

Post-Processing Method for Determining Peaks in Noisy Strain Gauge Data with a Low
Sampling Frequency

Peter L. Hill

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in
partial fulfillment of the requirements for the degree of

Master of Science

In

Mechanical Engineering

Alfred Wicks, Chair

Craig Woolsey

Ricardo Burdisso

May 9, 2017

Blacksburg, VA

Keywords: Strain Gauge, MATLAB, Digital Signal Processing

Post-Processing Method for Determining Peaks in Noisy Strain Gauge Data with a Low Sampling Frequency

Peter L. Hill

Abstract

The Virginia Tech Transportation Institute is recognized for being a pioneer in naturalistic driving studies. These studies determine driving behavior, and its correlation to safety critical events, by equipping participant's vehicles with data acquisition systems and recording them for a period of time. The driver's habits and responses to certain scenarios and events are analyzed to determine trends and opportunities to improve overall driver safety.

One of these studies installed strain gauges on the front and rear brake levers of motorcycles to record the frequency and magnitude of brake presses. The recorded data was sampled at 10 hertz and had a significant amount of noise introduced from temperature and electromagnetic interference. This thesis proposes a peak detection algorithm, written in MATLAB, that can parallel process the 40,000 trips recorded in this naturalistic driving study. This algorithm uses an iterative LOWESS regression to eliminate the offset from zero when the strain gauge is not stressed, as well as a cumulative sum and statistical concepts to separate brake activations from the rest of the noisy signal.

This algorithm was verified by comparing its brake activation detection to brake activations that were manually identified through video reduction. The algorithm had difficulty in accurately identifying activations in files where the amplitude of the noise was close to the amplitude of the brake activations, but this only described 2% of the sampled data. For the rest of the files, the peak detection algorithm had an accuracy of over 90%.

Post-Processing Method for Determining Peaks in Noisy Strain Gauge Data with a Low Sampling Frequency

Peter L. Hill

General Audience Abstract

The Virginia Tech Transportation Institute is recognized for being a pioneer in naturalistic driving studies. In these studies, participants are recorded with cameras and other sensors for a period of time. Researchers then look at this data and find the habits that tended to distract the drivers, like using their phones while driving, and other characteristics that bring insight on what causes crashes and unsafe driving behavior.

One of these studies installed strain gauges on the front and rear brake levers of motorcycles to record how hard the brakes were pressed, as well as how often. The strain gauge was sampled ten times a second, and had a significant amount of variation in the signal from temperature changes and interference from other electronic systems on the bike. This thesis proposes a method, written in MATLAB, that can quickly find all the brake activations in the 40,000 trips recorded in this naturalistic driving study. This program uses an iterative LOWESS regression, cumulative sum, and other statistical concepts to determine the brake activations in the signal.

This program was verified by comparing its brake activation it found to brake activations that were manually identified through video reduction. The algorithm had difficulty in accurately identifying activations in files where the peaks of the noise were close to the peaks of the brake activations, but this only described 2% of the sampled data. For the rest of the files, the program had an accuracy of over 90%.

Acknowledgements

I would like to thank the Virginia Tech Transportation Institute for allowing me to use the naturalistic driving data analyzed for this thesis, specifically Shane McLaughlin and the rest of the Motorcycle Research Group. I would also like to thank Dr. Alfred Wicks in the Mechanical Engineering department at Virginia Tech for taking on the role of my advisor, providing invaluable insight and feedback to this research, and to my overall development as a grad student. My Masters would not be complete without also having Dr. Burdisso and Dr. Woolsey as a part of my committee, and I want to thank them for their feedback and support in helping me obtain this degree.

Most importantly, I would like to thank my wife, Astleigh. This thesis would not have been finished without her unwavering support, her patience through long nights of staying up late to study and putting other things on hold, and her faith in me to “get it done”. She will forever be my inspiration.

Table of Contents

Abstract.....	ii
General Audience Abstract.....	iii
Acknowledgements.....	iv
List of Figures.....	vii
Acronyms and Abbreviations.....	ix
Chapter 1. Thesis Problem Definition.....	1
1.0 Chapter Introduction.....	1
1.1 Introduction to Naturalistic Driving Data.....	1
1.2 Motorcycle Naturalistic Driving Studies.....	2
1.3 Strain Gauge Installation.....	3
1.4 Thesis Problem Definition.....	4
Chapter 2. Literature Review.....	6
2.0 Chapter Introduction.....	6
2.1 Strain Gauge Installation.....	6
2.2 Software Data Filtering Techniques.....	7
2.3 Peak Detection.....	10
2.4 Chapter Conclusion.....	11
Chapter 3. Peak Detection Algorithm Implementation.....	13
3.0 Chapter Introduction.....	13
3.1 Null Offset.....	13
3.2 Detrended Data Validation.....	17
3.3 Peak Detection.....	20
3.4 Brake Activation Determination.....	22
3.5 Execution of Algorithm in MATLAB.....	25
3.6 Chapter Conclusion.....	26
Chapter 4. Discussion of Method.....	27
4.0 Chapter Introduction.....	27
4.1 Peak Detection.....	27
4.2 Peak Bounds Accuracy.....	33
4.3 Magnitude Use.....	34
4.4 Chapter Conclusion.....	35
Chapter 5. Conclusions.....	36

Chapter 6. Summary	37
Bibliography	39
Appendix.....	41
Appendix A. MATLAB Code: Main.m	41
Appendix B. MATLAB Code: Null_Offset.m.....	48
Appendix C. MATLAB Code: Fill_Gaps.m.....	52
Appendix D. MATLAB Code: Find_CuSum_Groups.m	55
Appendix E. MATLAB Code: Point_Diff.m.....	60

List of Figures

Figure 1. (Top) Strain gauges glued to brake levers, with a rear brake lever on the left and front brake lever on the right. (Bottom) Close-up of the strain gauge surface that is adhered to a brake lever.	4
Figure 2. Wheatstone bridge where R_1 , R_2 , and R_3 are resistors with a known value and R_4 is a strain gauge [12].	6
Figure 3. LOWESS regression as it would be performed on data at different points in the signal. The y-axis is the regression weight and the x-axis is the index of the point [22].....	10
Figure 4. Data sampled at 10Hz from a strain gauge installed on the front brake of motorcycle.....	14
Figure 5. First iteration of smoothing the data within the data range with a LOWESS filter.....	16
Figure 6. Fourth iteration of computing the data range and using LOWESS smoothing to find the null offset.	17
Figure 7. Detrended Data with positive activations. The null offset is equal to zero at every point of the line.	18
Figure 8. Data from a trip where the strain gauge may have fallen off of the brake, or otherwise is unusable.	19
Figure 9. Detrended data from Figure 8. The large peak at 1,200,000 milliseconds is not a brake activation but noise in the signal.....	20
Figure 10. Cumulative sum of the detrended data, which sums every point up to the current timestamp.	21
Figure 11. Cumulative distribution function of the positive peaks.	23
Figure 12. The amplitude gaps between points from Figure 11, calculated using Eq. 13. The red circle is around the point of the lowest amplitude considered a brake activation.	24
Figure 13. Original data with brake activations highlighted in green.	25

Figure 14. Brake activations highlighted for a file where the peak amplitude of the noise was relatively small compared to the brake activations, there was a slight null offset shift, noise frequency seemingly stayed the same, and brake activations were negative. 29

Figure 15. Brake activations highlighted for a file where the null offset changes over time, the noise peak amplitudes are relatively large and noise frequency changes over time. 30

Figure 16. Brake Activations highlighted in green for a file where there is a mix of large amplitude and smaller, yet frequent, amplitude brake activations. Brake activations that were not found by the algorithm are circled in red..... 31

Figure 17. Brake activations highlighted for a file that has a relatively small noise amplitude, frequent brake presses, and little null offset shift over time..... 32

Figure 18. Brake activations highlighted for a file that has a relatively small noise amplitude, little null offset shift over time with more infrequent and longer brake presses. 32

Figure 19. Activations are separated by the median, rather than the null offset. 33

Figure 20. Example of acceleration and speed data during a brake activation. Both brakes were used, with the red box outlining the activation. [26] 35

Acronyms and Abbreviations

CAN – Controller Area Network

CDF – Cumulative Distribution Function

DAS – Data Acquisition System

DSP – Digital Signal Processing

FFT – Fast Fourier Transform

NDS – Naturalistic Driving Study

PDF – Probability Distribution Function

SCE – Safety Critical Event

VTTI – Virginia Tech Transportation Institute

Chapter 1. Thesis Problem Definition

1.0 Chapter Introduction

The following chapter presents an overview of naturalistic driving research at the Virginia Tech Transportation Institute (VTTI), the use of strain gauges to measure the magnitude of brake pulls on a motorcycle, and the problems associated with analyzing this data.

1.1 Introduction to Naturalistic Driving Data

For over 15 years, the Virginia Tech Transportation Institute (VTTI) has conducted a variety of Naturalistic Driving Studies (NDS) [1]. These studies are meant to provide a picture of normal driving behavior for everyday people, accomplished by equipping participant's vehicles with data acquisition systems (DAS) and recording them as they drive. Any trip the participant takes in his or her vehicle, whether a typical trip to work or infrequent drive to the doctor's office, the DAS records various data elements such as video, speed and GPS location. These are recorded from sensors in the DAS, and from the controller area network (CAN) of the vehicle. VTTI can then use this data to determine factors that contributed to a safety critical event (SCE), such as a crash or near-crash event, or to get a picture of how people drive normally. This data can help improve automated and safety systems in vehicles to better serve the drivers.

For example, in one study researchers used the recorded video from an NDS to analyze the amount of time that drivers took their eyes off the road to perform certain cell phone related tasks (typing a text message, making a phone call, looking for the phone, etc.), and they determined which tasks posed the biggest threat of causing a crash [2]. Another study analyzed the risk of drowsy drivers to be involved in a crash or near-crash SCE as compared to normal drivers, and found that driving while drowsy has a 4 to 6 time higher risk of being involved in a crash or near crash event than those who are alert [3]. These are just two examples that help lawmakers determine what distractions should be regulated, and gives insight on future safety systems that could add the biggest safety benefits.

1.2 Motorcycle Naturalistic Driving Studies

VTTI has completed many Naturalistic Driving Studies with cars [4] [5] and heavy vehicles [6] [7], as well as with motorcycles [8]. Motorcycles are equipped with the same DAS that are used for cars and trucks, although they don't have the extra room like a trunk or a cabin to store the computer and recording systems. Instead, custom mounts are designed to hold the DAS for each model motorcycle, and installed on the frame of the bike. Most motorcycles also don't have a CAN network to record data from, so other sensors must be installed to gather information like speed and brake use. These sensors must be weather-proofed and securely installed to the bike so that months of use won't cause the sensors to fall off or malfunction.

One variable of particular interest for the motorcycle studies is the brake use of a driver, including both when the brakes were used and the strength of each brake press. Brake use is an important variable, as in a previous car study to categorize safe vs. unsafe driver's braking habits it was found that a driver's braking behavior was directly correlated to the number of SCEs encountered by that driver [9]. Not only can it describe driving behavior, but it can provide insight on whether braking power or reaction time were factors when a participant was involved in a crash or near-crash event. If a trend can be established between a motorcyclist's braking behavior and SCEs, this could lead to implementing better training at motorcycle institutions, or providing guidelines for safety systems. Regardless of how the results are used, it is valuable information in reducing the amount and severity of SCEs encountered by motorcyclists.

In order for trends to be established between braking instances and SCEs, the DAS must be able to record every instance of braking from a motorcycle, as well as its braking force. Since there is no CAN for the DAS to record this braking data from, single strain gauges were installed on the front and rear brake levers. These strain gauges measure the deformation of the brake lever, which can be correlated to the amount of braking force applied.

1.3 Strain Gauge Installation

Strain gauges are very useful for finding the strain put on a shaft, and have been used in studies ranging from the deformation in human bones [10] to structural health monitoring in bridges [11]. The strain gauges installed on the front and rear brake levers of the motorcycles were of the typical bonded metallic type, meaning they used a very fine metallic foil grid to measure the force applied to it when the brake lever was pressed. Each brake press puts the strain gauge into tension or compression, changing the resistance of the foil grid as the wire changes in length and thickness. These changes in resistance are very small, so it is critical that the strain gauges are adhered to the brake levers correctly as any imperfections or abnormalities in the setup will affect the measurement.

The shafts of the front and rear brakes were sanded in increasing fineness with sandpaper, and cleaned with solvents. Care was taken to ensure fingerprints were not left on the gauge by using scotch tape to maneuver and press the strain gauge into place, and then the tape was removed. The gauges were soldered to terminals that had a Wheatstone bridge, with care being taken so that the soldered connections did not produce any strain on the gauges when at rest. After everything was set into place, an epoxy was poured over the entire terminal. Figure 1 provides a visualization of the strain gauge setup on the brake levers.



Figure 1. (Top) Strain gauges glued to brake levers, with a rear brake lever on the left and front brake lever on the right. (Bottom) Close-up of the strain gauge surface that is adhered to a brake lever.

1.4 Thesis Problem Definition

In one NDS, over 100 motorcycles were equipped with DASs and strain gauges. They were recorded for about a year each, with the DAS logging the data from the sensors for every trip a participant took on the motorcycle. When the data was returned to VTTI and imported into the database, the following issues became apparent with the data recorded from the strain gauges:

- When the strain gauge was neither in compression or tension, the value was not zero. It was instead offset from zero by a large value that changed over time.
- There was a lot of high frequency noise.
- Some files had data that was unusable, due to the strain gauge detaching from the brake lever, or otherwise failing,

- Differences between brake levers, installation technique and other factors made it difficult to compare the magnitude of brake presses between bikes. Also, some bikes measured strain in tension and others in compression, which changed the sign of recorded brake activations.
- The units for the data that was recorded from the strain gauge was not documented, so it was unclear what exactly the data was measured in (millivolts, resistance, strain, etc.).

With over 40,000 trips and 365,000 miles of recorded data, it was not feasible to visually inspect and manually reduce every trip. Instead, an algorithm needed to be developed that could address these issues, accurately identify the brake presses, and determine the start and end time of each brake press. Finally, the brake activations needed to be quantified so that an analysis could be completed on the strength of brake presses and their correlation to SCEs. This thesis outlines a method to accomplish these tasks.

