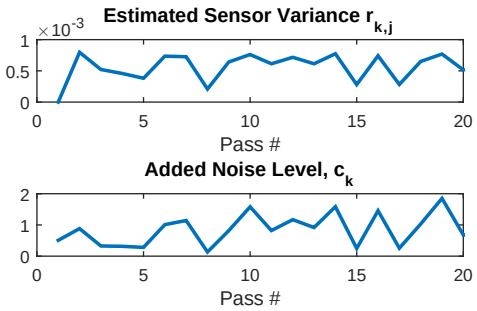
results in terms of error ratio, which is defined as,

$$\text{error ratio} = \sum_k \frac{||\mathbf{x}_k - \hat{\mathbf{x}}_k||_2}{||\mathbf{x}_k - \hat{\mathbf{z}}_k||_2}. \tag{5.15}$$

Here, the numerator is the difference between the true state and the fused data, and the denominator is the difference between the true state and the observed data. Thus, an error ratio of one would mean the fused data is no better than the observed data, and an error ratio of zero would mean the fused data perfectly matches the true state. A plot of the error ratio for various $p_{min}$ values is shown in Fig. 5.6 revealing that $p_{min} = 1.5 \times 10^{-5}$ is approximately optimal. In cases where a change occurs in the tracks, the error function tends to be convex. For this simulated data, the data fusion reduces the noise from the observed data by more than 70%.




(a)




(b)

Fig. 5.7: Estimation of the simulated offset values and the sensor noise level. (a) Comparison of the estimated offset with the simulated offset for the first 20 passes. (b) Comparison of the estimated sensor variance with the simulated noise level, $c_k$, for the same 20 passes.

In addition to studying $p_{min}$, we can also study two other novel components of our data fusion algorithm. The first is our estimation of the position error (or offset) using the cross correlation of the observed data with the estimated state from Eq. (5.8). Fig. 5.7a shows a comparison of how well our estimated offsets match the simulated offsets for one of the two simulated sensors, truncated to show only the first 20 passes for clarity. The true offset and the estimated offset do not line up well for the first several passes because the estimate of the current state of the tracks is not accurate. By the $4^{\text{th}}$ pass, the estimated offset and true offset line up closely.

The second parameter we estimate is the sensor variance, a measure of the noise in the data. When generating the data for this simulation, we multiplied the noise in each pass by a constant, $c_k$, shown in Eq. (5.14), as a way of representing the fact that the noise level tends to be constant for each pass. In our algorithm, we approximate the noise level using Eq. (5.10) by calculating the variance between the feature state estimate and the observed feature. In Fig. 5.7b we show both the estimated sensor variance and the amount of noise which was originally added to the simulated data. Although the values themselves are quite different, both parameters follow a similar trend. One interesting phenomenon is the interplay between the amount of added noise and the estimated offset. Notice that for pass #13 the value of $c_k$ is high; at the same pass, there is a difference between the estimated offset and the true offset in Fig. 5.7a, likely due to large amount of noise. If the estimated offset is incorrect, then the data are not properly aligned, so we

might expect the estimated sensor variance to be higher. However, this variance does not appear to increase dramatically because the features we are fusing together are relatively smooth over the length of the track. If the feature were less smooth, our approach would be more sensitive to misalignment.

## 5.4 Validation on Operational Data

In this section we validate our data fusion approach on data collected from the light-rail network described in Chapter 2. To evaluate the data fusion approach, we analyze data from a section of track where a faulty joint was repaired during the time period when both trains were on-line; the goal is to examine whether the proposed data fusion approach can help detect this repair more accurately and more rapidly using data from both trains than by detecting the change using either train individually.

One of the challenges in this study is that GPS units are low-cost ($<$\$30), do not have differential capabilities, and do not have an unobstructed view of the sky. Together, these factors lead to low-accuracy position estimates. While this makes the analysis challenging, it also means that the techniques are general for many systems, even if the position estimate is poor. In addition to position uncertainty, the train's speed can vary with each pass over the section of track of interest, further complicating comparisons between passes. To address these challenges, we use the signal-energy feature, studied in Chapter 3, which was shown to be robust to high levels of position uncertainty and to varying train speed. This feature consists of squaring the raw vibration data (which are collected in the time-domain), smoothing the squared signal (here we smooth over a period of 0.3 seconds), then interpolating the data spatially. For the interpolation in this paper, we sampled at a rate of 1.5 points per meter. For our analysis, we consider a 1 km section of track; thus the observed data vector of the extracted energy feature, $\mathbf{z}_{k,j}$, for each of the $k$ passes and $j$ sensors has a length of 1500.

The energy feature data for the each of two sensors mounted on both trains are shown in

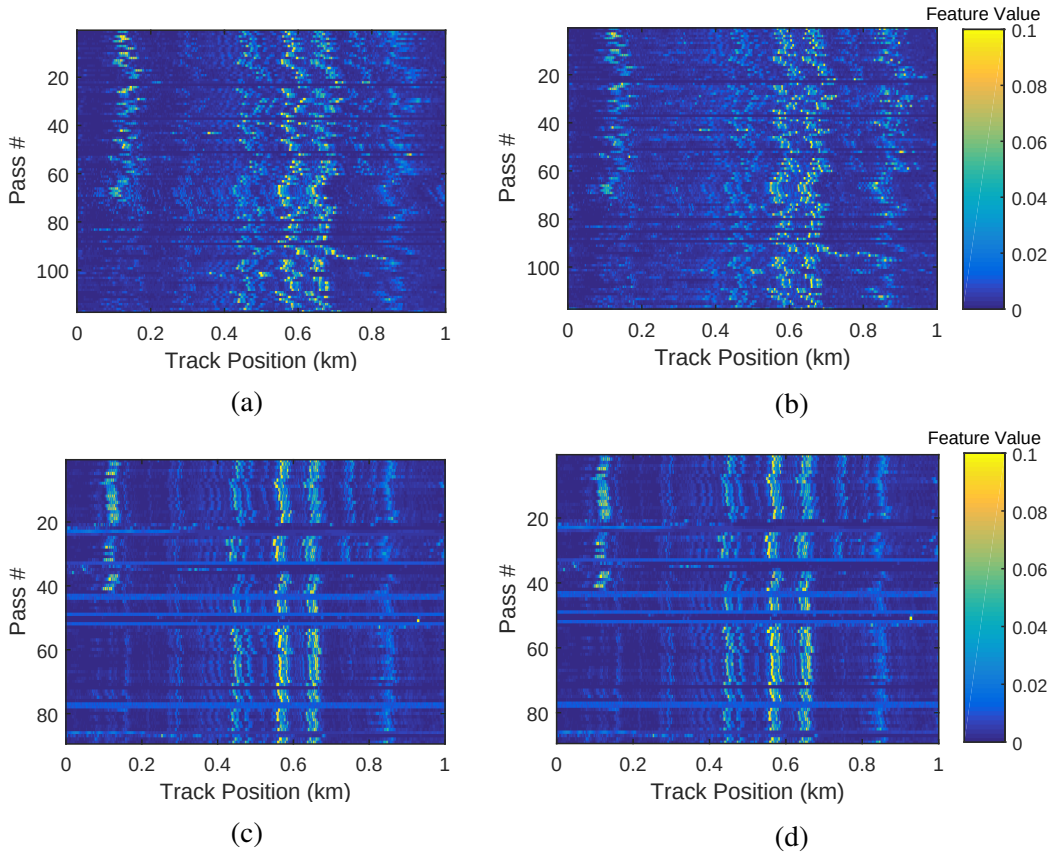
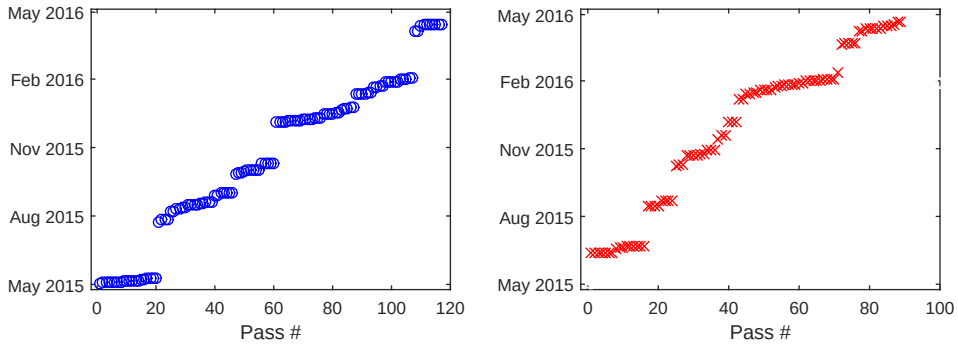



Fig. 5.8: Energy feature representation of the data from the instrumented trains as they pass over a 1 km section of track. Each horizontal line shows the data from one pass, where the color indicates the size of the signal-energy feature. (a) Data from Sensor 1 on Train 1. (b) Data from Sensor 2 on Train 1. (c) Data from Sensor 1 on Train 2. (d) Data from Sensor 2 on Train 2.

Fig. 5.8. Each horizontal line represents one pass where the color indicates the value of the feature. As in the simulation, each pass vector is normalized so it has a length of 1. By plotting each successive pass sequentially in rows, certain patterns of vertical lines emerge; this occurs because certain hardware in the tracks, like the two pieces of switch gear around 0.6 km, consistently cause high vibrations, resulting in large energy feature values. These vertical lines are not perfectly straight because of position uncertainty due to GPS error.