Chapter 2. Literature Review

2.0 Chapter Introduction

Motorcycles in a naturalistic driving study (NDS) at the Virginia Tech Transportation Institute (VTTI) had strain gauges installed on the front and rear brakes to measure the magnitude of brake presses. This chapter explores the applicable literature available on analyzing and filtering data.

2.1 Strain Gauge Installation

Essentially, a strain gauge is just a resistor that changes value when a force is applied to it. These minute changes in resistance are typically caused by an axial or bending force that changes the length and thickness of the metallic foil grid in the gauge. A Wheatstone bridge is commonly used to measure the change in resistance of a strain gauge, which has a wiring diagram as shown in Figure 2 [12]. The strain gauge is oriented with three other resistors that have a known, stable value, and an excitation voltage V_{EX} is applied to them.

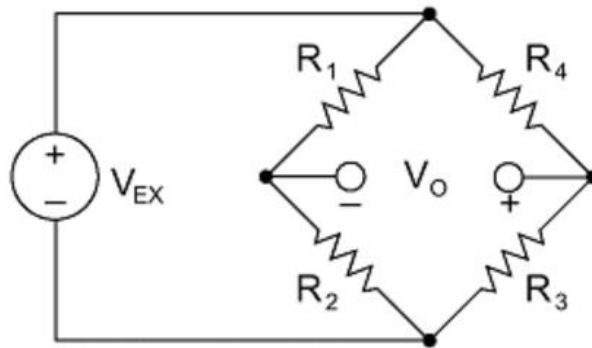


Figure 2. Wheatstone bridge where R_1 , R_2 , and R_3 are resistors with a known value and R_4 is a strain gauge [12].

The output voltage V_0 measured across the resistors is equivalent to

$$V_0 = \left[\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right] * V_{EX} \quad \text{Eq. 1}$$

where if $R_2/R_1 = R_3/R_4$, then V_0 is zero. Only one of the resistors is a strain gauge, called a quarter-bridge, so any change in V_0 should mean that there was a force placed on the strain gauge, and that changed its resistance. However, other factors like temperature and external electromagnetic fields can add noise into the signal and skew the data.

It is important to filter out as much of the noise in real-time when reading data from a strain gauge, before it is recorded by a data acquisition system (DAS). Temperature affects the data because it will cause the material the strain gauge is attached to, or in the strain gauge itself, to expand and contract. This effect can be mitigated by installing another strain gauge in a transverse direction of the original [13]. Another method is to attach a second strain gauge on another spot of identical material to where the first is attached, where it can be subjected to the same temperatures but never strained [14]. The second strain gauge in these cases acts as a baseline that can be compared against the other gauge to remove the effect of temperature. Other electrical signals and electromagnetic fields that the strain gauge is exposed to in its environment can be eliminated by using shielded wires, amplifiers, and other filters [15].

However, even with the best hardware filtering techniques it is inevitable that noise will be introduced into the signal. This noise then needs to be addressed through software.

2.2 Software Data Filtering Techniques

There are many software methods for identifying and eliminating unwanted noise from digital signals. A common practice in digital signal processing (DSP) is using a Fast Fourier Transform (FFT) to break down the signal into the multitude of periodic signals that compose it. The analyzed signal must be both discrete, meaning there are specific points in the signal that are measured, and periodic, meaning that the signal repeats itself [16]. One study used the FFT to filter the signal from strain gauges installed on bridges to analyze bridge loading over time [17]. The FFT transposed the signal into the frequency domain to find the frequencies of the unwanted noise, and then had a pass-band filter applied to it to eliminate these unwanted signals. The data is transposed back to the time domain, free of the noise signals that were in the original data. Another study used the FFT to eliminate electro-magnetic interference from

power equipment that was corrupting data recorded from a vibrating wire strain gauge [18]. The FFT made it possible to identify the periodic signal recorded from the vibrating wire strain gauge without the intrusion of the electro-magnetic interference.

In order for the FFT to accurately break down the recorded signal into the individual signals that compose it, the sampling frequency f_s must be

$$f_s \geq 2f_c \quad \text{Eq. 2}$$

where f_c is the highest recorded frequency in the signal [19]. This is to prevent aliasing, which is when a higher frequency signal takes on the characteristics of a lower frequency signal. If a signal is not sampled at a high enough rate, the higher frequency signal can appear to fit a lower sampling frequency. This can make it difficult to filter out as it is not correctly identified. The highest frequency signal that can be correctly identified at a certain sampling rate is called the Nyquist frequency f_N [20], defined as

$$f_N = \frac{1}{2}f_s \quad \text{Eq. 3}$$

The motorcycle strain gauge data was sampled at 10 hertz, which according to Eq. 3 means the Nyquist frequency would be at 5 hertz. With such a low sampling rate it is extremely likely that aliasing is occurring in the signal. However, the brake activations in the strain gauge data of the motorcycle naturalistic driving study (NDS) are low frequency and have a large amplitude. Since the higher frequency content isn't needed, a low pass filter can eliminate the higher frequency signals that don't need to be there. This is the first part described by a study that compared peak detection methods and decomposed the peak detection procedure into three parts: smoothing, baseline correction and peak finding [21].

A basic low pass filter that will smooth the data to eliminate high frequency signals is an averaging filter [16], defined as

$$y(i) = \frac{1}{M} \sum_{j=0}^{M-1} x(i+j) \quad \text{Eq. 4}$$

where y is the new data signal, M is the total number of points in the window, x is the original signal and i is the index of the current point. An averaging filter iterates through every point in a data sample and averages M points around the i^{th} point to create a new datum, which replaces the original at index i . Eq. 4 only averages the data after the current point being evaluated. To have that point in the center of the window, the equation can be rewritten as

$$y(i) = \frac{1}{M+1} \sum_{j=-M/2}^{M/2} x(i+j) \quad \text{Eq. 5}$$

Smoothing the data eliminates the high frequency content, and is needed in order to find the value that should equal zero when the strain gauge is not being stressed, called the null offset [12] [21]. A similar method to the averaging filter is the LOWESS regression model. The regression model works like the averaging filter by iterating through each point in the data stream and replacing it with a new, smoothed value, but is more robust against outliers. The outliers in the motorcycle NDS are mainly brake activations, which need to be eliminated as much as possible when smoothing the data so that the null offset can be found.

For the LOWESS regression model, regression weights w_i are determined for every point around the point being evaluated [22],

$$w_i = \left(1 - \left|\frac{x-x_i}{d(x)}\right|^3\right)^3 \quad \text{Eq. 6}$$

where x is the original value at that point, x_i is another value in the window, and $d(x)$ is the distance from x to the furthest value in the window. The point with the largest weight is the current point being smoothed, with the farthest point from that having the least weight. Points outside of the window have zero weight. The window centers around the point that is being evaluated, unless that point is in the beginning or the end of the data. In that case, the same number of points will still be used in the weighting calculation, but will extend out farther to the right or left, depending on whether the point is towards the beginning or the end. An example of this is show in Figure 3. After the weights are determined, a

weighted linear least-squares regression using a first degree polynomial is performed, giving a smoothed value. The smoothed data can be removed from the signal to get the value of the null offset, which will provide the baseline to find the peaks from.

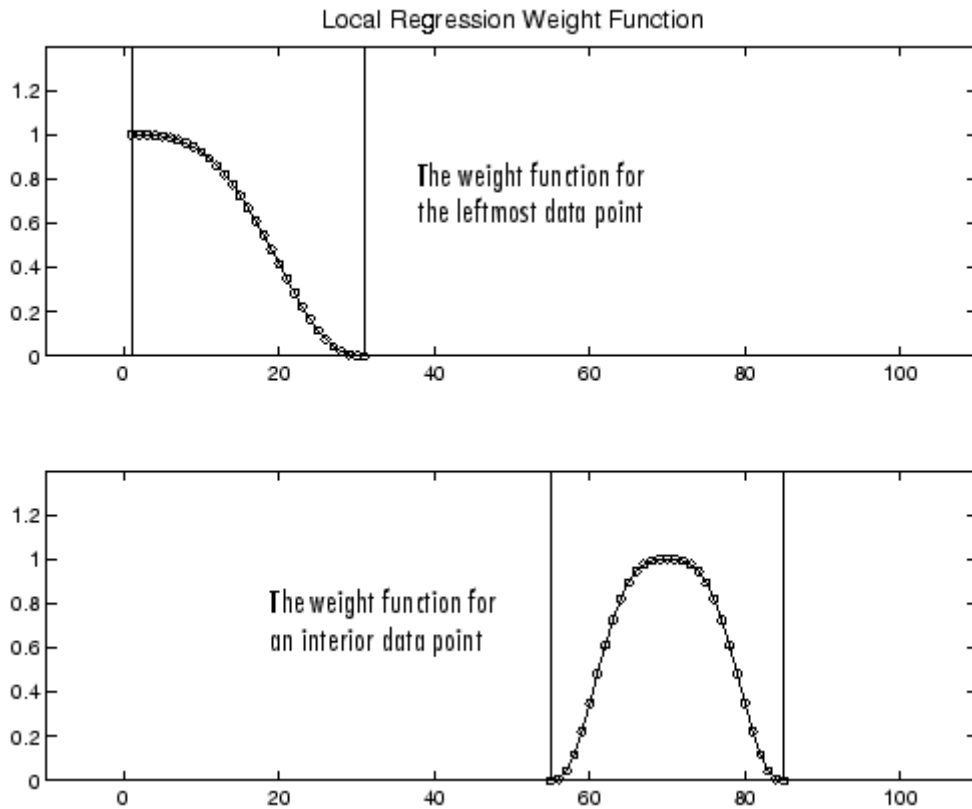


Figure 3. LOWESS regression as it would be performed on data at different points in the signal. The y-axis is the regression weight and the x-axis is the index of the point [22].

2.3 Peak Detection

Peak detection is the most important part of processing the strain gauge signal, as this will provide the brake activation start/end times and amplitudes. One peak detection study performed an undecimated discrete wavelet transform to eliminate noise in a signal, after which an algorithm was set up to find peaks [23]. This peak detection algorithm first used the differences between successive time points to find local maxima and minima, and then found the median absolute value of these differences.

Anything less than this median was defined as noise, and all maxima that were less than the median absolute value were not considered peaks. The maxima that were close together were then combined while retaining the highest peak. The bounds of the combined maxima were the closest local minima. Finally the slopes of these maxima peaks were found, and were considered peaks of interest if they had a value greater than half of the value of the noise [24].

The brake activations in the motorcycle NDS can be considered deviations from the normal signal. The cumulative sum is a method for monitoring mean deviations from a process mean, described by the equation

$$C_i = \sum_{j=1}^i (\bar{x}_j - \mu_0) \quad \text{Eq. 7}$$

where C_i is the cumulative sum up to and including the i^{th} sample, \bar{x}_j is the average of the j^{th} sample, and μ_0 is the target for the process mean [25]. This system is useful in monitoring production processes, as it can quickly determine points that deviate from the normal sample. This equation can be modified so that instead of finding the difference in means, it will sum the difference of each point in the strain gauge data stream from the null offset NO value at that same point

$$C_i = \sum_{j=1}^i (x_j - NO) \quad \text{Eq. 8}$$

This will provide data that increases whenever the data stream is over the null offset, and decreases as the stream goes below the null offset. Large jumps in the cumulative sum can be identified as brake activations, whereas smaller increases are just noise.

2.4 Chapter Conclusion

Data should be recorded from strain gauges through a Wheatstone bridge, and wires should be shielded and steps taken to eliminate as much noise as possible from temperature and other signals. The motorcycle NDS data was already recorded, so these hardware filtering techniques could not be applied. The data also was not sampled at a high enough frequency to break down the recorded noise signals with an FFT or other typical DSP methods. Instead, the data can be reduced by smoothing it to find the null

offset, and then when the brake activation peaks are determined they can be compared to the null offset to determine calibrated brake amplitudes.

Chapter 3. Peak Detection Algorithm Implementation

3.0 Chapter Introduction

This chapter gives a detailed description of the method used to find brake activations in the strain gauge data from the motorcycle naturalistic driving study (NDS), for both the front and rear brakes. Refer to the Appendix for the MATLAB code that implements this method.

3.1 Null Offset

The strain gauge data stream recorded at 10Hz has a positive or negative peak whenever the front or rear brake is applied, depending on whether the strain gauge attached to it was put in compression or tension. The strength of the brake press is directly correlated to the amplitude of the peak in the data stream. The amplitude of this activation should be measured from the value at which the strain gauge was neither in tension or compression. For a detrended signal, the amplitude of the signal when the strain gauge is at rest is zero. However, the strain gauge data collected from the motorcycles has an offset from zero when the strain gauge is neither in tension or compression, called the null offset. This value is not constant between bike trips, nor even within the same trip, due to thermal expansion of the gauge with temperature change and other noise. It is crucial to determine the null offset, though, because without it accurate start/end times of activations and peak amplitude values cannot be determined.

Figure 4 shows a plot of the data recorded by the DAS from a strain gauge installed on the front brake for a single trip. This particular strain gauge was installed so that when the brake lever was pulled, it was put in compression. This causes those activations to be shown as sharp, negative peaks in the data stream. Any time that the strain gauge is not being compressed, the data oscillates around the null offset. The signal oscillates at a high frequency because of external signals that were recorded, and finding the center of this oscillation using a LOWESS regression model will produce a line that follows the null offset.