Figs. 5.8a and 5.8b show 117 passes from the two sensors on Train 1. The dates of the passes are shown in Fig. 5.9a. A number of passes have very low values; this can happen for a variety of reasons like a malfunction in the acquisition hardware or because a very large vibration event




(a) Train 1 (b) Train 2

Fig. 5.9: Dates of the passes shown in Fig. 5.8. (a) Dates of the passes from Train 1. (b) Dates of the passes from Train 2.

occurred, and due to the normalization, all other values were reduced accordingly. The plots from each of the two sensors are nearly identical; this is both good and bad. It is good because it means that sensors placed within the cabin of the train are relatively insensitive to position, so they can be placed in future instrumentations wherever is most convenient. It is bad because the two sensors are so closely correlated, that when fusing the data, relatively little additional information can be gained. Of particular interest is the vertical line around 0.1 km which was due to high vibrations from a faulty track joint; this joint was replaced on December 15, 2015, or after pass #72; accordingly, the energy at this location of the track is reduced.

Figs. 5.8c and 5.8d show 89 passes from Train 2; the dates of the passes are shown in Fig. 5.9b. Here too we see a change in the data after the track joint was repaired, at 0.1 km, in this case after pass #47. Notice that there appears to be less GPS error in the data from the second train (i.e. the vertical lines are "straighter"). This likely has to do with the position of the GPS antenna. In Train 1, the antenna was placed under the metal roof so it had no direct view of the sky; in Train 2, the GPS was placed in the interurban light enclosure shown in Fig. 2.2 so it had a partial view of the sky.

If we view the data collected by each pass as an observation of the current state of the tracks, the question is whether we can combine data from multiple passes and between the two trains to build a better estimate of the state of the tracks than is given in any one pass. In the next

section we explore this question by combining the collected data using our proposed data fusion approach.

### 5.4.1 Data Fusion

In this section we explore fusion both at the train level, i.e. combining extracted features from different sensors, and at the network level, i.e. combining extracted features from different trains. In both cases, however, perhaps the most important component of fusion is combining features between passes, because as we have seen, individual passes can have abnormally low values or high position offsets, both of which are handled by our proposed fusion approach.

The estimate of the state of the tracks found by fusing together features extracted from Train 1 can be seen in Fig. 5.10a. Our approach is applied in an on-line manner; the estimate at a given pass relies only on the data from that pass and the data from the previous passes. Note that the large features values from the switch gear at 0.6 km end up controlling the alignment process, so the vertical line around 0.6 km is almost perfectly straight. The energy from the faulty track joint at 0.1 km is less straight, because it does not cause enough vibrations to control the alignment and is relatively far away from the track gear which does. However, the fusion appears to make the change in the tracks at 0.1 km more visible. In addition to the repair of the faulty joint at 0.1 km performed after pass #72, there is another track change which becomes apparent through the data fusion process: a correction to the alignment of the tracks around 0.8 km that was performed after pass #61. This is the result of tamping performed in November of 2015. The visibility of this change is it offers qualitative evidence of the value of fusion.

Fig. 5.10b shows the fused estimate of the state of the tracks using data from Train 2. Because more of the passes from Train 2 were spurious, the fused estimate of the track state using the data from Train 2 appears noisier than the estimate from Train 1. Note that the alignment for Train 2 differs from the alignment found from Train 1. In Train 1, the energy from the first switch occurs at 0.6 km, with some high values occurring just after 0.6 km. In the case of the second train, the

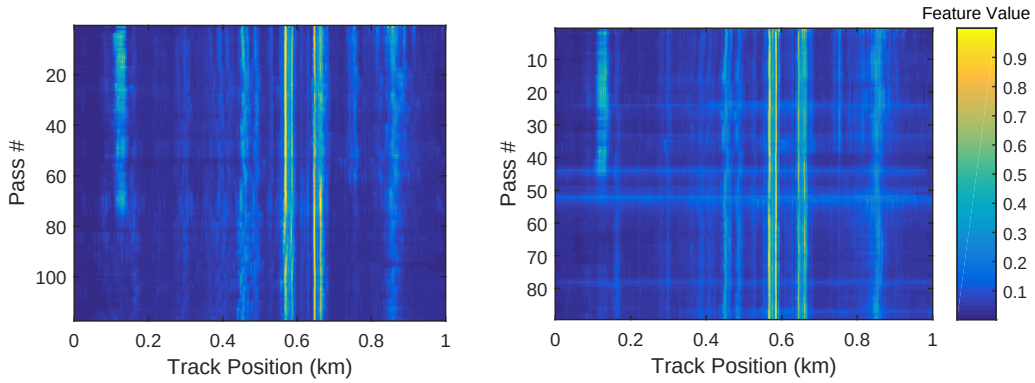



Fig. 5.10: Fused estimate of the state of tracks based on data from (a) Train 1 and (b) Train 2. In both cases we use $p_{min} = 1 \times 10^{-5}$.

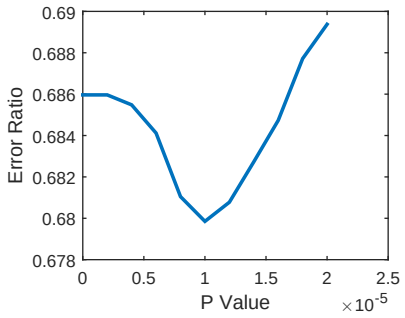
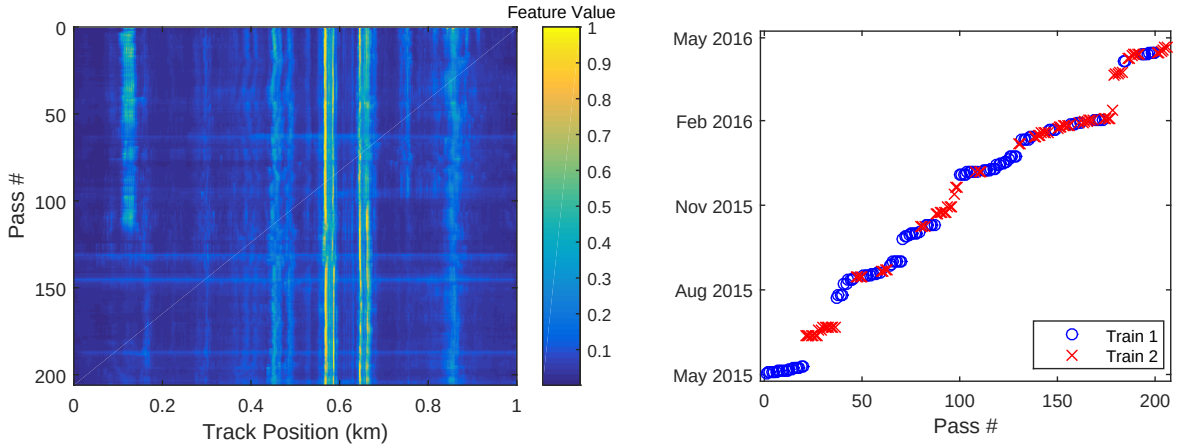



Fig. 5.11: Effect of the minimum prediction error value, $p_{min}$ on the estimated error produced by fusing operational data.

entirety of the high values from the first switch (a vertical line in the figure) occurs prior to 0.6 km; in essence, the data from the second train is shifted slightly to the left when compared to the data from the first train. This alignment is something which must be considered when fusing the data from both of the trains.

One benefit of the proposed data fusion approach is that there is only one parameter to set, $p_{min}$; however, a discussion of how that parameter was chosen is necessary. In the simulation considered in the previous section, the optimal $p_{min}$ value could be found because the ground truth was known. In this case, the ground truth is unknown. One method might be to choose $p_{min}$ empirically such that the fusion approach appears to smooth the data but not smooth it so much that some track changes are obscured. Doing so indicates that the value around $p_{min} = 1 \times 10^{-5}$




(a) (b)

Fig. 5.12: (a) Fused estimate for the state of the tracks combining data from both Train 1 and Train 2. (b) The dates that the passes occurred from both Train 1 and Train 2.

is reasonable. However a more quantitative approach is possible if a known change in the tracks occurs. In this case, considering the data from Train 1, we know that the track was repaired after pass #72. We can assume that the tracks are in one state (state 1) from pass 1-72, and another state (state 2) from pass 73-117. Using this assumption we take the "ground truth" for state 1 to be simply the average of the features for the first 72 passes shown in Figs. 5.8a and 5.8b, and the "ground truth" for the second pass, the average of passes 73-117. Just as in the simulation, we can calculate how the error ratio changes with different values of $p_{min}$ as shown in Fig. 5.11. Again we see a convex function and, in this case, we see that $p_{min} = 1 \times 10^{-5}$ is indeed optimal. In general, this technique only works for finding the optimal value of $p_{min}$ in cases were a known change occurs; without a known change the error function is not necessarily convex. However, this pseudo-simulation procedure could be used on *a priori* known change to estimate the optimal $p_{min}$ value for a section track.