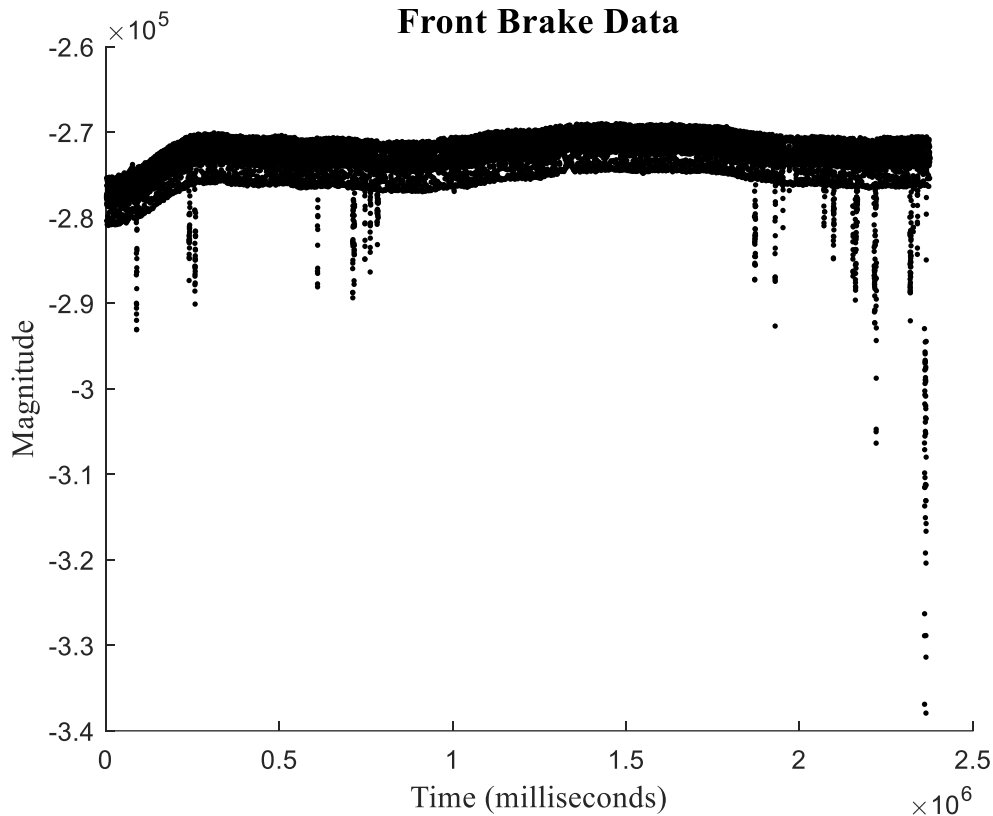


Figure 4. Data sampled at 10Hz from a strain gauge installed on the front brake of motorcycle.

LOWESS regression works well in finding the null offset because it eliminates high frequency signals using a weighted average. The weighted average is resistive to outliers, meaning the sharp brake activation peaks don't pull the null offset from its true value as much as an averaging filter would. The regression needs an established window to average over. By experimenting with different window sizes, 4000 points was determined to be an appropriate window for this algorithm. At 10Hz, 4000 points is a little over three minutes of data that is averaged with weights to create one new point. Because of this, only files that were at least four minutes long were analyzed, as anything below this trip length did not work properly with the regression and window size.

It is important for the null offset to be an accurate interpolation of the signal when the strain gauge was neither in tension or compression, so the data from when the brake was activated needs to have as little effect as possible on the null offset calculation. The activations have not been determined yet, so they can't be eliminated from the signal. In order to make LOWESS regression more robust against the

brake activation outliers, it is used in combination with a defined data range, while also iteratively smoothing the data. For each iteration, a data range is established, called the offset window, and the LOWESS regression is calculated only on the data that falls within that range. This is repeated three more times, for a total of four iterations.

Figure 5 shows the original strain gauge data from Figure 4 with a defined offset window, highlighted in red, and a null offset line calculated from that offset window. The offset window is defined by

$$OW_i = NO_i \pm SD \quad \text{Eq. 9}$$

where NO is the previously calculated null offset, SD is the standard deviation of the original data, and i is the index of the current point being evaluated. If the datum at i has a amplitude that is within the offset window, the value is kept to be used in the LOWESS regression. If it falls outside the window, the amplitude of the previous point is used instead. For the first iteration, the null offset is estimated to be the median of the data, so every point in the null offset matrix has the same value. After each iteration, the null offset with begin to change at each timestamp. The green line in Figure 5 is the calculated null offset from the offset window. The null offset is calculated using the smooth function in MATLAB, spanning 2000 points, and specifying the LOWESS method.

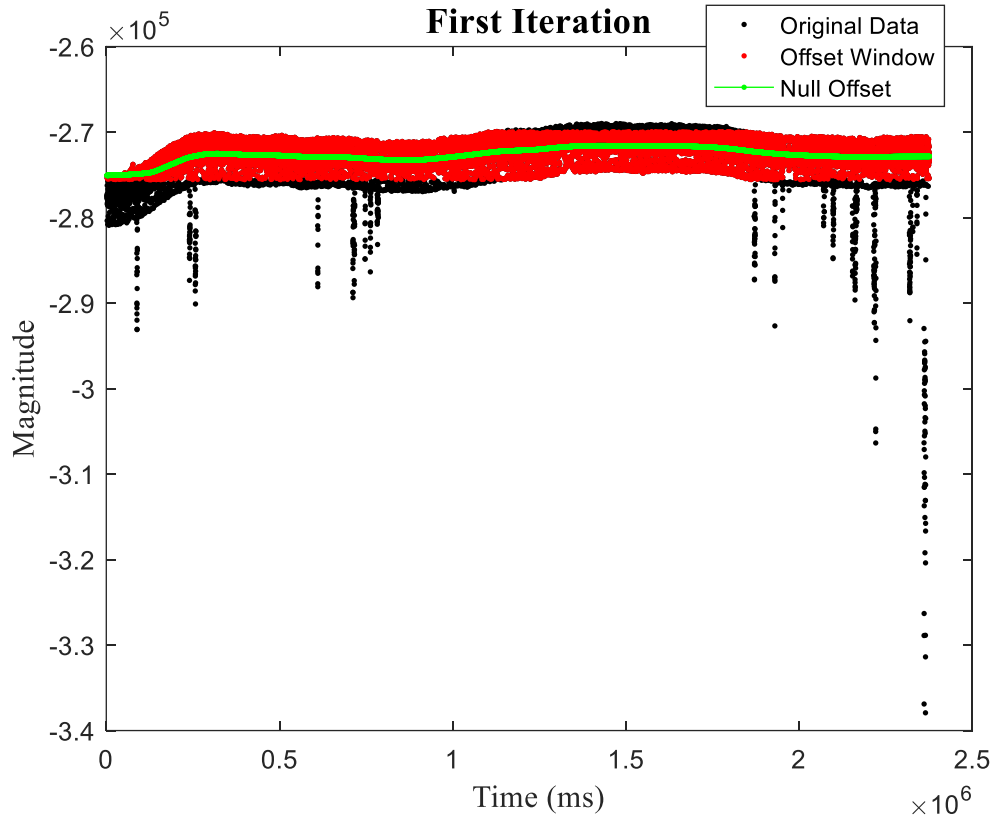


Figure 5. First iteration of smoothing the data within the data range with a LOWESS filter.

The offset window in Figure 5 clearly doesn't cover all of the oscillations around the desired null offset, which results in the calculated null offset being inaccurate in those areas the window doesn't cover. By repeating the process, the calculated null offset gets closer and closer to its true value. For each iteration, the offset window is calculated from Eq. 9, where the value of NO_i is the value of the calculated null offset from the previous iteration at that index. Figure 6 plots the fourth and final iteration, where the null offset very closely follows the center of the noisy data when the strain gauge is at rest.

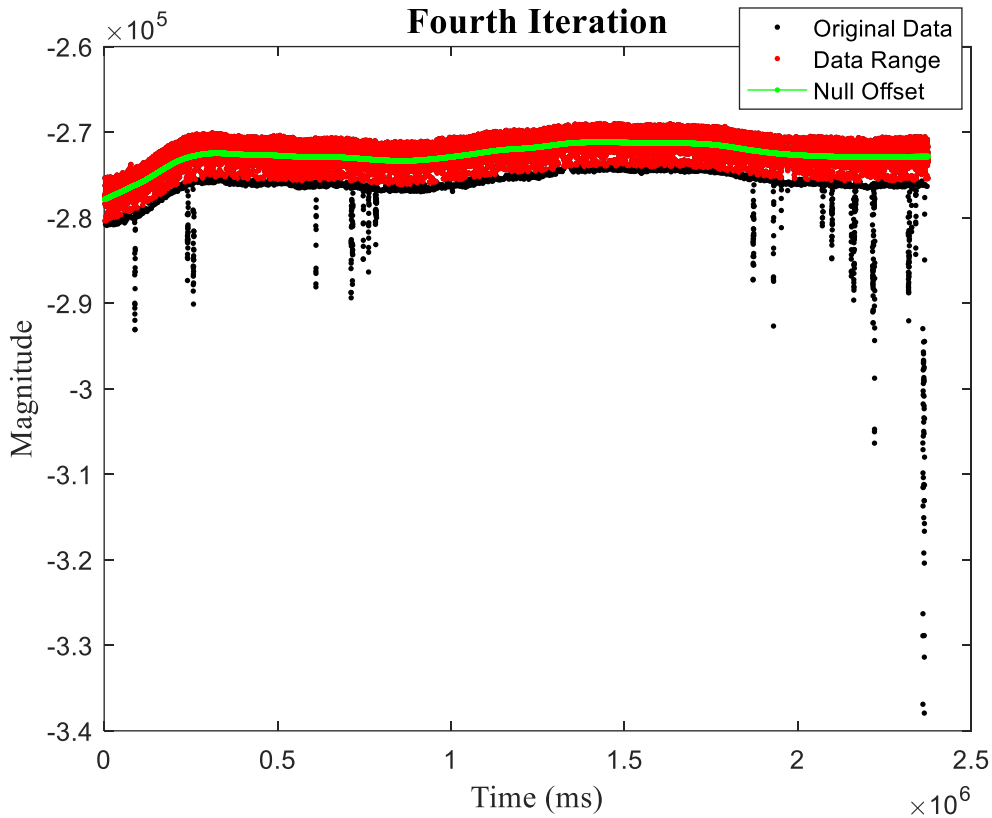


Figure 6. Fourth iteration of computing the data range and using LOWESS smoothing to find the null offset.

3.2 Detrended Data Validation

The null offset is the point that should be zero, where the strain gauge is neither in compression or in tension. Subtracting the null offset from the original data snaps the data to the x-axis, eliminating the null offset and its drift over time. This makes it possible to determine the activation direction, whether positive or negative, by simply comparing which direction has a larger amplitude. Figure 7 shows the original data from Figure 4 detrended and flipped around the x-axis so that the activations are positive. Making all activations positive is important when identifying where they are in the next sections.

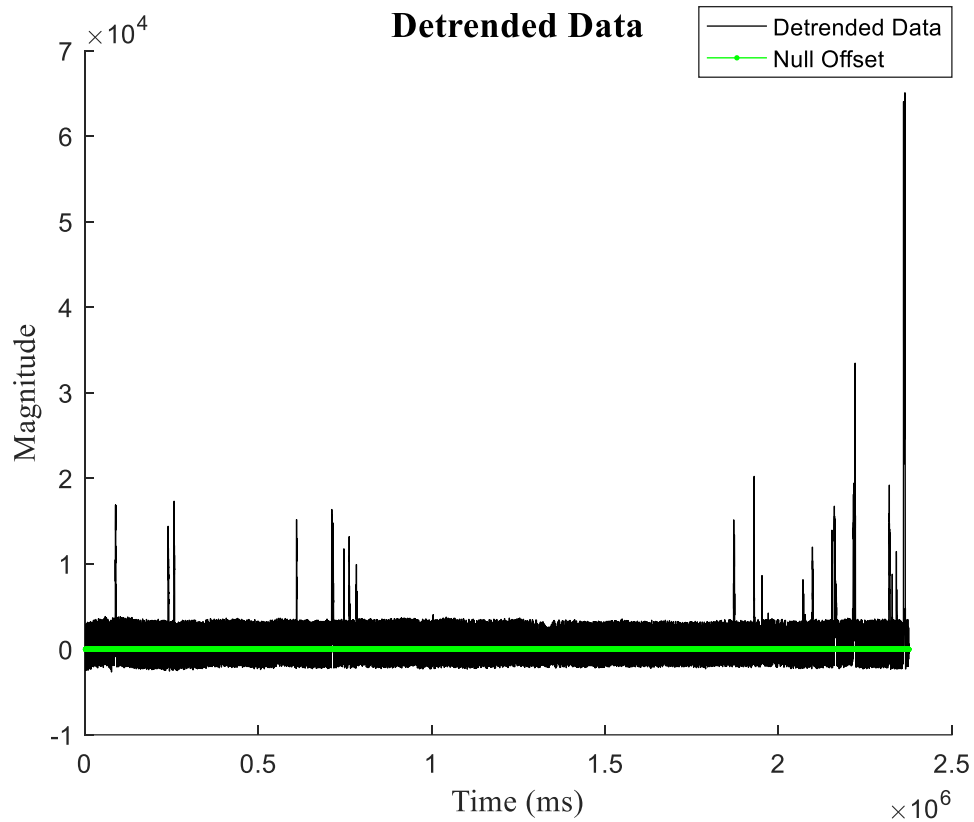


Figure 7. Detrended Data with positive activations. The null offset is equal to zero at every point of the line.

An issue with the strain gauges in the motorcycle NDS is that sometimes they would fail over time, fall off the motorcycle, or otherwise become so noisy that the data wasn't usable because brake activations were not recorded. It is important to identify these files as well, as they can give the appearance of having brake activations when detrended, which will result in false activations being identified and skewing the result set. Figure 8 is an example of data from a strain gauge that is was not recorded properly. There are no clear brake activations, and the null offset drifts drastically in both the positive and negative directions.

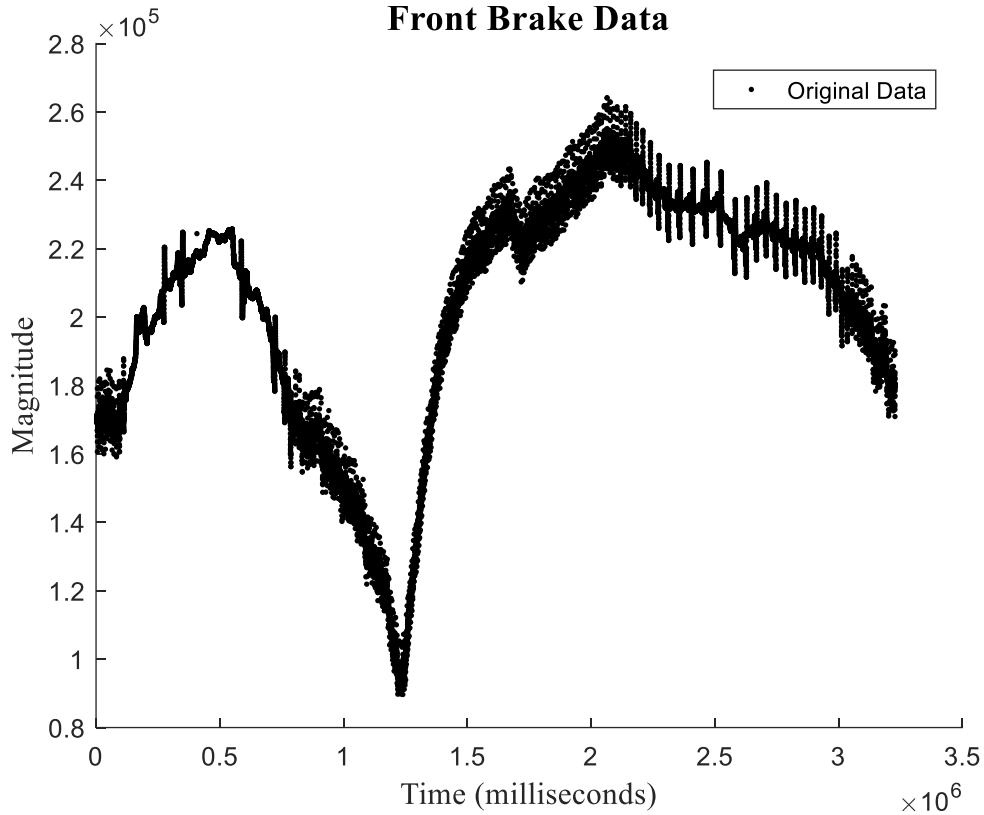


Figure 8. Data from a trip where the strain gauge may have fallen off of the brake, or otherwise is unusable.