Using this same $p_{min}$ value and the procedures outlined above, we can fuse data from both trains. The first step is to interlace the passes according to the date and time when they were recorded, as seen in Fig. 5.12b. If this were applied in an on-line setting, it would be equivalent to processing the newest collected data over a section of track as it was acquired. The estimate of

the state of tracks from combining both trains is shown in Fig. 5.12a. In this case, we consider only two sensors on each train, however, different number of sensors on each train could be handled similarly. Because of the alignment step, the difference in GPS error between the trains is handled, and the combined estimate appears to be as consistent as the data from either of the trains alone.

Thus far, in applying our data fusion approach on operational data, we have only been able to offer qualitative evidence that combining multiple sources of data improves the estimate versus analyzing the features independently. Part of this stems from the fact that the ground truth of the state of the tracks in terms of the energy feature is not known. In the next section, we introduce an anomaly detection method to evaluate the proposed data-fusion as part of the complete data processing pipeline shown in Fig. 5.1.

### 5.4.2 Evaluation of Data Fusion

In order to evaluate the fusion of the operational data, it is first necessary to consider the objectives of rail operators when conducting an inspection. As mentioned in section 5.1, operators want inspection techniques that are low-cost and reliable, and detect faults soon after they occur. We will evaluate our data fusion approach in-terms of the second two objectives, reliability and time of detection, because, in practice, the "ground truth" feature state of the tracks for the operational data is not known. Since data fusion is but one step in an overall data processing pipeline, we must introduce an anomaly or change detection approach (change detection and anomaly detection are used here interchangeably) to assist in evaluating the effect of our proposed Level 1 Fusion on the overall detection performance.

In Chapter 3, we evaluated three change detection approaches and found a Haar filter to be the most effective for train-based monitoring when using energy features. Building on this work, we will use the Haar filter to assess how accurately and how rapidly changes can be detected both in analyzing the extracted features independently and in analyzing the fused features. Fig. 5.13

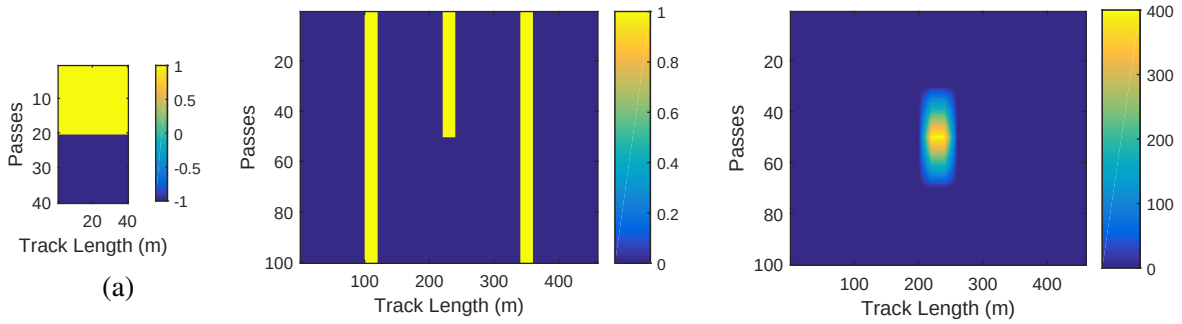
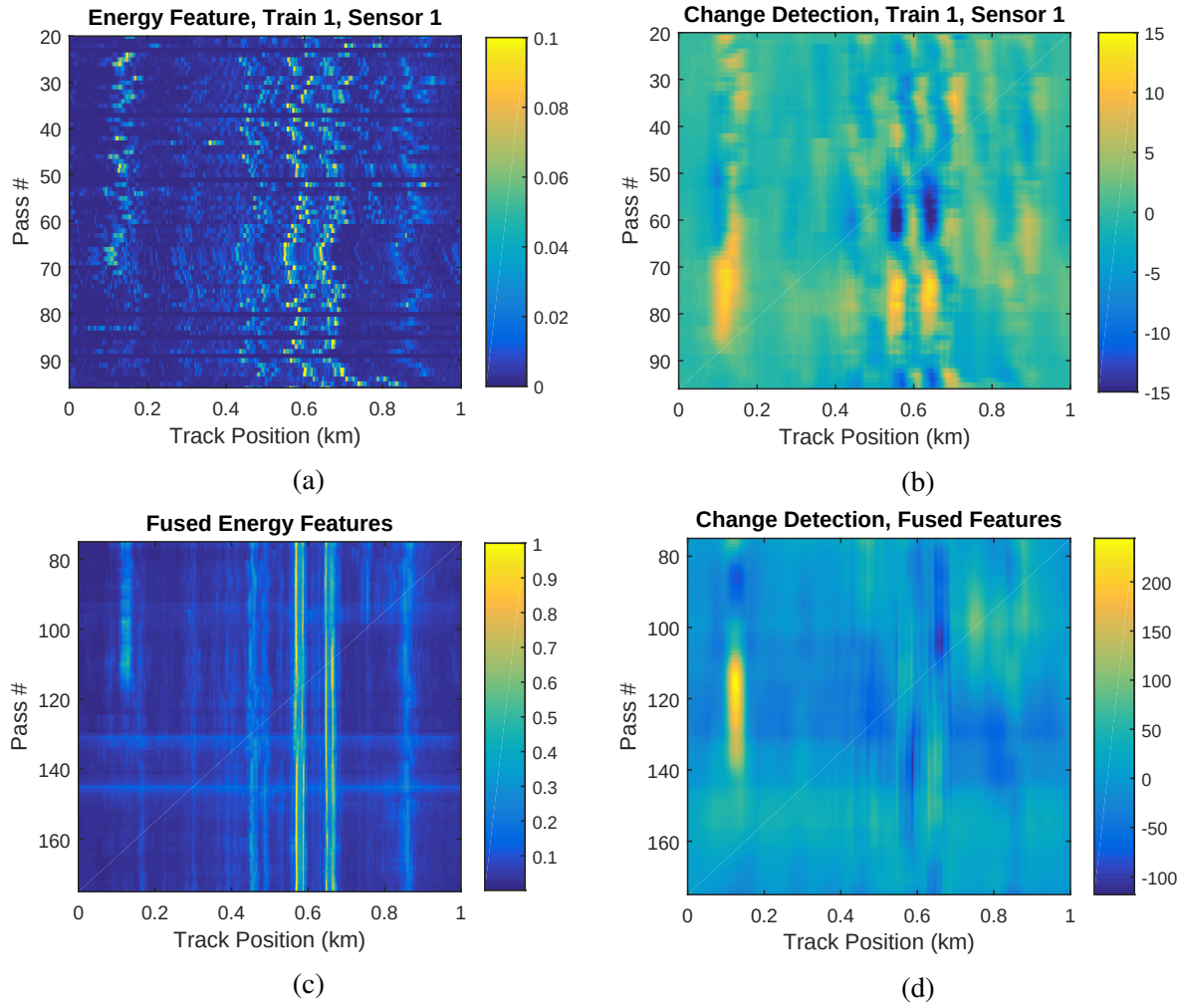



Fig. 5.13: An example of change detection with the Haar filter. (a) The Haar template, (b) a simplistic example of data with a change, and (c) the result of filtering the example data in (b) with the Haar template shown in (a). Note that only the vertical change is detected.

shows an example of how change detection with a Haar filter works. A Haar template is shown in Fig. 5.13a; this template is convolved with the input data shown in Fig. 5.13b. The result, shown in Fig. 5.13c, highlights locations in the input data where a vertical change occurs. We are only interested in "vertical changes" given the plotting convention in this study where each pass is shown as a row; such changes indicate that the state of the tracks has changed from one pass to another. Note that the Haar template has two blocks, one where values are equal to 1 and another where the values are -1. The height of each block, what we will refer to as the support length, controls how many passes must be collected before a change is detected. In this example, the support length is 20, meaning 20 passes would have to be collected after a change before the peak change detection value will occur. We will explore a variety of different support lengths in Fig. 5.15.

Using the same parameters (support length of 20 passes over a track width of 40 meters) we can apply the Haar filter to some operational data. The ultimate goal is to determine whethere the proposed fusion algorithm makes change detection easier. Thus we apply the Haar filter on the features prior to fusion and after fusion. The results are shown in Fig. 5.14. In both examples, the change in the tracks at 0.1 km is detected to some extent. What differs is how well that change is detected relative to other erroneous changes. In Fig. 5.14b, the change detection results from




(a) (b) (c) (d)

Fig. 5.14: Change detection results for different levels of data fusion. The raw data are shown on the left panels; the right panels show the result of applying the Haar filter. (a) The raw energy feature data from Train 1 and Sensor 1 for select passes of interest. (b) The resultant change detection output for the raw feature data. Notice the magnitude of erroneous changes (at 0.6 km) are higher than the true change at 0.1 km. (c) The data fused from both trains for select passes of interest. (d) The resultant change detection output for the fused data from both trains. This has a better result, with the magnitude of true change 1.95x higher than the magnitude of any other change.

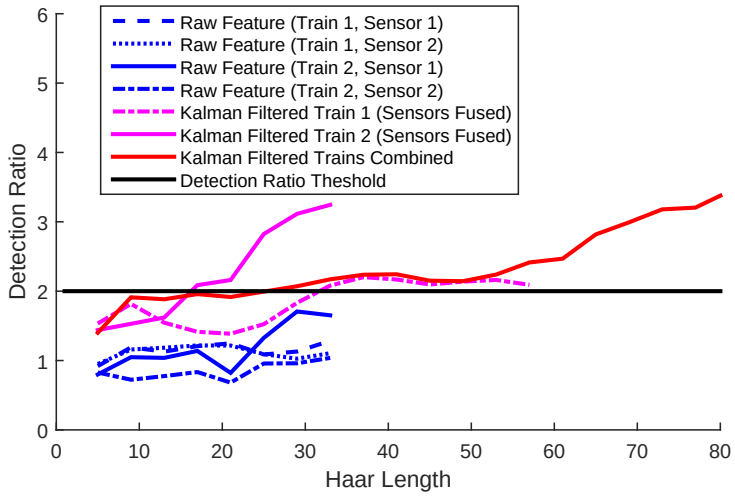



Fig. 5.15: Change detection results for Haar templates with different support lengths. Here the detection ratio is the ratio of the magnitude of the true change to erroneous changes. The longer the support length (the # of passes in Fig. 5.13a) the longer it would take to detect a change in the tracks. Note that the raw feature data never achieves a detection ratio above 2, independently of length, so is not reliable. Train 1 achieves a detection ratio of 2 considering 33 passes; Train 2 achieves this ratio after just 17 passes. Using the combined data, this ratio is achieve in 24 passes. This information is shown in tabular format in Table 1.

analyzing raw features directly, the erroneous changes detected in the switch gear (around 0.6 km) are larger than the changes detected from the actual track change at 0.1 km. In this case, the "detection ratio," the ratio of the magnitude of the true changes to the magnitude of the erroneous changes, is below 1. A similar change detection analysis on the fused data, Fig. 5.14d, produces a much improved detection ratio of 1.9.

In addition to the change from the repair of the faulty joint, a second change can be seen in the change detection results from the fused data. As discussed previously, a portion of the track (around 0.75 km) was tamped after pass # 98. Although this change is nearly impossible to detect in the raw features, the change is evident in the fused data shown in Fig. 5.14c and the second highest change detection value occurs at 0.75 km and pass # 98 in Fig. 5.14d. Although this tamping change is not the focus of this study, it is noteworthy as evidence that the proposed fusion technique might be helpful in detecting more subtle track changes.

Thus far we have assumed that the support length of the Haar template is always 20 passes;

| | Before Change | | Change Detected | | Detection Period | |
|---|---|---|---|---|---|---|
| | Last Pass # | Date | Pass # | Date | # of Passes | # of Days |
| Train 1 | 72 | Dec 8 2015 | 104 | Jan 26 2016 | 32 | 49 |
| Train 2 | 41 | Dec 8 2015 | 58 | Feb 1 2016 | 17 | 55 |
| Combined | 113 | Dec 8 2015 | 137 | Jan 14 2016 | 24 | 37 |

Table 5.1: Detection table, assuming "detection" occurs when the detection ratio exceeds 2. Note that information on the features prior to fusion is not shown as change detection on the raw feature does not achieve a detection ratio above 2.

this is an important parameter to explore further as it controls how soon the changes could be detected. We can vary the support length while reporting the resultant detection ratio, as shown in Fig. 5.15. The results of analyzing the features independently are shown in blue; the detection ratio hovers around 1, meaning the change in the tracks is roughly equivalent to erroneous changes. In other words, true changes cannot be detected reliably. We only show support lengths of up to 30 passes for the raw data due to the limited number of passes.

Fusing data from a single train tends to produce higher detection ratios, shown in magenta in Fig. 5.15. The data from Train 2 offers particularly high detection ratios, perhaps because the GPS alignment between passes is less noisy. The detection ratio produced when fusing together all the data is shown in red. This complete fusion appears slightly better than Train 1 and slightly worse than Train 2 in terms of detection accuracy for a given number of passes. However, the benefit of fusing all the data together is evident in how soon a change can be detected (in days) rather than simply in terms of the number of passes.

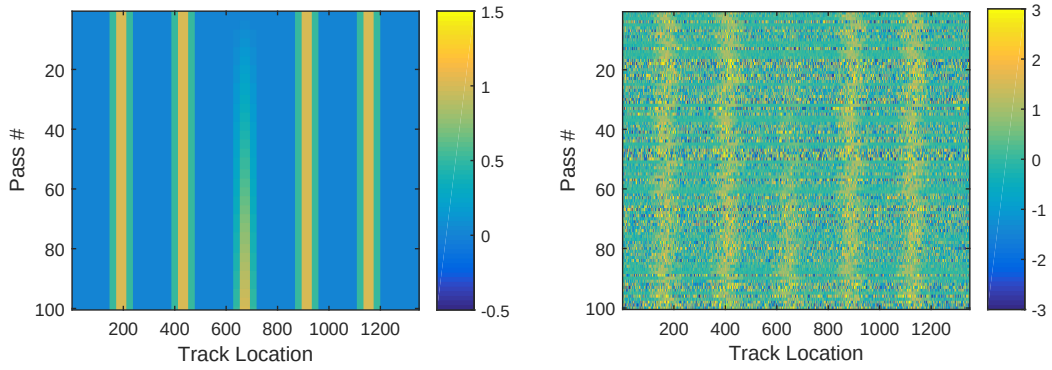
To illustrate the strength of fusing the data from both trains together, let us consider a hypothetical detection scenario using the historical data we have collected. In this case, the last pass recorded before the track was repaired was on Dec 8th, 2015 (repair work started the following day). Let us assume the the minimum acceptable detection ratio is 2 (i.e. a detection event is triggered if one change is twice as high as all other changes). The question is how soon a detection event will be triggered assuming that the data are analyzed after each successive pass, and that in each analysis, Haar templates with different support lengths are tried. If only data

from Train 1 were to be used, a detection ratio of 2 would be achieved on Jan 26th 2016, using a Haar template with a support length of 33 (as is shown in Table 1). This would mean detection would require 49 days. If only data from Train 2 were to be used, detection would occur on Feb 1 2016, with a support length of 17. Note that fewer passes are required from Train 2, but that the data was collected less often, so the overall detection time is longer. If data from both trains is considered (i.e. all the data is fused together), detection would occur on Jan 14th, 2016, with a support length of 24. Thus by fusing the data from both trains, the change could be detected sooner than if either train was analyzed on its own.

## 5.5 Gradual Change

The majority of this thesis has focused on detecting sudden changes in rail-infrastructure, i.e. changes that occur from one pass over the tracks to another. Sudden changes are the type of change most readily detected in an unsupervised setting, and a variety of rail anomalies occur suddenly. In part, we have focused on sudden changes, particularly maintenance work, because these changes are well documented so validating detected changes is possible. However, many rail anomalies occur more gradually and detecting gradual deterioration can prevent later catastrophic failure like train derailment. This section explores whether our data analysis pipeline can be used to identify such gradual changes.

The key difference between the detection of sudden changes and the detection of gradual changes is the time-scale. While sudden changes can be detected by comparing data between passes occurring on different days, detecting gradual changes requires comparing data from different months or years. To consider a longer time-scale, we can simply use a detection template tailored for gradual change detection. The other steps in our data-processing pipeline, like the proposed fusion approach, remain valid.




(a) True Feature State

(b) Observed Features

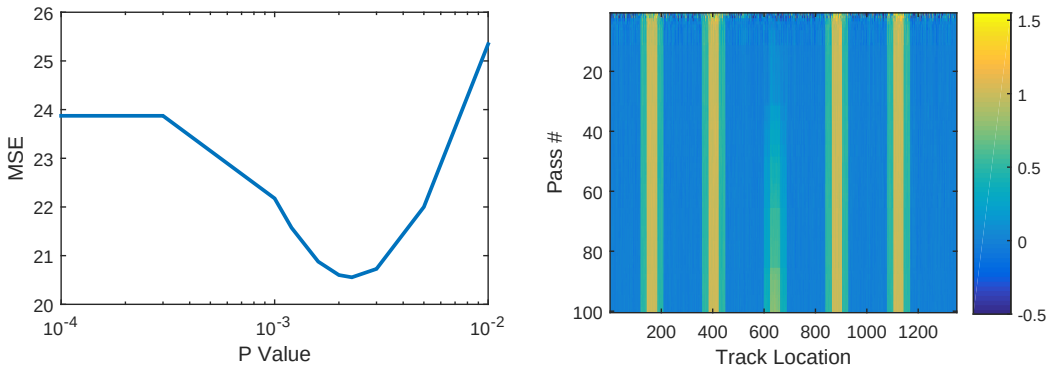
Fig. 5.16: Simulated data with a gradual change. (a) The true feature state of the tracks with a linear change over time at location 670. (b) The noisy observed features.

### 5.5.1 Gradual Change Detection Method

In the previous section, one of our objectives was to detect faults soon after they occur. In the case of gradual changes, it not possible to determine a specific point when gradual deterioration occurs. Instead, we only want to localize where in the tracks gradual changes are occurring. This may be of interest to a manager of a rail-network for prioritizing certain certain sections of track for further inspection.

To explore this idea, we first simulate a gradual change, shown in Fig. 5.16a. We assume that the gradual change is a linear increase in the feature value with each pass at track location 670, representing, for example, the rails becoming increasingly misaligned. The other high values shown in Fig. 5.16a (at locations 190, 430, 910 and 1150) are assumed to be switchgear or other track hardware which naturally generates high feature values and should not be detected.