To be able to identify these files in the algorithm, the null offset is calculated through iterative LOWESS regression, plotted in Figure 9. It appears that there is a brake activation in the detrended data at about 1,200,000 milliseconds, but really that is due to the huge drift of the null offset over time. This drift can be used to identify the bad files, as the relative variability of the bad signal is significantly greater than a good one. This can be characterized as a ratio r , defined as

$$r = \frac{SDD}{SBD} \quad \text{Eq. 10}$$

where SDD is the standard deviation of the Detrended Data, and SBD is the standard deviation of the bad data. This ratio was completed for a group of known good files and bad files, and it was discovered that the bad files had a ratio less than 0.5, and the good files were greater than 0.5. This check is done for every file, and if the file does not have valid data then it is not analyzed any further.

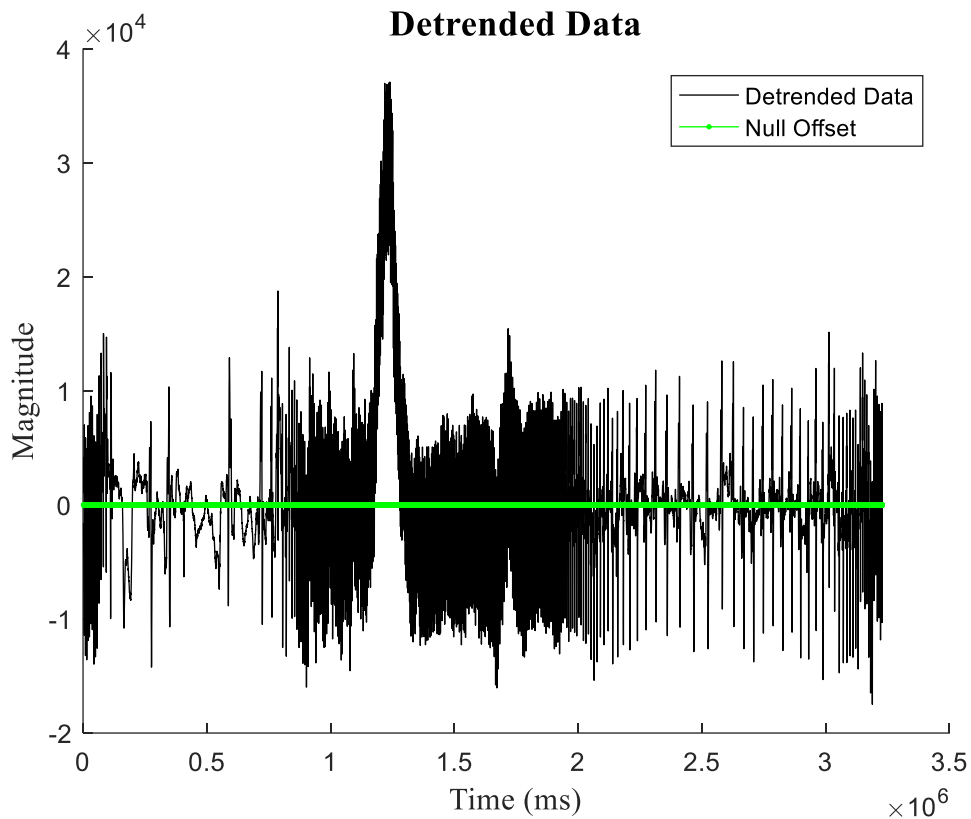


Figure 9. Detrended data from Figure 8. The large peak at 1,200,000 milliseconds is not a brake activation but noise in the signal.

3.3 Peak Detection

At this point when processing a file, the data has been determined to be valid, the null offset has been established, and the detrended data has been produced. The noisy strain gauge signal now oscillates around zero without a trend and with occasional peaks of larger amplitude, which are the brake presses. While these brake presses are relatively simple to pick out by eye, an algorithm has to be ‘taught to see’ these brake activations. This process begins by finding all the peaks in the data, positive or negative, including those from the noise oscillating around zero. In order to be able to determine which peaks in the data are actual brake presses, the algorithm will need something to compare this against, so all peaks must be found with their max amplitude recorded.

A cumulative sum will identify when the signal goes from positive to negative, and is defined by

$$C_i = \sum_{j=1}^i (x_j - \mu) \quad \text{Eq. 11}$$

where C_i is the cumulative summation from 1 to index i , x_j is the amplitude at index j , and μ is the mean.

For this application, the mean of the data is the null offset, which has already been subtracted from the data to create the detrended dataset. This means that Eq. 11 can be rewritten as

$$C_i = \sum_{j=1}^i (d_j) \quad \text{Eq. 12}$$

where d_j is the amplitude of a detrended data point at index j . Figure 10 is a plot of the cumulative sum of the detrended data from Figure 7.

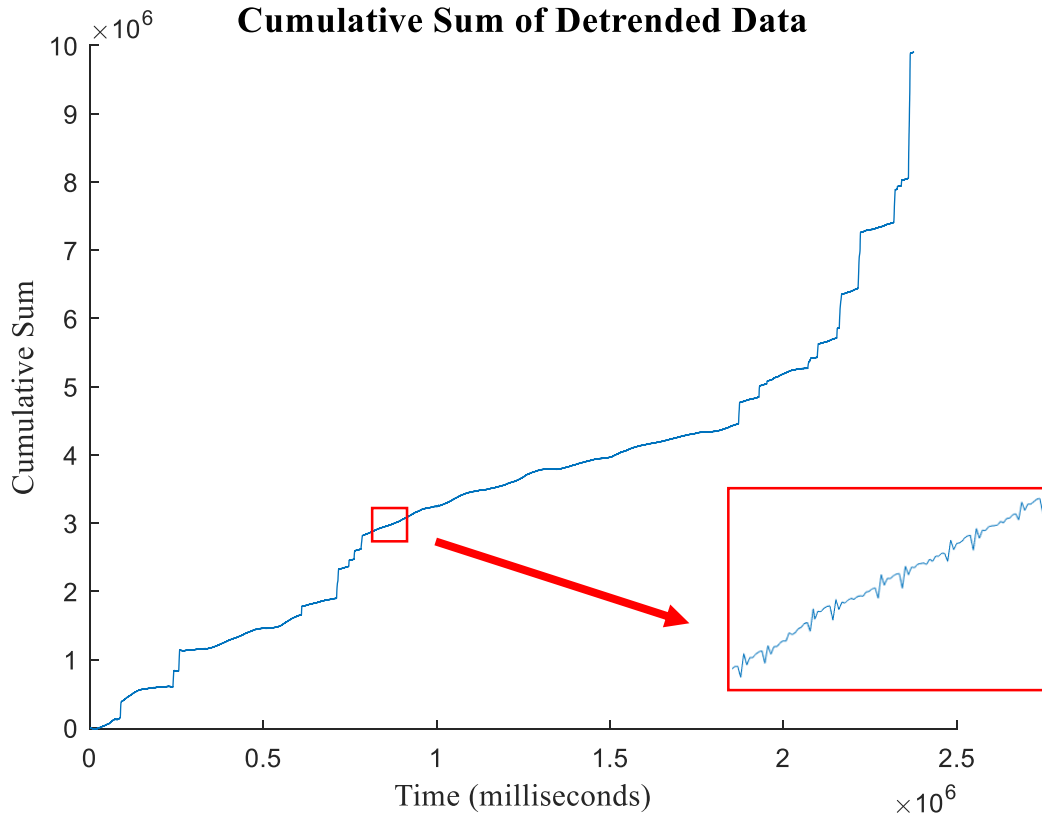


Figure 10. Cumulative sum of the detrended data, which sums every point up to the current timestamp.

Overall, the cumulative sum data in Figure 10 has a positive slope, with large increases where there was a brake activation. The large increases maximize the scaling to a point where it can appear that the noise content was lost, but when zoomed in the noise signal can still be seen. Anytime the detrended

signal was positive, the cumulative sum increases. Likewise, when the detrended signal was negative, the cumulative sum decreases.

To provide the most accurate start and end times, the median of peak activations was found. Instead of looking for positive and negative values, the peaks were found off of above or below this median value. Using the median value ensures that any noise or slight miscalculations of the null offset don't cause the activations to seem longer than they actually are. The start of the peak was considered to be the point before the median crossing, and the end of the peak was the point after the median crossing.

3.4 Brake Activation Determination

The final step is to determine which peaks are the brake activations. All the data has been characterized as either being part of a negative or positive peak. Since the brake activation direction has already been determined (and made positive if it wasn't already), negative peaks are discounted. All of the peaks are put into a cumulative distribution function (CDF), as shown in Figure 11. A CDF is essentially the same as probability distribution function (PDF), which produces the probability of a value occurring. The difference is that a CDF is on already existing points, so instead of producing a probability percentage it gives the percentage of points that are below a certain value.

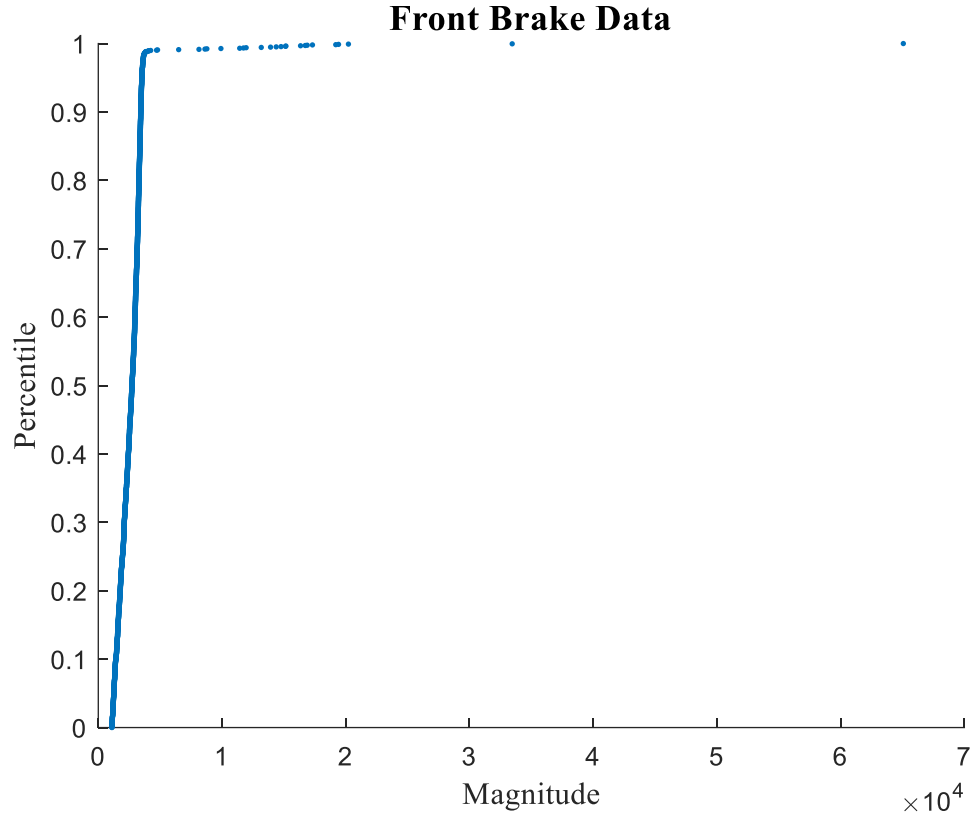


Figure 11. Cumulative distribution function of the positive peaks.

This percentile of points is useful because it concentrates all of the peaks from the noise together, while accentuating the brake activation peaks. In Figure 11, almost all of the positive peaks are grouped into a line with a large slope. There is a sharp turn where the data goes from being extremely concentrated to spread out. The point of the sharp turn is where the amplitude of the positive peaks becomes significantly greater than the previous peaks. Because the brake activations are extreme outliers compared to the rest of the data, the points above the value will be brake activations.

In order to find this point, the gap between points is found, defined as

$$g_i = x_{i+1} - x_i \quad \text{Eq. 13}$$

where g_i is the gap from the current point at index i to the next point. The results from using Eq. 13 on every point in plotted in Figure 12.