Following the same method in Eq. 5.14, we can generate the observations of the gradual change as shown in Fig. 5.16b. Note that while in Section 5.3 we normalized the data, here the data is not normalized. We follow the same procedure when analyzing operational data later in this section; gradual changes are so slight that normalization could obscure them or make them appear when in fact a gradual change has not occurred. We can combine passes of this noisy data using our proposed data fusion approach to estimate the track state shown in Fig. 5.17b.

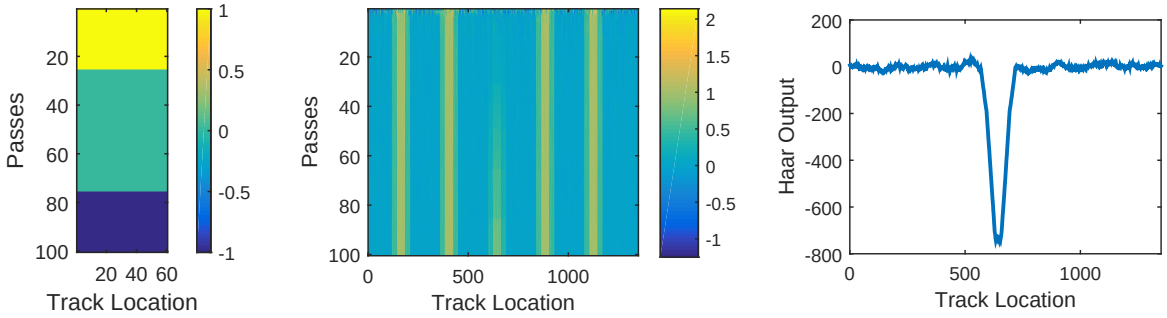



(a) Selection of $p_{min}$ (b) Fusion Output

Fig. 5.17: Result of data fusion approach. (a) Because the ground-truth is known in simulation, we can determine optimal value for $p_{min}$. (b) The output of the fusion approach using $p_{min} = 3.2 \times 10^{-3}$ with the colorbar set to match the true feature state data in Fig. 5.16b.

The similarity between this plot and the true feature state of the tracks (shown in Fig. 5.16a) compared to the observed data (shown in Fig. 5.16b) demonstrates the value of data fusion. The one parameter which we need to select for this fusion is $p_{min}$, the minimum prediction error at each iteration of the data fusion approach. If we plot error as a function of $p_{min}$, as shown in Fig. 5.17a, we see it is a convex function with a minimum value at $p_{min} = 3.2 \times 10^{-3}$. This is the value used to produce the fused estimate in Fig. 5.17b.

The last step in our data processing pipeline is to localize where the gradual change occurs. We use a split Haar template, similar to the one shown in Fig. 5.13a. However here the positive and negative blocks are separated by zeros. This separation means that the positive and negative blocks apply to data separated in time, allowing for the gradual change to become more significant. This template is convolved with the output from our data fusion process; the result is plotted in Fig. 5.18c. The gradual change is detected by the large negative value; had the gradual change been a decrease in the feature value over time, the change would have been detected by a positive value. The location of the extreme value, at track location 690, successfully localizes the gradual change. Note that the output plotted in Fig. 5.18c is a vector, while the output shown in Fig. 5.13c was a matrix. This is because the number of passes in the template is equal to the number of passes in Fig. 5.18b, so convolution is equivalent to sliding the template across the




(a) Split Haar Template (b) Fusion Output (c) Change Detection

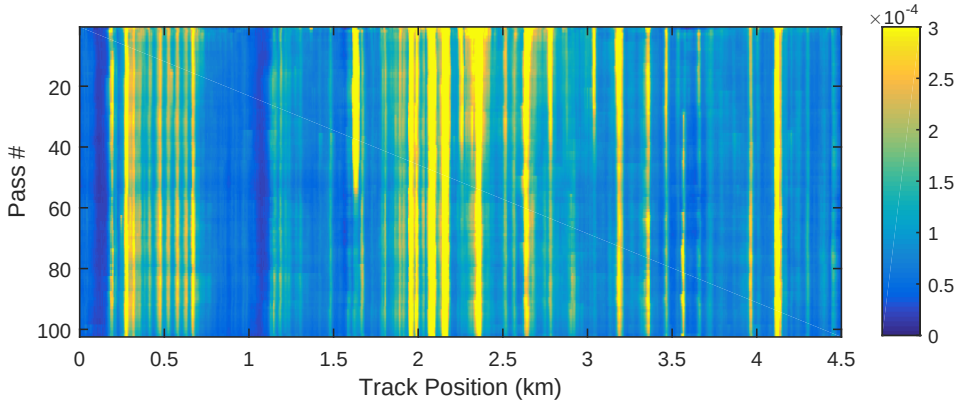
Fig. 5.18: An example of change detection with the split Haar filter. (a) The template is separated by zeros to test the difference between two distinct time periods. (b) The data to which the Haar template is applied. (c) Change detection output note. Note that here the height of the template matches the height of the input data, so the output is a vector (plotted here as a line).

data horizontally; if the template has fewer passes than the input data, a 2D dimensional output is produced.

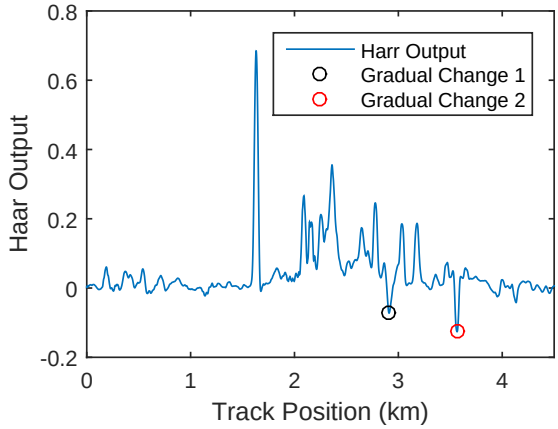
### 5.5.2 A Validation of the Gradual Change Detection Method

We can apply this same method to operational train data. In this case, we use data from Train 2 over a 5.5km section of track; the one 1km section of track studied earlier is included in this data-set and spans the distance from 1.5km to 2.5km. Note that the fused data from Train 2, shown in Fig. 5.19a, appears smoother than the data shown previously, because in this case, the data has not been normalized. In an attempt to find gradual changes, we convolve a split Haar template with this data producing the results shown in Fig. 5.19b; in this case, the template has a width (along the length of the track) of 30m as in the previous section.

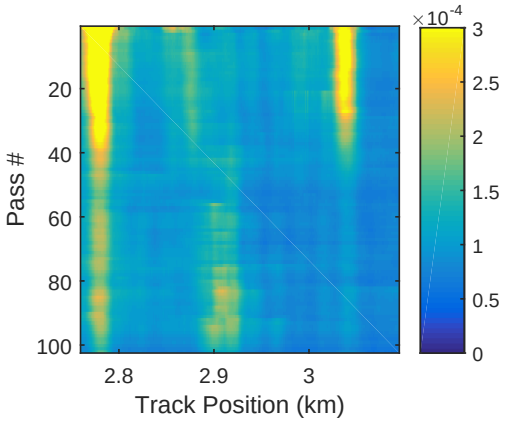
In general, positive change values indicate maintenance activities because the sign means the feature decreases over time, and the feature here is indicative of track roughness. Similarly, negative change values indicate an increase in the feature over time, which could mean a deterioration in the track geometry. The largest amplitude in the change detection output occurs around 1.7km and represents the repair of the misaligned track joint after pass #41 studied in the previous section. Between 2.1km and 3.3km there are a number of positive peaks; these result from tamping



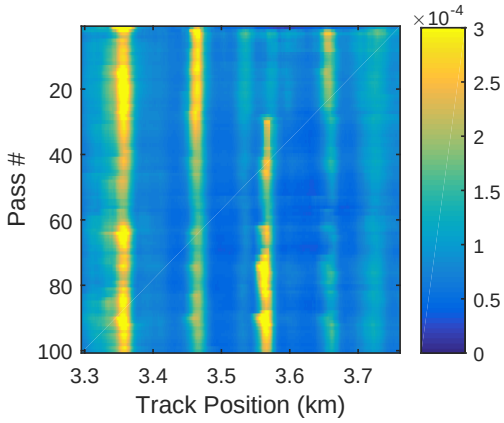

(a) Fused output, energy feature from Train 2




(b) Change Detection Output




(c) Detail of gradual change 1




(d) Detail of gradual change 2

Fig. 5.19: An example of change detection with the split Haar filter. (a) The template is separated by zeros to test the difference between two distinct time periods. (b) The data to which the Haar template is applied. (c) Change detection output note. Note that here the height of the template matches the height of input data, so the output is a vector (plotted here as a line).

in this area after pass #27 and #33 (the tamping took place over a week). Thus far these positive changes are actually sudden changes which are also detected by the gradual detection template; these could be removed by cross checking with the results of sudden detection.

The most intriguing results are the negative change values around 2.9km and 3.57km. Greater detail of the feature values for each of these changes are shown in Fig. 5.19c and Fig. 5.19d; the data shows an increase in signal energy over time at these locations, which could well indicate deterioration in the track leading to a bumpier train ride. Although, we cannot verify whether deterioration occurred because we do not have ground truth information on the state of the tracks, we can infer that our proposed analysis pipeline is potentially suitable for detecting gradual changes.

## 5.6 Discussion

In this chapter, we have presented a novel data fusion approach for data-driven track-monitoring from in-service trains. The goal is to build a formal method to combine extracted features from multiple passes over the tracks from multiple sensors on multiple trains. We overcome some of the challenges inherent in combining data from different sources with a novel adaptive Kalman filter. The approach is computationally efficient and easy to implement as it requires choosing only a single parameter, prediction error variance, which can be approximated empirically or through simulation. We use a data-driven approach for detecting track irregularities by comparing current data to a historical baseline. Thus, combining features from multiple trains allows for a more reliable baseline and more rapid detection once a change has occurred.

The benefits of our proposed data fusion approach are shown both with a simulated example, and an example from the operational train system. In the case of the latter, a faulty track joint was repaired, and data from two operational trains passing over the joint both before and after the repair was used to detect when the repair occurred. Furthermore, by changing the detection template, our approach appears capable of detecting gradual changes in the network which may

be indicative of deterioration in the track. The performance of the proposed method indicates that fusing data helps to detect track changes and could be an important component in a data processing pipeline for network-level track monitoring from in-service trains.

## 5.7 Future Work

Future work should explore fusing other types of features, and, data from heterogeneous sensors. Of particular interest would be using the type of low-cost sensors found in smartphones. By allowing smartphone data to be included in track models, crowd-sourcing data for track monitoring would be possible. This would allow the proposed method to scale more rapidly to much larger track networks.

In addition, throughout this chapter we have assumed that the position offset due to GPS error was constant over a single pass. While this assumption appeared to be valid in many of the cases we examined, future studies should investigate the validity of this assumption in more detail. Intuitively, as the length of track of interest increases, the position offset within each pass is more likely to vary. Instead of using the cross correlation to determine a single value for the position offset, future work might use an Extended Kalman Filter (EKF) to determine the position of the train more accurately, using both GPS and inertial data from the train.

# Chapter 6

# Conclusions

In this thesis we have argued that information on the state of a track network can be extracted from vibration data collected from in-service trains. In particular we have explored suitable analysis techniques, and have found data-driven approaches to be promising. We have studied methods to model the track state, ways to detect track irregularities, and approaches to fuse different data sources. We have validated our proposed approaches on a large dataset collected from two instrumented trains over more than two years. The detailed conclusions resulting from this work are presented as responses to the research questions posed in section 1.7.

**RQ1: How can the state of the tracks be modeled from the collected data, and in particular, how can operational variability be mitigated when building track models?**

We found two distinct methods for mitigating the operational variability of the train. Either features robust to these sources of variability could be used in building the track model, what we called implicit modeling, or this variability could be handled by building our model using the physics of the problem, what we called explicit modeling. In the former case, we found signal-energy to be the best feature. In the case of physics-based models, given the limited amount of information we gleaned about the tracks from the collected vibration data, additional constraints were needed to make the problem stable. We found imposing sparsity on the solution yielded stable results that provided insight into the track condition.

For the implicit model, features were extracted from the signal to imply the state of the tracks; the focus was on selecting features robust to these sources of uncertainty. Of the features we tested, we found signal-energy features to be the most robust. However, some of the same properties of the features that make them robust, like the fact that they were averaged over a length of track, also made them describe the condition of the tracks less precisely. Thus these features allowed for the detection of track changes, but they did not provide insight into the natural of the change.

For the explicit model, the track profile was estimated by solving an inverse problem. Without constraints, such a problem is ill-posed, but we found that by enforcing sparsity in the identified track profile, stable solutions could be achieved. Explicit models, due to their physics-based formulation, account for the train's variable speed. Furthermore, this method minimized position uncertainty because the train could be localized relative to the few identified bumps (a process known as map-matching). For the several examples we tested, explicit models worked well, identifying the change in the tracks, and the nature of the change in the tracks.

The shortcoming of our proposed sparse approach is that it works for sections of track where the train is excited by particular aspects of the track, such as switch gear or track joints, and thus may not work for all track layouts. However, given that particular track hardware like joints and switchgear cause many of the track-related accidents, the sparse approach may be a useful modeling tool, particularly given its ability to handle varying operational conditions.

**RQ2: How rapidly and how reliably can relevant changes in rail-infrastructure be detected from an in-service train, and how can detection occur in an unsupervised fashion?** Given our data-driven approach, our ability to detect changes in the tracks related directly to our ability to accurately model the tracks from the collected data. The optimal change detection technique thus depended on the type of track model.

For implicit track models, we tested two common unsupervised change detection approaches, CUSUM and GLR, as well as a Haar template, and found that the Haar template, perhaps the most simplistic approach, worked best. It should be noted that the changes we studied were primarily

sudden track changes, as the time and location where the change occurred can be found in an unsupervised fashion. However, detecting the location of gradual changes may also be of interest to network managers, and we found that a split Haar template could localize such changes.

For the explicit model, the track state consisted of the height of the found sparse bumps; given the low dimensionality of the feature, we found a simple threshold was able to detect a change in the tracks. How well the relevant changes could be detected again relates to the quality of the track model. Using the track features directly, we found that the change detection approaches were only slightly more likely to identify true track change than spurious changes. However, by fusing data from multiple passes and multiple trains, true changes were far more likely to be detected than spurious changes, and could be detected in a fewer number of days.

**RQ3: How can a data-driven monitoring framework be scaled-up to include data from multiple trains across an entire rail-network?**

There are several challenges inherent in scaling-up such monitoring; we addressed this question by first identifying the challenges, then developing a data fusion approach which addressed them.

For example, as more sensors are included in the track model, the probability that at least one sensor is faulty increases. We found this could be addressed by using a novel type of adaptive Kalman filer that weighted the data from each sensor and from each pass according to how similar the data was to historical values. This technique worked even in cases where a change occurred in the tracks, because these changes tend to be localized, so the data from functioning sensors still resembled historical data.

In general, we found that through data fusion, data from multiple sensors, multiple passes, and multiple trains could be combined and that the resultant change detection was more reliable and could detect changes more rapidly than analyzing the data independently.

# Chapter 7

# Future Work

This thesis provides a data-driven framework for track monitoring from in-service trains, and presents several approaches to facilitate this type of monitoring. But the work presented thus far is just the beginning. This chapter presents the potential next steps in further developing the proposed technology. In particular, we discuss the data-collection tools that would be required to study network-level infrastructure monitoring in more detail and we discuss the applicability of the proposed methods within other domains.

## 7.1 Future Dataset

Throughout this thesis, many of the novel approaches were first tested on simulated data, then validated on operational data. We had to rely on simulated data because we did not have a large enough number of known track changes in the operational data, that testing on them would be statistically significant. This situation could be remedied by a two-fold process: first collecting vibration data from more trains (over possibly a larger track network), and second, gathering more information from track inspectors about the condition of the track.

The hundreds of passes over each section of track that we collected from two trains over a three year span exceeded other datasets which have been collected from operational trains

(a) (b)

Figure 7.1: Collecting data from a smart phone. (a) Shows the device resting on the train. (b) Shows the device in the hand of the user.

for the purpose of track monitoring. However, each day, hundreds of trains pass over each section of track; if all 83 trains in Pittsburgh's light-rail network were instrumented, two orders of magnitude more passes could have been collected. We were often confined to looking at severe track changes because we did not have enough passes so that more subtle track changes could be detected with statistical significance.

In Section 5.4.2, detecting a change in the tracks at the desired accuracy level took 24 passes when analyzing data from two trains. Given how few trains were instrumented, collecting these passes took 37 days. However, on the day the tracks reopened for service after maintenance, 24 trains passed over the section of track of interest in the first 4 hours of the day. Had more trains been instrumented, detection could have occurred far more rapidly.

While type of data-acquisition system we built could be deployed on a larger number of trains, the quality the sensors and the acquisition system may not be necessary. A more economical solution may be to place consumer devices, like smartphones, on the trains; these devices have all the required sensors and can transmit the collected data to a central repository over cellular networks.

Preliminary data collected from a cellphone in Fig. 7.1a indicates that it is roughly equivalent


Figure 7.2: A potential user interface

to the data collected by the sensors we used mounted inside the cabin of the trains. Data collected from a cell phone in use by a passenger (in this case, the author), shown in Fig. 7.1b could be equally valuable. However an easier solution still may be to crowd-source data from the smart phones of consenting passengers. This would allow the system to scale much more readily.

But it is not just more raw data from the train that needs to be collected. Additional information is needed about the state of the tracks from the track inspectors. In this thesis we studied how to detect only a handful of changes, like misaligned joints, a faulty road crossing, and the effects of tamping on the tracks. This is largely due to the limited information we have received about the state of the tracks. Because of this limitation, we were only able to verify our approaches on these discrete events.

To address this, one solution may be to build a user interface, preferably on a mobile device, that allows inspectors to input track information in a way that this information could be directly used by the monitoring algorithm. One realization of such an interface is shown in Fig. 7.2.