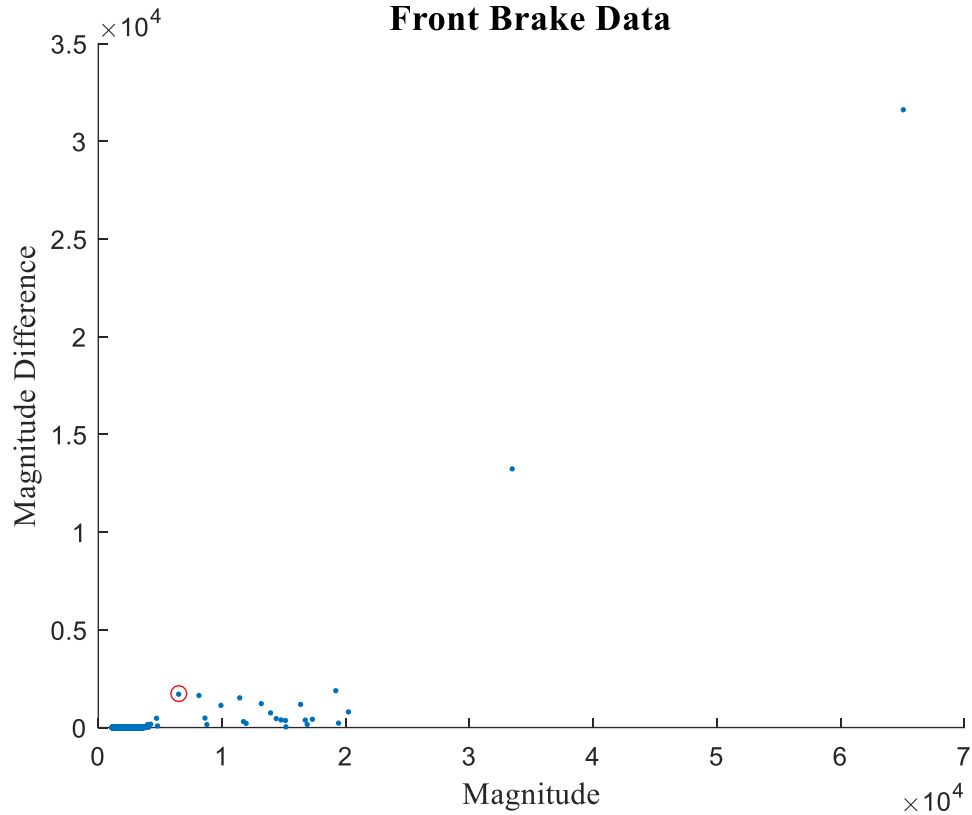


Figure 12. The amplitude gaps between points from Figure 11, calculated using Eq. 13. The red circle is around the point of the lowest amplitude considered a brake activation.

An amplitude threshold can be calculated using the data from Figure 12 to find the point considered to be the start of brake activations. This threshold, MT , is defined as

$$MT = 2 * SDG \quad \text{Eq. 14}$$

where SDG is the standard deviation of all the gaps. All the positive peaks above the threshold are considered activations. Eq. 14 follows the reasoning that data that are normally distributed have outliers that are greater than 2 times the standard deviation. The noise can be considered to be basically normally distributed, and since the brake activations are considered outliers this equation does a great job at finding the brake activation peaks.

Figure 13 shows the original data from Figure 4 with the brake activations determined by this method highlighted.

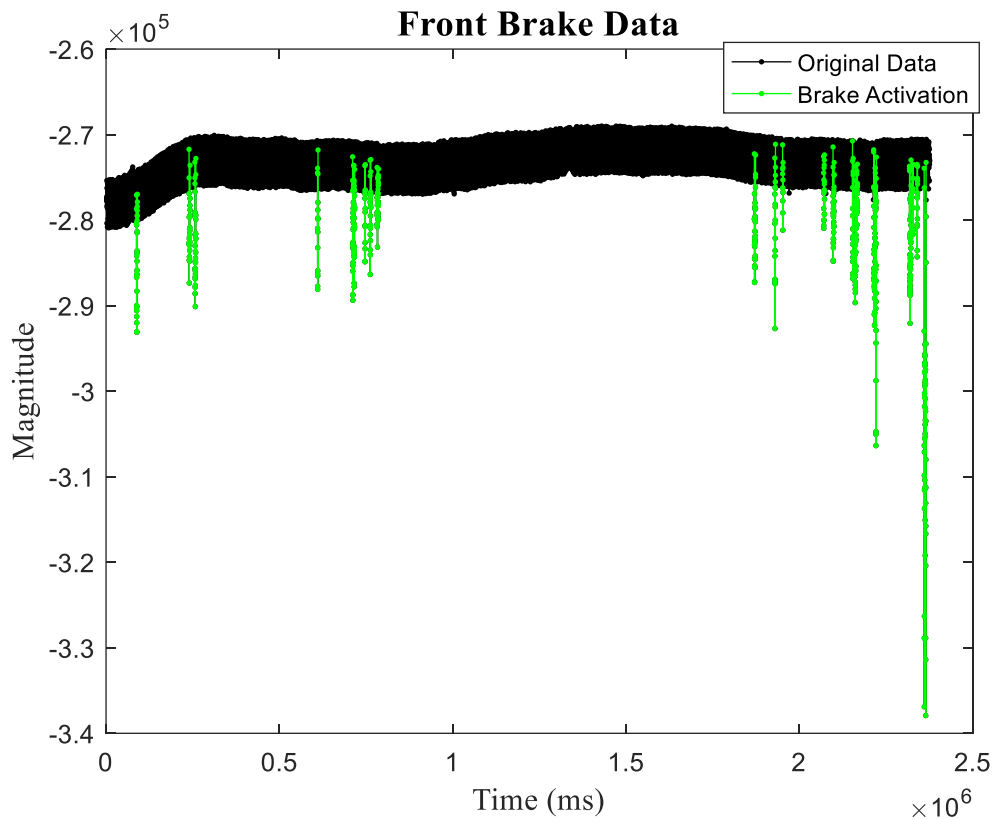


Figure 13. Original data with brake activations highlighted in green.

3.5 Execution of Algorithm in MATLAB

The main code of this algorithm is written as a function in MATLAB. This makes it simple to iterate through trips, as the data from a trip can be input to the function. All of the motorcycle NDS data was kept in DB2 tables, which is where the data was queried from, and to which the brake activations were written to when they were found. The function will take an input in the form of two integers and a matrix that can easily be queried from a table, and likewise produces an output that can be put back into a table. Another useful aspect of having this algorithm as a function is that it can be integrated into a parfor loop, or create jobs to send to a cluster for parallel processing. This is an important for processing this NDS data, as there were over 40,000 trips to evaluate.

3.6 Chapter Conclusion

Strain gauge data from the front and rear brakes first is processed to find the null offset line. This is accomplished by applying a data range and smoothing the data with a LOWESS regression, iterating this process four times to ensure accuracy. The data is detrended by subtracting the null offset from the original data, and validated to make sure that brake activations were recorded and the data wasn't compromised. A cumulative sum determines positive peaks from negative peaks, and a cumulative distribution function of the peaks separates brake activations from the noise. This algorithm is implemented as set of functions written in MATLAB that are in the attached Appendix.

Chapter 4. Discussion of Method

4.0 Chapter Introduction

Using the algorithm described in Chapter 3 (see the appendices for the MATLAB code), strain gauge data from motorcycle trips were analyzed for brake activations. This chapter will discuss the accuracy of the algorithm to detect brake presses and establish start/end times, and how the peak amplitude value from a brake press was used for driving behavior analysis.

4.1 Peak Detection

The peak detection algorithm, explained in Chapter 3, has to be able to accurately and reliably find brake activations in strain gauge data that varies wildly between trips and motorcycles. The frequency at which the brakes are hit, as well as the amplitude of brake peaks, change depending on how the participant uses the brakes in a trip. Also, the strain gauge picked up external signals that introduced noise that was different for every trip and bike. The combination of how the brakes were used, along with the effect of recorded external signals, affected the reliability of the algorithm to accurately pick out brake activations. In order to break down how the relationship between brake activations, noise, and the ability of the algorithm to determine brake activations correlated, the noise content is described by the following characteristics:

- Noise frequency – The frequency of an external signal or signals that was recorded by the data acquisition system (DAS). These signals show up in the strain gauge data, but are not actually from the strain gauge being stressed. The exact frequency of this noise is not calculated, as the strain gauge data was not sampled at a high enough rate to be determine this. Instead, this is a qualitative attribute that can be seen by looking at the distance between peaks of the noise signal.
- Noise amplitude – The peak amplitude of the noise. Specifically, the amplitude of the noise signal relative to the brake activation’s amplitude is an important factor in peak detection accuracy.

- Null offset shift – The amount the null offset changes over time. The null offset is the value of the centerline of the data, the point that should be zero when the strain gauge is neither in tension or compression. This value changes over time, and can increase and decrease in value over a trip.

These characteristics are all descriptive qualitative attributes determined by visually inspecting a trip, not exact quantities that are calculated. Some combinations of these variations affected the accuracy of the algorithm more than others. A sample of files were reduced using video data of the front brake, and speed and acceleration data, so that times when the brakes were activated were verified. The brake activations found by the algorithm compared to the video reduction were compared.

Figure 14 shows the data recorded from a strain gauge, with the brake activations determined by the algorithm highlighted in green. The peak amplitude of the noise relative to the peak amplitude of the brake activations is small, the null offset has little drift over time, and the noise frequency doesn't appear to change. All brake activations highlighted by the algorithm were all verified to be correct, and the algorithm found all brake activations present in the trip.

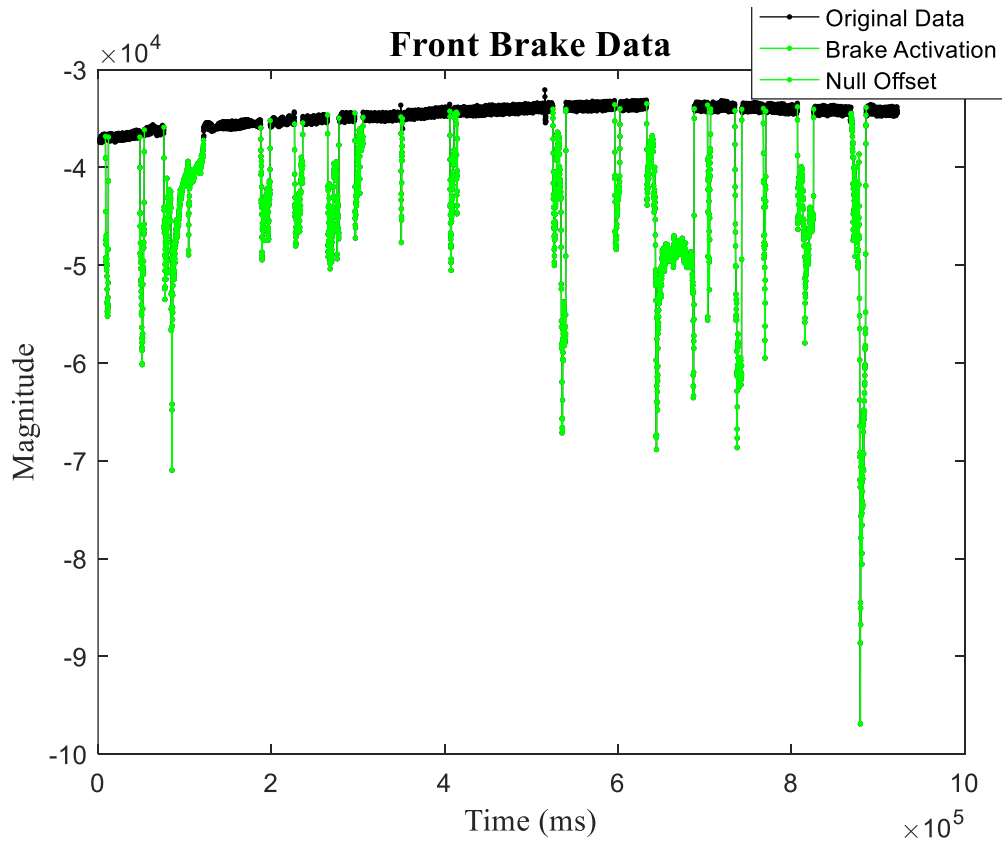


Figure 14. Brake activations highlighted for a file where the peak amplitude of the noise was relatively small compared to the brake activations, there was a slight null offset shift, noise frequency seemingly stayed the same, and brake activations were negative.

In Figure 15, the strain gauge data has a noise frequency that varied greatly over time, seen most clearly in the data ranging from 0 to 1,000,000 milliseconds. The noise data starts off with peaks fairly close together, but starts to get more spread apart and has its farthest distance between peaks at 500,000 milliseconds. The noise amplitudes have a significant value, but their repeated nature help identify them as noise. The null offset also shifts over time, decreasing and increasing in value. Again, all brake activations were correctly identified by the algorithm as verified through the deceleration and speed data.

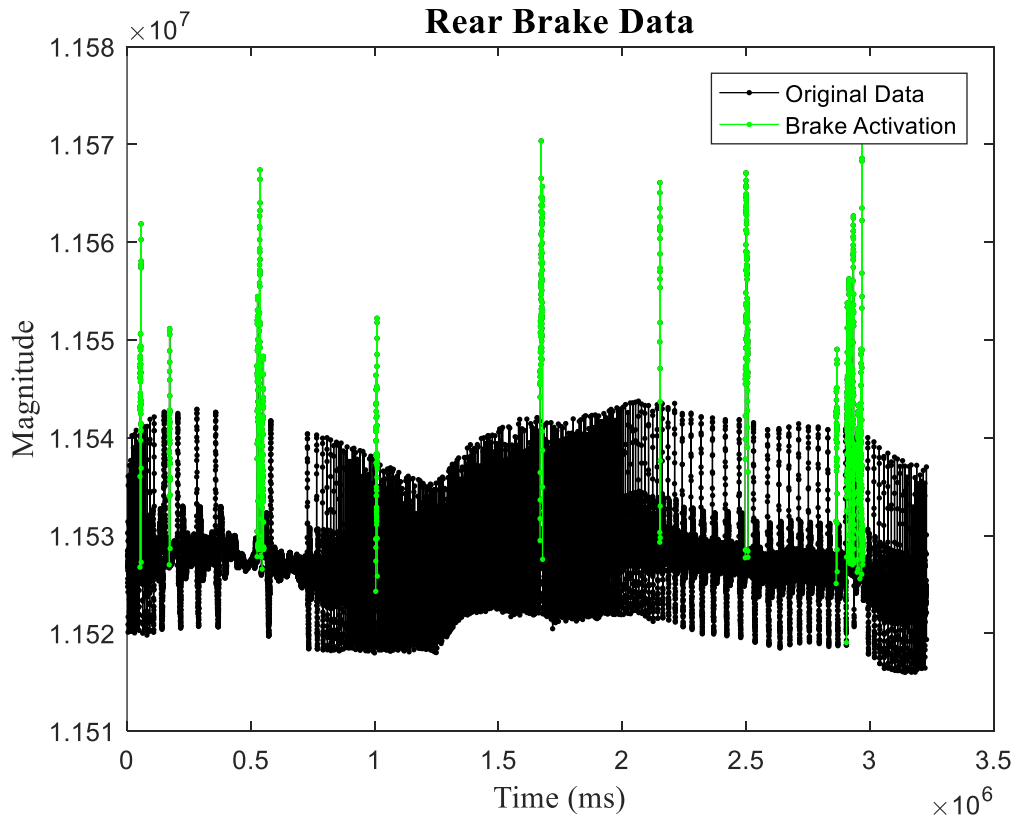


Figure 15. Brake activations highlighted for a file where the null offset changes over time, the noise peak amplitudes are relatively large and noise frequency changes over time.