Work by Mascareñas et al. [39] has shown the importance of keeping a human in the loop. While data-driven analysis of tracks has many benefits over the qualitative visual inspection

used today, perhaps the most effective system would be a hybrid of the two. Humans have a strong ability to make sense of unstructured data, and experienced inspectors accumulate valuable knowledge on how to infer the state of the tracks from limited information. Such a human-computer interface could try to leverage the strengths of the human inspectors while empowering them with rich data visualizations.

## 7.2 Applicability in Other Domains

While the work to date has only been validated on passenger rail, we believe these methods could be applied to many other domains. Some uses might include track monitoring in other rail applications, other types of vehicle-based infrastructure monitoring, and even applications outside of the civil-engineering domain. We discuss an example of each in the following section, along with the challenges and benefits of using the methods proposed in this thesis.

In the United States, as in many countries, freight rail is a larger industry than passenger rail, so adopting the techniques to monitoring tracks from freight trains would be an important extension. While some sources of noise would be reduced (typically there is no ventilation system) other challenges may exist such as different responses between when the train cars are full of cargo versus when they are empty, a change which would be far more drastic than a full versus an empty passenger train. Variation due to loading could be handled by first determining whether each train-car is loaded or unloaded, then comparing the recorded data to the historical data for that loading condition.

More widely applicable than freight, would be to modify the proposed method to monitor road condition from operational motor vehicles. Some researchers have used motor vehicles for monitoring particular aspects of roads, such as detecting potholes [19] or tracking the mode shapes of bridges [53]. Our data-driven method, where the current state of the road could be compared to the historical state of the road, would offer a simple way to monitor many aspects of road health from accelerometers on vehicles. The major challenge with motor-vehicles compared to

trains is the variation in the path between passes over a particular section of road. Trains, in contrast, always travel on the same part of the tracks, so the position uncertainty is one-dimensional along the length of the tracks. Motor-vehicles would have two-dimensional position uncertainty. Using GPS position alone, the position uncertainty may be too great to make comparing current data to historical data feasible. However, as autonomous and semi-autonomous vehicles become more prevalent, additional position data from, for example, LiDAR sensors, may reduce the position uncertainty sufficiently to make a data-driven road-monitoring approach practical.

This work may have other applications beyond infrastructure monitoring. The core contribution has to do with building reliable models of an object from repeated passes over that object. One application may be to place sensors on the belt of a conveyor-belt system and use the collected data to monitor the condition of the conveyor's mechanical system. Using our data-driven approach, issues such as a failing ball-bearing might be identified as a change relative to the historical behavior at the location of the bearing within the conveyor-belt system. Such a method might be more economical than instrumenting each section of the conveyor system, particularly for larger conveyor systems common in mining operations.

# Bibliography


[1] Google Maps: Ground truth project. URL https://sites.google.com/a/pressatgoogle.com/google-maps-for-iphone/google-maps-metrics. Accessed: 2016-05-26. 1.1

[2] Predix: The industrial internet platform. *GE Platform Brief*, 2016. 1.1

[3] Keiiti Aki and Bernard Chouet. Origin of coda waves: source, attenuation, and scattering effects. *Journal of geophysical research*, 80(23):3322–3342, 1975. 4.4.1

[4] Port Authority. Agency profile. URL http://www.portauthority.org/paac/CompanyInfoProjects/AgencyProfile.aspx. Accessed: 2016-05-26. 1.1

[5] D. Barke and W. K. Chiu. Structural Health Monitoring in the Railway Industry: A Review. *Structural Health Monitoring*, 4(1):81–93, March 2005. ISSN 1475-9217, 1741-3168. doi: 10.1177/1475921705049764. URL http://shm.sagepub.com/content/4/1/81. 1.3

[6] E. G. Berggren, M. X. D. Li, and J. Spnnar. A new approach to the analysis and presentation of vertical track geometry quality and rail roughness. *Wear*, 265(910):1488–1496, October 2008. ISSN 0043-1648. doi: 10.1016/j.wear.2008.01.029. URL http://www.sciencedirect.com/science/article/pii/S0043164808001828. 3.2, 1

[7] M. Bocciolone, A. Caprioli, A. Cigada, and A. Collina. A measurement system for quick rail inspection and effective track maintenance strategy. *Mechanical Systems and Signal Processing*, 21(3):1242–1254, April 2007. ISSN 0888-3270. doi: 10.1016/j.ymssp.2006.

02.007. URL http://www.sciencedirect.com/science/article/pii/S0888327006000434. 1.3, 1.5, 5.1

[8] Daniel Cantero and Biswajit Basu. Railway infrastructure damage detection using wavelet transformed acceleration response of traversing vehicle. *Structural Control and Health Monitoring*, 22(1):62–70, 2015. 1.5

[9] A. Caprioli, A. Cigada, and D. Raveglia. Rail inspection in track maintenance: A benchmark between the wavelet approach and the more conventional Fourier analysis. *Mechanical Systems and Signal Processing*, 21(2):631–652, February 2007. ISSN 0888-3270. doi: 10.1016/j.ymssp.2005.12.001. URL http://www.sciencedirect.com/science/article/pii/S0888327005002402. 1.5, 3.2.1

[10] G.A. Carr, C. Diaz, and J. Bloom. Method and apparatus for track geometry measurement, October 21 2003. URL http://www.google.com/patents/US6634112. US Patent 6,634,112. 1.5

[11] F. Cerda, S. Chen, J. Bielak, J. H. Garrett, P. Rizzo, and J. Kovačević. Indirect structural health monitoring of a simplified laboratory-scale bridge model. *Smart Structures and Systems*, 13(5):859–868, May 2014. 3.2.1

[12] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009. 1.5

[13] Siheng Chen, Fernando Cerda, Piervincenzo Rizzo, Jacobo Bielak, James H Garrett, and Jelena Kovačević. Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring. *IEEE Transactions on Signal Processing*, 62(11):2879–2893, 2014. 1.5, 3.2.1

[14] J Corbin. *Statistical Representations of Track Geometry*. Federal Railroad Administration, Office of Research and Development, 1980. 3.2.1

[15] Roy R Craig and Andrew J Kurdila. *Fundamentals of structural dynamics*. John Wiley &

Sons, 2006. 3.2.1

[16] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977. 4.3

[17] M. Dumitriu. Influence of the Suspension Damping on the Ride Comfort of Passenger Railway Vehicles. *Scientific Bulletin University Plitechnica of Bucharest Series D*, 74(4), 2012. ISSN 1454-2358. 3.2.2

[18] Nour-Eddin El Faouzi, Henry Leung, and Ajeesh Kurian. Data fusion in intelligent transportation systems: Progress and challenges–a survey. *Information Fusion*, 12(1):4–10, 2011. 5.1

[19] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 29–39. ACM, 2008. (document), 5.1, 7.2

[20] C. Esveld. *Modern Railway Track: Digital Edition*. MRT-Productions, 3.4 edition edition, April 2015. 1.3

[21] C Esveld, A Jourdain, G Kaess, and MJ Shenton. Historic data on track geometry in relation to maintenance. *Rail Engineering International, Edition*, (2):p16, 1988. (document), 3.2.1, 3.4

[22] D Evans. The internet of things: How the next evolution of the internet is changing everything. *Cisco Report*, 2011. URL `http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf`. 1.1

[23] Matineh Eybpoosh, Mario Berges, and Haeyoung Noh. Sparse representation of ultrasonic guided-waves for robust damage detection in pipelines under varying environmental and operational conditions. *Structural Control and Health Monitoring*, 23(2):369–391, 2016.

4.1

[24] Matthew Finnegan. Boeing 787s to create half a terabyte of data per flight, says Virgin Atlantic. URL http://www.computerworlduk.com/news/data/boeing-787s-create-half-terabyte-of-data-per-flight-says-virgin-atlanti Accessed: 2016-05-26. 1.1

[25] Dongdong Ge, Xiaoye Jiang, and Yinyu Ye. A note on the complexity of l p minimization. *Mathematical programming*, 129(2):285–299, 2011. 4.3

[26] Economic Development Research Group et al. Failure to act: The economic impact of current investment trends in surface transportation infrastructure. American Society of Civil Engineers, 2011. 1.2

[27] Fredrik Gustafsson and Fredrik Gustafsson. *Adaptive filtering and change detection*, volume 1. Wiley New York, 2000. 3.1, 3.2.3, 3.2.3

[28] David L Hall and James Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997. 5.1, 5.2

[29] Oliver Heirich, Andreas Lehner, Patrick Robertson, and Thomas Strang. Measurement and analysis of train motion and railway track characteristics with inertial sensors. In *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1995–2000. IEEE, 2011. 1.3, 1.5, 4.1

[30] Oliver Heirich, Paul Robertson, and Thomas Strang. Railslam-localization of rail vehicles and mapping of geometric railway tracks. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5212–5219. IEEE, 2013. 4.1

[31] Stefan Hensel, Carsten Hasberg, and Christoph Stiller. Probabilistic rail vehicle localization with eddy current sensors in topological maps. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1525–1536, 2011. 1.5

[32] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal*

*of Basic Engineering*, 82(1):35–45, 1960. 5.2

[33] Tomislav Kos, Ivan Markezic, and Josip Pokrajcic. Effects of multipath reception on gps positioning performance. In *Elmar, 2010 Proceedings*, pages 399–402. IEEE, 2010. 5.2