Figure 16 has brake activations highlighted in green from a file with little null offset drift and a moderate noise to activation amplitude ratio. The large brake activations were all correctly identified by the algorithm. However, the brake activations circled in red are not. These smaller brake activations have a low noise to activation amplitude ratio. Even in visual inspection of the graph, it is not immediately apparent that these are brake activations because of their high frequency and small amplitudes compared to the noise. A video reduction, though, confirmed that they were brake activations.

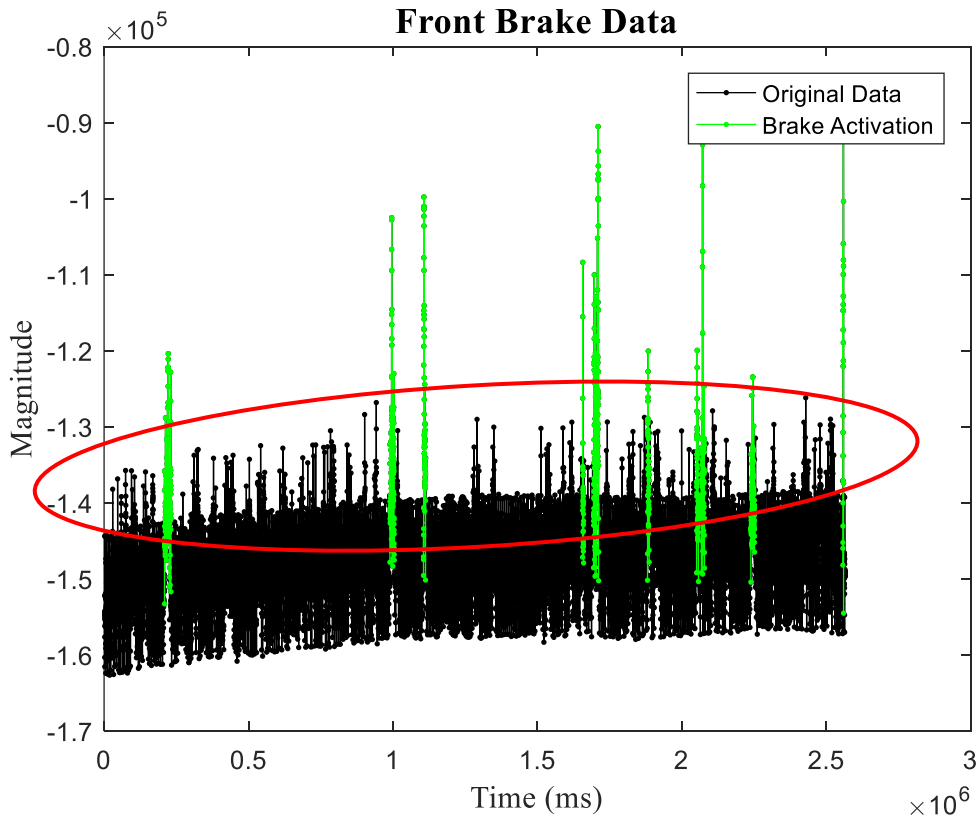


Figure 16. Brake Activations highlighted in green for a file where there is a mix of large amplitude and smaller, yet frequent, amplitude brake activations. Brake activations that were not found by the algorithm are circled in red.

While there were many brake activations missed in this file, this combination of characteristics (high number of smaller brake activations mixed with large brake activations, with a small noise to activation amplitude ratio) was not common in the sampled files. Out of the 50 sampled, only one showed this much inaccuracy, or 2%. The other files were at least 90% accurate in peak detection. Figure 17 and Figure 18 are examples of two more trips, with the brake activations identified by the peak detection algorithm highlighted.

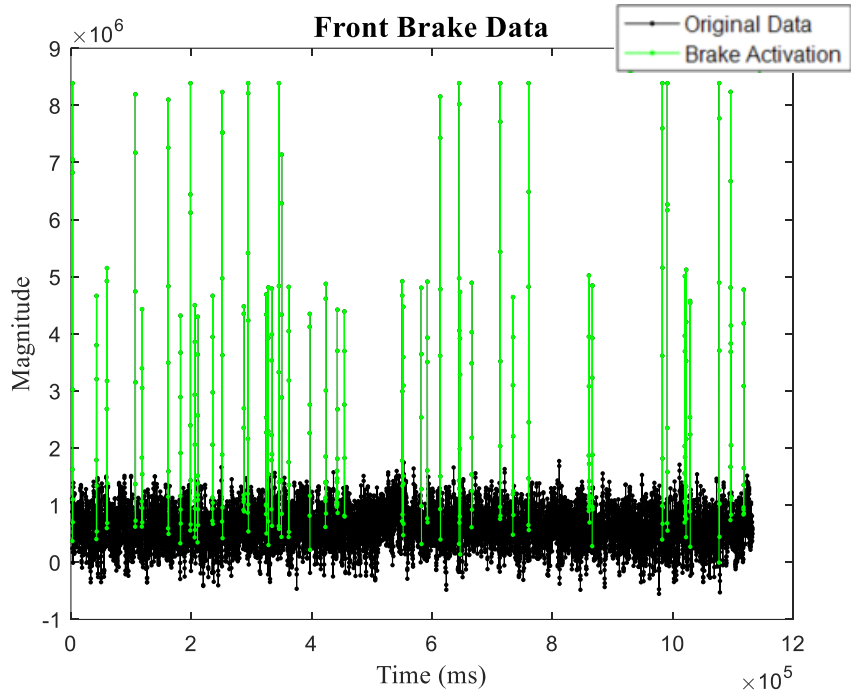


Figure 17. Brake activations highlighted for a file that has a relatively small noise amplitude, frequent brake presses, and little null offset shift over time.

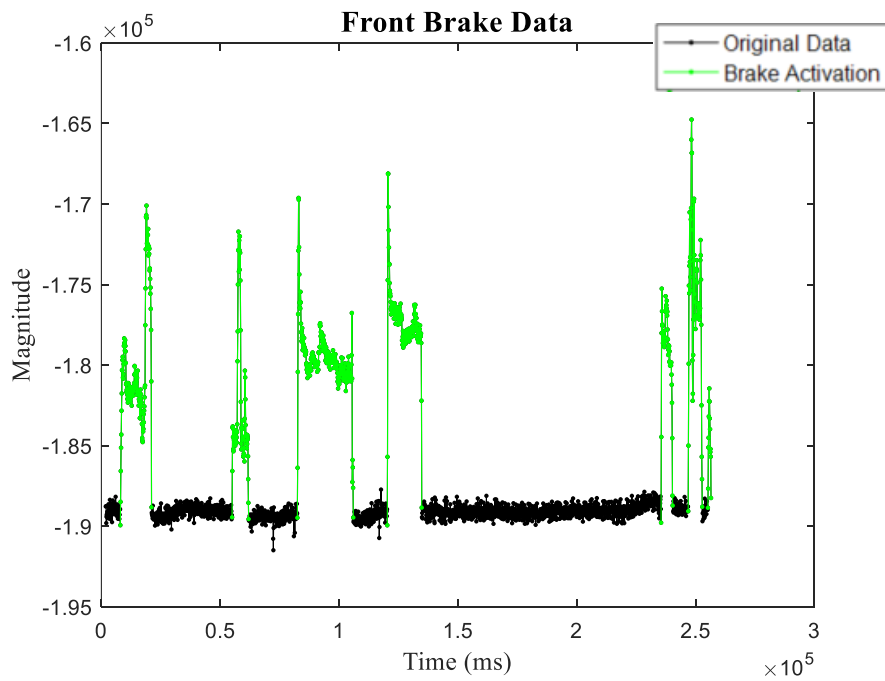


Figure 18. Brake activations highlighted for a file that has a relatively small noise amplitude, little null offset shift over time with more infrequent and longer brake presses.

4.2 Peak Bounds Accuracy

The null offset is the value of the data when the strain gauge is at rest. Ideally, it makes sense to calculate the start and end times of the brake activations as the point where the data stream crosses the null offset. However, noise in the signal can greatly affect these bounds if the null offset is used. The algorithm described in Chapter 3 instead used the median value of all positive peaks as the starting and end point of a brake activation.

Figure 19 shows a sample of strain gauge data highlighting three brake activations. The second two activations would be considered the same activation if compared to the null offset. However, by using the median of peaks, this is broken into two activations. A reduction of the video confirmed that these are, in fact, two separate activations.

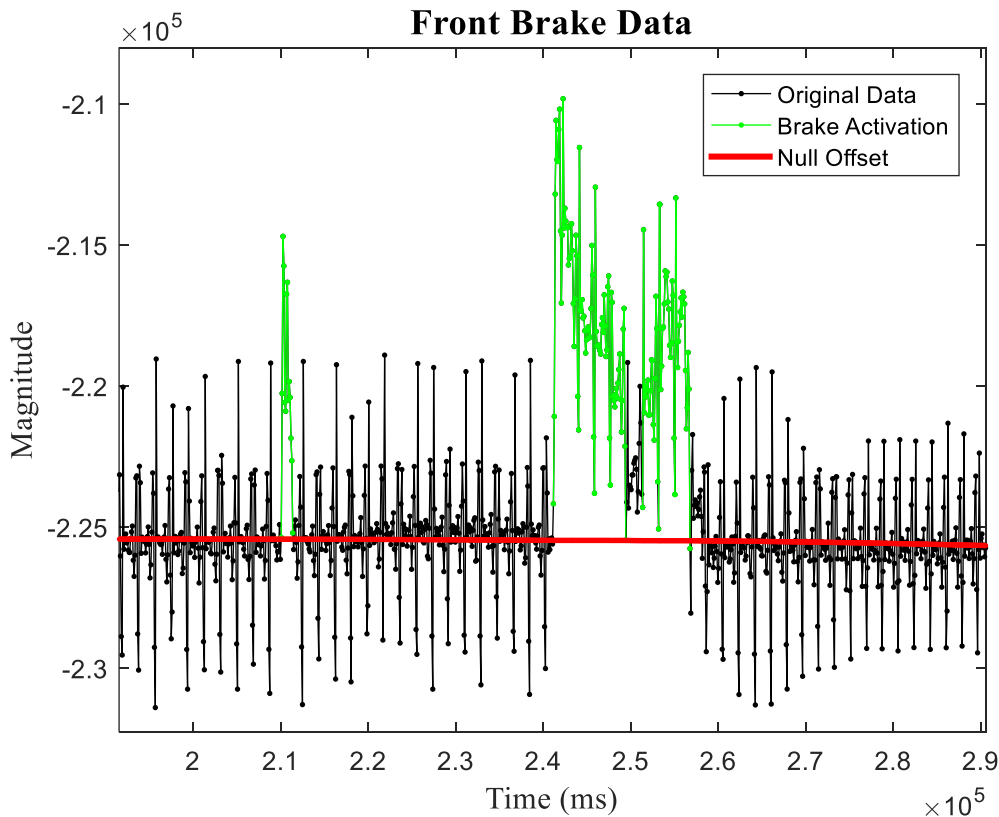


Figure 19. Activations are separated by the median, rather than the null offset.

4.3 Magnitude Use

The purpose of finding these brake activations are to use them to describe driving behavior of motorcyclists. The way that they brake, both the magnitude of a brake press and the frequency at which they brake, can help researchers understand more about safety critical events (SCEs), like a crash or near-crash. Because of differences in brake levers, installation techniques and other factors, the detrended amplitudes of brake activations should not be directly compared between bikes. Instead, these brake activations should be related to each other by participant and categorized before comparing across bikes.

By calculating a cumulative distribution function (CDF) of all the detrended brake activation amplitudes for a participant, every brake activation can be transformed into a percentile of how hard the brake was pressed compared to others. These percentiles can be described in qualitative measures, which are able to be compared between bikes that had similar scenarios. For example, one participant who avoided a crash might have had a harder than normal brake press, whereas a participant who was in a similar scenario performed a softer than normal brake press and was involved in an accident. This is a very basic way to compare across activations, and get an idea of braking behavior.

The speed and acceleration data that was also recorded can give a more in-depth and quantitative result for each brake activation. Figure 20 shows an example of the speed and acceleration data during a brake activation. Though the acceleration data is a little noisy, it can be filtered to find the max deceleration during the braking instance. Combined with speed, a profile of braking strength for each participant at certain speeds can easily be established, which provides a more useful description of a participant's braking behavior.

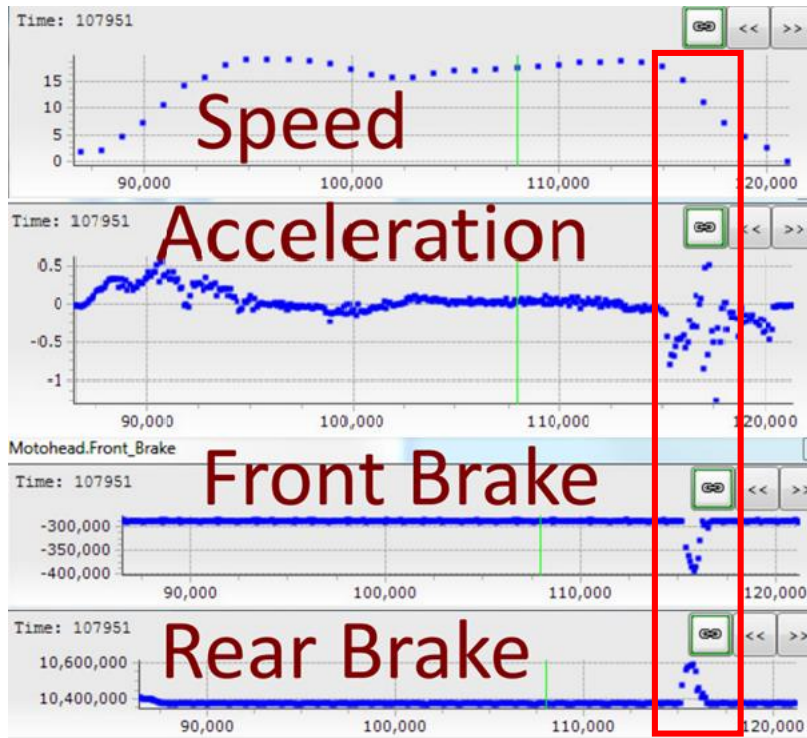


Figure 20. Example of acceleration and speed data during a brake activation. Both brakes were used, with the red box outlining the activation. [26]

4.4 Chapter Conclusion

Overall, the algorithm described in Chapter 3 correctly identifies brake presses in strain gauge data that varied between trips and bikes. The algorithm was not as accurate with files that had a large noise amplitude relative to the brake activation amplitude, but this only accounted for 2% of the sampled files. The rest of the files had a greater than 90% accuracy. The bounds of a brake activation were found to be the most accurate when using the median of all the brake presses as the start/end point, not just the null offset. To compare brake presses between trips and bikes to answer research questions about driving behavior, a CDF of the detrended brake amplitudes relates the data to percentiles that can then be compared. Combining this with other recorded data, like speed and acceleration, further expands this analysis.

Chapter 5. Conclusions

This thesis focuses on a post-processing method to find brake activations from strain gauges installed on brake levers of a motorcycle NDS. This method was developed for data that was already recorded, and it had to deal with noise issues that were present in the setup of the strain gauges and data acquisition system (DAS). During the literature review, it became clear that strain gauges are extremely sensitive to external forces and electromagnetic signals. And while care was taken when equipping the motorcycles with the strain gauges to prevent force on them when they were at rest, better hardware filtering could have been accomplished to eliminate more of these factors.

If this study were to be repeated, the wires leading from the strain gauges to the DAS should be shielded against other electromagnetic interference [15]. The effect of temperature change proved to be significant, so another strain gauge should be installed in a transverse direction of the original [13], or on another unstressed spot that is made of the same material as the brake lever yet is still subjected to the same shifts in temperature as the original gauge [14]. These secondary gauges can help calibrate the reading from the first gauge to eliminate the null offset. The DAS could also pre-process the data by running it through a low pass filter to eliminate any high frequency noise still in the signal. Also, it is incredibly important that the data is well documented in how it was recorded and processed. Without this information, it makes it difficult to quantitatively evaluate the brake activations.

The algorithm can be improved to be more accurate in finding activations in trips where the amplitude of the noise was relatively larger compared to the amplitude of the brake activations. These files were not very common (2% of the sampled data) and are not projected to have a significant effect on the results of the braking behavior analysis completed by VTTI, but if this algorithm is used in other applications it may become more significant. Adjusting the parameters used to determine peaks, like Eq. 14, or using wavelet theory, may help improve the peak detection accuracy.

Chapter 6. Summary

Strain gauges were installed on motorcycles in a naturalistic driving study (NDS) to be able to record when brakes were pressed, and how hard they were pressed. This thesis outlined an algorithm developed to find these brake activations, which showed up as significant outlier peaks in the strain gauge data. Proper hardware filtering was not implemented while recording the data, so the recorded data was full of noise from electromagnetic signals and other interference. The algorithm follows this process:

- Determine a null offset using a LOWESS regression over an established data range. Repeat this three more times to increase accuracy of the null offset line.
- Subtract this calculated null offset from the data to find the detrended data set.
- Using the detrended data, determine valid data by analyzing peak ratios. Invalid data would be cases where the strain gauge fell off of the brake lever, or otherwise was so noisy that no valid brake activations would be possible to be found.
- Find all peaks, positive and negative, using a cumulative sum.
- Determine peaks that can be considered brake activations by first using a cumulative distribution function to order the peaks by amplitude value. By calculating the gaps in amplitude between peaks, the higher brake activation amplitudes can be identified.

This algorithm was found to be accurate in determining brake activations, though some improvement can be made for trips with a relatively large noise amplitude compared to brake activation amplitude. It was also found that it was more accurate to use the median peak value of all positive peaks as the start and end crossing point for activations, than using the null offset.

Because the unit of measure was unknown, and different bikes used different brake levers that could have an effect on the strain induced by an activation, they were instead transformed to percentiles using a cumulative distribution function. This way each activation's peak was compared to all others within that bike. These percentiles could then be compared between bikes by qualitatively describing the data. Using

data from other sensors, specifically speed and acceleration data, greatly increases the depth of analysis of the brake activations.

In conclusion, it is best to filter strain gauge data as much as possible using hardware, good wiring practices, and pre-processing of the data with the data acquisition system. This is because strain gauges are extremely sensitive to any forces and external signals. However, this algorithm was found to consistently find brake activations, and established a system for comparing brake values that did not have a known quantitative description.

Bibliography

- [1] "About VTTI," [Online]. Available: <http://www.vtti.vt.edu/about/about-vtti.html>. [Accessed 11 11 2016].
- [2] G. A. S. S. A. G. F. M. J. F. Y. O. R. L. P. M. A. H. R. J. H. J. M. & D. T. A. Fitch, "The impact of hand-held and hands-free cell phone use on driving performance and safety-critical event risk.," National Highway Traffic Safety Administration, Washington, DC, 2013.
- [3] S. D. T. A. N. V. L. S. J. a. R. D. Klauer, "The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data," National Highway Traffic Safety Administration, Washington, DC, 2006.
- [4] V. K. S. K. R. D. T. H. Neale, "The 100 Car Naturalistic Driving Study," U.S. Department of Transportation, National Highway Traffic Administration, Washington D.C., 2002.
- [5] "InSight Data Access Website," [Online]. Available: <https://insight.shrp2nds.us/>.
- [6] K. Grove, J. Atwood, P. Hill, G. Fitch, M. Blanco, F. Guo, S. Russell, M. Marchese, P. Bartholomew and T. Richards, "Field Study of Heavy-Vehicle Crash Avoidance Systems," National Highway and Traffic Safety Administration, Washington D.C., 2016.
- [7] M. Blanco, R. J. Hanowski, R. L. Olson, J. F. Morgan, S. A. Soccolich, S.-C. Wu and F. Guo, "The Impact of Driving, Non-Driving Work, and Rest Breaks on Driving Performance in Commercial Motor Vehicle Operations," Federal Motor Carrier Safety Administration, Washington, DC, 2011.
- [8] V. Williams, S. McLaughlin and J. Atwood, "Factors that Increase and Decrease Motorcyclist Crash Risk," Motorcycle Safety Foundation, Irvine, CA, 2016.
- [9] S. G. D. T. A. N. V. L. S. J. D. a. R. D. J. Klauer, "Comparing Real-World Behaviors of Drivers With High versus Low Rates of Crashes and Near-Crashes," National Highway Traffic Safety Administration , Washington, DC, 2009.
- [10] W. G. J. H. A. E. G. & J. S. S. L. E. Lanyon, "Bone Deformation Recorded in vivo from Strain Gauges Attached to the Human Tibial Shaft," *Acta Orthopaedica Scandinavica*, vol. 46, no. 2, pp. 256-268, 1975.
- [11] P. Omenzetter and J. M. W. Brownjohn, "Application of time series analysis for bridge monitoring," *Institute of Physics Publishing*, vol. 15, pp. 129-138, 2006.
- [12] National Instruments, "Strain Gauge Measurement – A Tutorial," August 1998. [Online]. Available: http://elektron.pol.lublin.pl/elekp/ap_notes/NI_AN078_Strain_Gauge_Meas.pdf. [Accessed 2016].
- [13] K. Hoffmann, "Applying the Wheatstone Bridge Circuit," [Online]. Available: www.hbm.com.

- [14] Micro-Measurements, "Strain Gage Thermal Output and Gage Factor Variation with Temperature," 2014.
- [15] "Noise Control in Strain Gage Measurements," 09 July 2013. [Online]. Available: www.micro-measurements.com. [Accessed 8 November 2016].
- [16] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, www.DSPguide.com, 2001.
- [17] R. Kamnik, B. Kovačič, A. Štrukelj, N. Vatin and V. Murgul, "Preparation, installation and signal processing of strain gauges in bridge load testing," *Applied Mechanics and Materials*, Vols. 725-726, pp. 903-912, 2014.
- [18] G. H. Tan, Y. K. Poh, N. H. Osborne, C. C. Ng and J. M. Brownjohn, "Reducing False Alerts by improving Vibrating Wire Strain Gauge Data Quality using Fast Fourier Transform," *International Conference on Deep Excavations*, 2008.
- [19] B. A. Olshausen, "Aliasing," 10 October 2000. [Online]. Available: <http://redwood.berkeley.edu/bruno/npb261/aliasing.pdf>.
- [20] A. V. Oppenheim, R. W. Schaffer and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed., Englewood Cliffs, NJ: Prentice-Hall, 1999.
- [21] C. Yang, Z. He and W. Yu, "Comparison of public peak detection algorithms for MALDI mass spectrometry data analysis," *BMC Bioinformatics*, 2009.
- [22] "Filtering and Smoothing Data," MathWorks, [Online]. Available: <https://www.mathworks.com/help/curvefit/smoothing-data.html>.
- [23] K. R. Coombes, S. Tsavachidis, J. S. Morris, K. A. Baggerly, M.-C. Hung and H. M. Kuerer, "Improved Peak Detection and Quantification of Mass Spectrometry Data Acquired from Surface-Enhanced Laser Desorption and Ionization by Denoising Spectra with the Undecimated Discrete Wavelet Transform," The University of Texas, Houston, TX.
- [24] K. R. Coombes, H. A. Fritsche Jr., C. Clarke, J.-N. Chen, K. A. Baggerly, J. S. Morris, L.-C. Xiao, M.-C. Hung and H. M. Kuerer, "Quality Control and Peak Finding for Proteomics Data Collected from Nipple Aspirate Fluid by Surface-Enhanced Laser Desorption and Ionization," vol. 49, no. 10, pp. 1615-1623, 2003.
- [25] D. C. Montgomery, *Introduction to Statistical Quality Control*, Third Edition, Hoboken, NJ: John Wiley & Sons, Inc., 1997.
- [26] M. McCall, "<https://www.msf-usa.org>," 2015. [Online]. Available: https://www.msf-usa.org/downloads/msf100_2016/Motorcycle_Research_at_VTTI_MSF_100_Study_presentation.pdf. [Accessed 5/16/17 May 2017].

Appendix

Appendix A. MATLAB Code: Main.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MATLAB code written for 'Post-Processing Method for Determining Peaks %
% in Noisy Strain Gauge Data with a Low Sampling Frequency', supporting %
% the thesis for %
%
%                               Master of Science %
%                               in %
%                               Mechanical Engineering %
%
%                               Peter L. Hill %
%                               Virginia Tech %
%                               Blacksburg, VA %
%                               Dr. Al Wicks, Committee Chair %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This function uses the peak detection process described in Chapter 3 to
% find the brake activations of motorcycles equipped with strain gauges on
% the brake levers. The inputs to this function are acquired from a DB2
% database where the strain gauge data is kept, and the results are written
% back to that database.
%
% Outputs:
```

```

% --File_Output: A matrix with 8 columns
%
%     --Column 1: The trip's unique identifier
%
%     --Column 2: Timestamp start of brake activation in
%
%                 milliseconds
%
%     --Column 3: Timestamp end of brake activation in
%
%                 milliseconds
%
%     --Column 4: Detrended brake activation max magnitude
%
%     --Column 5: Original brake activation max magnitude
%
%     --Column 6: Timestamp of max magnitude in brake activation
%
%                 in milliseconds
%
%     --Column 7: Index of max magnitude in brake activation
%
%     --Column 8: The vehicle's unique identifier
%
% --File_Status: Status of the data
%
%     -->2: File ran succesfully and brake activations were found
%
%     -->4: File did not have valid data
%
%     -->5: Not brake activations were found
%
%     -->6: File was not at least 4 minutes long
%
%     -->8: Null offset was not created
%
%     -->9: No cumulative sum was created
%
%
% Inputs:
%
% --File_ID: Unique identifier for that trip
%
% --Data: Time-series Matrix queried from DB2 database with 2 columns
%
%     --Column 1: Timestamp in milliseconds
%
%     --Column 2: Magnitude recorded from strain gauge
%
% --Vehicle_ID: Unique identifier for the vehicle

```

```

function [File_Output,File_Status] = Main(File_ID,Data,Vehicle_ID)

%Initialize output matrix that will be recording in result table
File_Output = [];

%Check that the file was at least 4 minutes long. There needs to enough
%points that an accurate null offset can be established using the function
%"Null_Offset", which smooths data using LOWESS regression across 2000
%points. Sampling at 10Hz, 4 minutes provide 2400 data points. Time is in
%milliseconds
if (Data(end,1) - (Data(1,1))) > 240000

%Function that returns the null offset of the current file
Null_Offset_Line = Null_Offset(Data);

if ~isempty(Null_Offset_Line)

%Subtract the null offset from the current file to create the detrended
%data
Detrended_Data = [Data(:,1) Data(:,2)-Null_Offset_Line(:,2)];

%Make sure the data is valid. Files where the variance of the
%original data compared to detrended data typically did not have
%valid data. The 0.5 number came from testing a set of valid and
%nonvalid files, and it was found that valid files had a ratio

```

```

%greater than 0.5
if std(Detrended_Data(:,2))/std(Data(:,2)) > 0.5

%Determine activation direction by finding which direction, positive or
%negative, had the greatest peaks
if max(Detrended_Data(:,2)) <= abs(min(Detrended_Data(:,2)))

%If this is true, direction is negative. In this case, multiply
%all data by -1 and treat as if positive activations. It will be
%multiplied back by -1 at the end
Data = [Data(:,1) (Data(:,2) * -1)];
Detrended_Data = [Detrended_Data(:,1) Detrended_Data(:,2) * -1];
Null_Offset_Line = [Null_Offset_Line(:,1) Null_Offset_Line(:,2)*-1];
end

%Complete cumulative sum, grouping positive and negative runs together
Cu_Groups = Find_CuSum_Groups(Detrended_Data);

if ~isempty(Cu_Groups)

%Find median activation threshold above the null offset. This helps
%with breaking up some activations and having more accurate start/end
%times
Cu_Groups_Pos = Cu_Groups(Cu_Groups(:,5) == 1,:);
Cu_Groups_Mags = zeros(size(Cu_Groups_Pos,1),1);

%Iterate through positive peaks and find the max detrended value in
%that peak

```

```

for i = 1:size(Cu_Groups_Pos,1)