[34] G Lederman, Z Wang, J Bielak, H Noh, JH Garrett, S Chen, J Kovačević, F Cerda, and P Rizzo. Damage quantification and localization algorithms for indirect shm of bridges. In *7th International Conference on Bridge Maintenance, Safety and Management (IABMAS)*, pages 640–647, 2014. doi: 10.1201/b17063-93. 1.5

[35] G. Lederman, Z. Wang, J. Bielak, H. Y. Noh, J. H. Garrett, S. Chen, J. Kovačević, F. Cerda, and P. Rizzo. Damage quantification and localization algorithms for indirect SHM of bridges. In *IABMAS*. Shanghai, CN, July 2014. 3.2.1

[36] G. Lederman, S. Chen, J. Garrett, J. Kovačević, H. Noh, and J. Bielak. Track-monitoring from the dynamic response of an operational train. *Journal of Mechanical Systems and Signal Processing, Submitted*, 2016. (document), 1.5, 4.4.2, 4.10

[37] Jun Seok Lee, Sunghoon Choi, Sang-Soo Kim, Choonsoo Park, and Young Guk Kim. A mixed filtering approach for track condition monitoring using accelerometers on the axle box and bogie. *IEEE Transactions on Instrumentation and Measurement*, 61(3):749–758, 2012. 1.5

[38] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. 4.3, 4.3

[39] David Mascareñas, Crystal Plont, Christina Brown, Martin Cowell, N Jordan Jameson, Jessica Block, Stephanie Djidjev, Heidi Hahn, and Charles Farrar. A vibro-haptic human–machine interface for structural health monitoring. *Structural Health Monitoring*, 13(6): 671–685, 2014. 7.1

[40] TJ Matarazzo and SN Pakzad. Structural identification using expectation maximization (stride): An iterative output-only method for modal identification. *J. Eng. Mech*, 10:1061,

2015. doi: 10.1061/(ASCE)EM.1943-7889.0000951. 3.2

[41] Raman K Mehra. On the identification of variances and adaptive Kalman filtering. *IEEE Transactions on Automatic Control*, 15(2):175–184, 1970. 5.2

[42] M. Molodova, Z. Li, and R. Dollevoet. Axle box acceleration: Measurement and simulation for detection of short track defects. *Wear*, 271(12):349–356, May 2011. ISSN 0043-1648. doi: 10.1016/j.wear.2010.10.003. URL http://www.sciencedirect.com/science/article/pii/S0043164810003376. 1.3, 1.5, 3.1

[43] M. Molodova, Z. Li, A. Nunez, and R. Dollevoet. Automatic Detection of Squats in Railway Infrastructure. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):1980–1990, October 2014. ISSN 1524-9050. doi: 10.1109/TITS.2014.2307955. 1.5, 5.1

[44] Daniel Moore. Inspectors missed defect that caused fiery oil train derailment in West Virginia. *Pittsburgh Post-Gazette*. URL http://powersource.post-gazette.com/powersource/2015/10/09/Officials-Broken-rail-missed-on-2-inspections-caused-fiery-West-Virgini stories/201510090262. Accessed: 2016-05-26. 1.3

[45] Y. Naganuma, M. Kobayashi, M. Nakagawa, and T. Okumura. Condition monitoring of shinkansen tracks using commercial trains. In *2008 4th IET International Conference on Railway Condition Monitoring*, pages 1–6, June 2008. 1.3, 1.5

[46] J Nielsen, E Berggren, T Lölgen, R Müller, B Stallaert, and L Pesqueux. Overview of methods for measurements of track irregularities important for ground-borne vibration. *Railway Induced Vibration Abatement Solutions (RIVAS)*, 2013. 1.5

[47] H Noh, R Rajagopal, and AS Kiremidjian. Sequential structural damage diagnosis algorithm using a change point detection method. *Journal of Sound and Vibration*, 332(24): 6419–6433, 2013. 1.5, 3.1

[48] Haeyoung Noh and Anne S Kiremidjian. Application of a sparse representation method

using k-svd to data compression of experimental ambient vibration data for shm. In *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*, pages 79814N–79814N. International Society for Optics and Photonics, 2011. 4.1

[49] Eugene J O'Brien, Cathal Bowe, and Paraic Quirke. Determination of vertical alignment of track using accelerometer readings. In *IMechE Stephenson Conference for Railways: Research for Railways, 21-23 April, 2015*, 2015. URL `http://researchrepository.ucd.ie/handle/10197/7000`. 1.5, 4.2

[50] National Academy of Engineering. *Grand Challenges for Engineering*. `http://www.engineeringchallenges.org/`. Accessed: 2014-09-21. 1.2

[51] Federal Railroad Administration Office of Safety Analysis. *FRA Safety Data Site*. `http://safetydata.fra.dot.gov/officeofsafety/default.aspx`. Accessed: 2016-05-22. (document), 1.1, 1.1

[52] Y. Oshima, K. Yamamoto, K. Sugiura, A. Tanaka, and M. Hori. Simultaneous monitoring of the coupled vibration between a bridge and moving trains. In *Proc. 5th IABMAS Conference*, page 186, Philadelphia, USA, July 2010. URL `http://www.crcnetbase.com/doi/abs/10.1201/b10430-114`. 1.3, 1.5

[53] Y. Oshima, K. Yamamoto, and K. Sugiura. Damage assessment of a bridge based on mode shapes estimated by responses of passing vehicles. *Smart Structures and Systems*, Volume 13, Number 5, May 2014. 3.2.1, 7.2

[54] ES Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954. 3.1

[55] J. Real, P. Salvador, L. Montalbn, and M. Bueno. Determination of Rail Vertical Profile through Inertial Methods. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 225(1):14–23, January 2011. ISSN 0954-4097, 2041-3017. doi: 10.1243/09544097JRRT353. URL `http://pif.sagepub.com/content/225/1/14`. 1.3, 1.5, 3.2, 4.2

[56] Samer S Saab. A map matching approach for train positioning part I: development and analysis. *IEEE Transactions on Vehicular Technology*, 49(2):467–475, 2000. 4.1

[57] Seyed Mohsen Shahandashti, Saiedeh N Razavi, Lucio Soibelman, Mario Berges, Carlos H Caldas, Ioannis Brilakis, Jochen Teizer, Patricio A Vela, Carl Haas, James Garrett, et al. Data-fusion approaches and applications for construction engineering. *Journal of Construction Engineering and Management*, 137(10):863–869, 2010. 5.1

[58] Craig Smith. By the numbers: 125+ amazing youtube statistics. URL `http://expandedramblings.com/index.php/youtube-statistics/`. Accessed: 2016-05-26. 1.1

[59] Alan N Steinberg, Christopher L Bowman, and Franklin E White. Revisions to the JDL data fusion model. In *AeroSense'99*, pages 430–441. International Society for Optics and Photonics, 1999. 5.1, 5.2

[60] Hamid A Toliyat, Karim Abbaszadeh, Mina M Rahimian, and Leslie E Olson. Rail defect diagnosis using wavelet packet decomposition. *IEEE Transactions on Industry Applications*, 39(5):1454–1461, 2003. 1.5

[61] William Vega-Brown, Abraham Bachrach, Adam Bry, Jonathan Kelly, and Nicholas Roy. Cello: A fast algorithm for covariance estimation. In *IEEE International Conference Robotics and Automation (ICRA)*, pages 3160–3167. IEEE, 2013. 5.2

[62] Martin Vetterli and Jelena Kovacevic. *Wavelets and subband coding*. Number LCAV-BOOK-1995-001. Prentice-hall, 1995. 3.1

[63] Martin Vetterli, Jelena Kovačević, and Vivek K Goyal. *Foundations of Signal Processing*. Cambridge University Press, 2014. 4.3

[64] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–511. IEEE, 2001. 3.1

[65] Z Wang, S Chen, G Lederman, F Cerda, J Bielak, JH Garrett, P Rizzo, and J Kovačević. Comparison of sparse representation and fourier discriminant methods: Damage location classification in indirect lab-scale bridge structural health monitoring. In *The 2013 Structures Congress*, pages 436–446, 2013. 4.1

[66] Christopher P Ward, PF Weston, EJC Stewart, Hong Li, Roger M Goodall, C Roberts, TX Mei, Guy Charles, and Roger Dixon. Condition monitoring opportunities using vehicle-based sensors. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 225(2):202–218, 2011. 1.5

[67] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995. 5.2, 5.2

[68] PF Weston, CS Ling, C Roberts, CJ Goodman, P Li, and RM Goodall. Monitoring vertical track irregularity from in-service railway vehicles. *Proceedings of the institution of mechanical engineers, Part F: Journal of Rail and Rapid Transit*, 221(1):75–88, 2007. 1.5

[69] PF Weston, CS Ling, C Roberts, CJ Goodman, P Li, and RM Goodall. Monitoring vertical track irregularity from in-service railway vehicles. *Proceedings of the institution of mechanical engineers, Part F: Journal of Rail and Rapid Transit*, 221(1):75–88, 2007. 1.3, 1.5, 5.1

[70] Franklin E White et al. A model for data fusion. In *Proc. 1st National Symposium on Sensor Fusion*, volume 2, pages 149–158, 1988. 5.1

[71] Yongchao Yang and Satish Nagarajaiah. Structural damage identification via a combination of blind feature extraction and sparse representation classification. *Mechanical Systems and Signal Processing*, 45(1):1–23, 2014. 4.1