    Start_Index = find(Detrended_Data(:,1) == Cu_Groups_Pos(i,1));
    End_Index = find(Detrended_Data(:,1) == Cu_Groups_Pos(i,2));
    Cu_Groups_Mags(i,1) = max(Detrended_Data(Start_Index:End_Index,2));
end

%Find the median of the positive peaks. This will be about the peak
%amplitude of the noise when the strain gauge is at rest
Median_Magnitude = median(Cu_Groups_Mags);

%Using the calculated median of the original Cu_group_Activations,
%perform the cusum again to find more accurate start/end timestamps of
%the peaks
Med_Detrended_Data = [Data(:,1) ...
    (Data(:,2) - (Null_Offset_Line(:,2) + Median_Magnitude))];
Cu_Group_TS = Find_CuSum_Groups(Med_Detrended_Data);

%Find the positive peaks, grouping the detrended and original max
%magnitudes of the peaks
Positive_Indexes = find(Cu_Group_TS(:,5)==1);
%Initialize matrix
Activations = zeros(size(Positive_Indexes,1),6);
for i = 1:size(Positive_Indexes,1)
    Curr_Mag = Cu_Group_TS(Positive_Indexes(i),3);
    Start_Index = find(Data(:,1)== Cu_Group_TS(Positive_Indexes(i),1));
    Stop_Index = find(Data(:,1) == Cu_Group_TS(Positive_Indexes(i),2));
    %Original data peak magnitude

```

```

Max_Value = max(Data(Start_Index:Stop_Index,2));

%Index of magnitude within peak

Max_Index = find(Data(Start_Index:Stop_Index,2) == Max_Value(1,1));

%Add back in start_index to have the correct index of magnitude

%from the start of the file

Max_Index = Max_Index(1,1)+Start_Index-1;

%Actvation Matrix has columns: [timestamp start, timestamp end,
%   detrended max peak value, max peak of original data,
%   timestamp of max peak, index of max peak]

Activations(i,1:6) = [Cu_Group_TS(Positive_Indexes(i),1:2) ...
    Detrended_Data(Max_Index(1,1),2) Max_Value(1,1) ...
    Data(Max_Index(1,1),1) Max_Index(1,1)];

end

%Find the point where positive peaks in the detrended data are
%significantly higher than others, meaning they are brake presses

if ~isempty(Activations)

    Mag_Thrshold = Point_Diff(Activations(:,3));

end

%Find all the peak values that are greater than or equal to the scaled
%peak magnitude that is considered a brake activation.

Final_Activations = Activations(Activations(:,3)>=Mag_Thrshold(1,1),:);

%Put File_ID and Vehicle_ID in with Activations. If there
%were no activations, set the file status to 5. If there

```

```

%were activations, set the file status to 2
if isempty(Final_Activations)
    File_Status = 5;
else
    File_Status = 2;
    Ones_Mat = ones(size(Final_Activations,1),1);
    File_Output = [File_ID*Ones_Mat Final_Activations ...
        Vehicle_ID*Ones_Mat];
end
else
    File_Status = 9; %No cumulative sum data created
end
else
    File_Status = 4; %File did not have valid data
end
else
    File_Status = 8; %Null offset was not created
end
else
    File_Status = 6; %File was not greater than 4 minutes
end

```


Appendix B. MATLAB Code: Null_Offset.m

```
% This is a sub function used in the "Main" function. This function computes  
% the null offset through iterative LOWESS regression from a dataset input
```

```
% Output:
```

```
% --Null_Offset_Line: Matrix with two columns
```

```
%           --Column 1: Timestamp in milliseconds
```

```
%           --Column 2: Calculated null offset value
```

```
% Inputs:
```

```
% --Data: Matrix with two columns:
```

```
%           --Column 1: Timestamp in milliseconds
```

```
%           --Column 2: Recorded strain gauge value
```

```
function Null_Offset_Line = Null_Offset(Data)
```

```
% Initialize output matrix
```

```
Null_Offset_Line = [];
```

```
% Standard deviation of dataset
```

```
STD = std(Data(:,2));
```

```
% Median of data set
```

```
MED = median(Data(:,2));
```

```
% Find initial data window, which is the data within the null offset +- the
```

```

%standard deviation. For the first window, the median is used.
Data_Range = Data(Data(:,2) < (MED+STD) & Data(:,2) > (MED-STD),:);
%Perform LOWESS regression to smooth data and find null offset
Lowess_STD_Data = [Data_Range(:,1) smooth(Data_Range(:,2),2000,'lowess')];

if ~isempty(Lowess_STD_Data)

%The original dataset and null offset need to have the same amount of
%points, so the Fill_Gaps function puts data points where they are
%missing since establishing the data range excludes some points. Find a
%new data range using the calculated Null_Offset_Line, and smooth that
%range with LOWESS regression.
Null_Offset_Line = Fill_Gaps(Data,Lowess_STD_Data);
Data_Range2 = Data(Data(:,2) < (Null_Offset_Line(:,2)+STD) ...
    & Data(:,2) > (Null_Offset_Line(:,2)-STD),:);
Lowess_STD_Data2 = [Data_Range2(:,1) ...
    smooth(Data_Range2(:,2),2000,'lowess')];

%Repeat the previous steps
Null_Offset_Line = Fill_Gaps(Data,Lowess_STD_Data2);
Data_Range2 = Data(Data(:,2) < (Null_Offset_Line(:,2)+STD) ...
    & Data(:,2) > (Null_Offset_Line(:,2)-STD),:);
Lowess_STD_Data3 = [Data_Range2(:,1) ...
    smooth(Data_Range2(:,2),2000,'lowess')];

%Repeat the previous steps

```

```
Null_Offset_Line = Fill_Gaps(Data,Lowess_STD_Data3);
Data_Range2 = Data(Data(:,2) < (Null_Offset_Line(:,2)+STD) ...
    & Data(:,2) > (Null_Offset_Line(:,2)-STD),:);
Lowess_STD_Data4 = [Data_Range2(:,1) ...
    smooth(Data_Range2(:,2),2000,'lowess')];

%Fill Gaps of final LOWESS regression to have the final Null Offset
Null_Offset_Line = Fill_Gaps(Data,Lowess_STD_Data4);
end
```


Appendix C. MATLAB Code: Fill_Gaps.m

```
% This is a sub function used in the "Main" function. This function looks
% for values missing in Column 1 of matrix B that are in Column 1 of matrix
% A. It then inserts this missing value into a new row of column B, and
% gives the second column a value comparable to the data around it.
```

```
% Output:
```

```
% --Processed_Data: This is matrix B with the missing values from matrix A
```

```
% Inputs:
```

```
% --A: Matrix with two columns. Column 1 should be the same data type as
```

```
% Column 1 of matrix B, containing all the same data and more than
```

```
% matrix B.
```

```
% --B: matrix with two columns. Data that is in column 1 of matrix A and
```

```
% not in column 1 of matrix B will be inserted into matrix B
```

```
function Processed_Data = Fill_Gaps(A,B)
```

```
% Find points that aren't in both matrices
```

```
Missing_Points = ismember(A(:,1),B(:,1));
```

```
% Iterative counter for matrix B
```

```
j=0;
```

```
% Initialize matrix
```

```
Processed_Data = [];
```

```

%Iterate through the Missing_Points matrix, which has a value of 1 if
%matrices A and B have data at the same point or 0 if not
for i = 1:length(Missing_Points)
    %If matrix B and A share a point
    if Missing_Points(i)==1
        %If there were two values for a timestamp, with only one in matrix B,
        %then j should not increment
        if i ~= 1 && A(i,1) == A(i-1,1)
            %If the next value in matrix B is the same as the current value
            if j ~= 0 && B(j,1) == B(j+1,1)
                j=j+1;
            elseif j == 0
                j = j+1;
            end
        else
            j=j+1;
        end
        Processed_Data = [Processed_Data; A(i,1) B(j,2)];
    else
        %If there isn't a point there and it's the first point (since j can't
        %stay 0 to select a data point)
        if i == 1
            Processed_Data = [Processed_Data; A(i,1) B(1,2)];
        %If there isn't a point there and it's not the first point
        elseif i ~= 1 && j==0
            Processed_Data = [Processed_Data; A(i,1) B(1,2)];
        end
    end
end

```

```
else
```

```
    Processed_Data = [Processed_Data; A(i,1) B(j,2)];
```

```
end
```

```
end
```

```
end
```

Appendix D. MATLAB Code: Find_CuSum_Groups.m

```
% This is a sub function used in the "Main" function. This function computes  
% the cumulative sum across the data input, and then uses this  
% data to determine the positive/negative peaks with start/end times.
```

```
%
```

```
% Output:
```

```
% --Cu_Groups: A matrix with five columns:
```

```
%     --Column 1: Timestamp start of peak in milliseconds
```

```
%     --Column 2: Timestamp end of peak in milliseconds
```

```
%     --Column 3: Amplitude of cumulative growth for that peak
```

```
%     --Column 4: Count of peaks in that peak
```

```
%     --Column 5: Direction of peak
```

```
%
```

```
% Input:
```

```
% --Detrended_Data: Original data with null offset subtracted from it
```

```
function Cu_Groups = Find_CuSum_Groups(Detrended_Data)
```

```
% Initialize variables
```

```
% Cumulative sum at each point of Detrended_Data
```

```
Cum_Sum_Data = zeros(size(Detrended_Data,1),1);
```

```
% Cumulative Sum up to index i
```

```
C_i = 0;
```

```
% Output matrix with columns defined at the top of this function
```



```

Cu_Groups = [];

%Number of points in peak

PCount = 0;

%Current direction of peak, 1 for positive and -1 for negative

Curr_Dir = 0;

%Direction for next peak

Next_Dir = 0;

%Complete cumulative sum of the data, which is defined as

% $C_i = \text{Sumation}(j=1 \text{ to } i)(x_j - \text{mean})$ , where  $(x_j - \text{mean})$  is the

%detrended_data magnitude value. At the same time, also keep track of the

%growth direction.

for j = 1:size(Detrended_Data,1)

%The very first point of Detrended_Data

if PCount == 0

Timestamp_Start = Detrended_Data(j,1);

Mag_Start = Detrended_Data(j,2);

PCount = 1;

Cum_Sum_Data(j,1) = Detrended_Data(j,2);

C_i = Cum_Sum_Data(j,1);

%This is the second point of the peak, meaning the direction of the

%peak can be determined

elseif PCount == 1

Cum_Sum_Data(j,1) = C_i + Detrended_Data(j,2);

C_i = Cum_Sum_Data(j,1);

```

```

Timestamp_End = Detrended_Data(j,1);
Mag_End = Cum_Sum_Data(j,1);
PCount = PCCount + 1;
Prev_Value = Cum_Sum_Data(j,1);
%Peak is negative
if Mag_End < Mag_Start
    Curr_Dir = -1;
%Else if peak value hasn't changed
elseif Mag_End == Mag_Start
    Timestamp_Start = Timestamp_End;
    Mag_Start = Mag_End;
    PCount = 1;
    Curr_Dir = 0;
%Else peak is positive
else
    Curr_Dir = 1;
end
else
    Cum_Sum_Data(j,1) = C_i + Detrended_Data(j,2);
    C_i = Cum_Sum_Data(j,1);
%Detemine direction of next point
if Cum_Sum_Data(j,1) < Prev_Value
    Next_Dir = -1;
elseif Cum_Sum_Data(j,1) > Prev_Value
    Next_Dir = 1;
end

```

```

%If the next direction isn't flat, is the same as current direction
%and isn't the last point
if Next_Dir ~= 0 && Next_Dir == Curr_Dir && j ~= size(Detrended_Data,1)
    Timestamp_End = Detrended_Data(j+1,1);
    Mag_End = Cum_Sum_Data(j,1);
    PCount = PCount + 1;
    Prev_Value = Cum_Sum_Data(j,1);
%Else if the next direction isn't flat, is the same as current
%direction and is the last point
elseif Next_Dir ~= 0 && Next_Dir == Curr_Dir ...
    && j == size(Detrended_Data,1)
    Timestamp_End = Detrended_Data(j,1);
    Mag_End = Cum_Sum_Data(j,1);
    PCount = PCount + 1;
    Cu_Groups = [Cu_Groups; Timestamp_Start Timestamp_End ...
        Mag_End-Mag_Start PCount Curr_Dir];
%Else if the next direction isn't flat and is not the same as
%the current direction
elseif Next_Dir ~= 0 && Next_Dir ~= Curr_Dir
    Cu_Groups = [Cu_Groups; Timestamp_Start Timestamp_End ...
        Mag_End-Mag_Start PCount Curr_Dir];
    Timestamp_Start = Detrended_Data(j-1,1);
    Mag_Start = Mag_End;
if j == size(Detrended_Data,1)
    Timestamp_End = Detrended_Data(j,1);
else

```

```
Timestamp_End = Detrended_Data(j+1,1);  
end  
Mag_End = Cum_Sum_Data(j,1);  
PCount = 2;  
Curr_Dir = Next_Dir;  
end  
end  
end
```

Appendix E. MATLAB Code: Point_Diff.m

```
% This is a sub function used in the "Main" function. This function computes
% the cumulative distribution function of all the positive peaks from the
% strain gauge data, and finds the point at which the activations are
% significantly greater than previous peaks
%
% Output:
% --Threshold_Value: The magnitude value at which any peak that has a value
% greater than or equal to is considered a brake
% activation
%
% Input:
% --Detrended_Max_Data: Detrended max magnitude for every positive peak

function Threshold_Value = Point_Diff(Peak_Data)

% Calculate the cumulative distribution function
[f,x]=ecdf(Peak_Data);

% Find the index at the 90th percentile
temp = abs(f-0.90);
Index = find(temp == min(temp));

% Find the difference in magnitude from one peak to the peak of the next
% highest value
```

```

Gaps = [];
for i = 2:size(x,1)
    Gaps = [Gaps; x(i) (x(i)-x(i-1))];
end

% Brake activations are greater than this threshold
Threshold = 2*std(Gaps(:,2));

% Starting from the 90th percentile index, find the point at which the
% difference in peak_magnitude is greater than the threshold
Break_Point = [];
i = Index(1,1);
while isempty(Break_Point) && i <= size(Gaps,1)
    if Gaps(i,2) > Threshold
        Break_Point = Gaps(i,:);
    end
    i = i+1;
end

% Find the actual, detrended peak magnitude at this point
if ~isempty(Break_Point)
    Threshold_Value = Peak_Data(find(Peak_Data == Break_Point(1,1)),1);
else
    Threshold_Value = [];
end

```