The Pennsylvania State University

The Graduate School

Graduate Program of Acoustics

TERRAIN CLASSIFICATION FOR CONDITIONED BASED MAINTENANCE

A Thesis in

Acoustics

by

R. Troy Taylor

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2014

The thesis of R. Troy Taylor was reviewed and approved* by the following:

Karl M. Reichard
Research Associate
Assistant Professor of Acoustics
Thesis Advisor

Daniel A. Russell
Professor of Acoustics
Director of Distance Education

Amanda D. Hanford
Research Associate

Victor W. Sparrow
Professor of Acoustics
Interim Head of the Graduate Program of Acoustics

*Signatures are on file in the Graduate School.

Abstract

Reducing the total cost of owning and operating equipment is an important goal for both the U.S. military and private industry. Key goals for logistics support equipment are increasing fuel efficiency and reducing maintenance costs. This research addresses techniques to classify terrain in order to better predict maintenance schedules, reduce subsystem damage and improve fuel efficiency in ground vehicles.

The United States Army Material Systems Analysis Activity (AMSAA) has developed a measurement and processing approach algorithm for Terrain Regime Identification and Classification (TRIC) that classifies the harshness of the vehicle's operating conditions based on the roughness (z-axis motion), bumpiness (magnitudes of vehicle pitch and roll), steepness of the terrain, and the vehicle's speed. The terrain classification can contribute to maintenance prediction algorithms, as rates of damage to the various sub-systems of the vehicle should be correlated to the harshness of terrain traversed. The algorithm uses inputs from sensors measuring vehicle pitch and roll mounted on the vehicle body, vertical acceleration from a sensor mounted on the axle, vehicle speed, and position (via GPS). Spectra from a z-axis accelerometer mounted in the body of the vehicle have been compared to spectra from a z-axis accelerometer mounted on the axle of the vehicle to see if the same information regarding the roughness can be extracted in both cases, or if a

translation model can be developed to produce comparable results, and potentially reduce instrumentation costs.

The TRIC algorithm has been reproduced in MATLAB to see what additional information and uses can be obtained. The classifications from TRIC were compared to vehicle operating mode classifications previously developed by the Applied Research Laboratory (ARL) at Penn State based on vehicle operating data such as engine speed, vehicle speed and accelerator position. Color-coded classifications overlaid on maps of the test courses show operating modes as a function of the vehicle's path. This overlay technique is also useful for comparing vehicle mode classification to terrain classification.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to acknowledge and give thanks to my adviser, Karl Reichard, for giving me the opportunity to do this research, for giving me guidance, and supporting me along the way. I am very appreciative of Jeff Banks for providing funding for my research. I would also like to acknowledge the contributions of others at the Penn State Applied Research Laboratory including Scott Pflumm, Matt Rigdon, Mitch Lebold, Brian Murphy, Jesse Pentzer, and David Fecek. I also want to express my gratitude to Tim Eden at ARL. Though he was not a part of this project, he became a personal mentor to me and provided valuable feedback.

I would also like to thank the students, faculty and staff of the Graduate Program in Acoustics at Penn State for helping me grow as a student and a scientist.

Finally, I would like to thank my wife, Amanda, and two sons, Jadon and Nathan, for being willing to move across the country to support me in my education and for the patience they showed when I could not always provide the attention that they needed and deserved.

This work is supported by Army PEO-SBCT project under Contract Number N00024-12-D-6404 Delivery Order Number 0126. The content of the information does not necessarily reflect the position or policy of the Government or the U. S. Army, and no official endorsement should be inferred.

Chapter 1 - Introduction

1.1 Motivation

Reducing the total cost of owning and operating equipment is important for both the U.S. military and private industry. Key components of this cost for logistics support equipment are maintenance and fuel. There are a number of different approaches to reducing maintenance costs. One area is failure prediction. Different methods have been developed to predict failure in certain vehicle mechanisms to determine where maintenance should be focused [1]. Recently, terrain classification has been proposed as a method for predicting future component failures as part of condition based maintenance (CBM) in addition to the methods that are already in place. Terrain classification could be an important component in diagnostic and prognostic algorithms, as the time that a vehicle spends on various terrains can be directly correlated to damage accumulated in the sub-systems of the vehicle [2]. This research is focused on the classification of the harshness of terrain traversed based on the vehicle's response to the terrain, rather than performing a terrain profile. It will look at a current method of terrain classification and on ways of improving this method and reducing the cost of sensors and cabling needed to collect the required data. In the future, this will help to identify causes of different health degradation in vehicles of the same type or vehicles conducting the same missions. Once enough data is collected, vehicles driven at test centers and in the field can be compared

and analysts will be able to correlate sub-system damage accumulation to terrain traversed. This will help in improving and refining maintenance schedules.

The Army Materiel Systems Analysis Activity (AMSAA) has developed a Terrain Regime and Identification Classification (TRIC) [3] algorithm to classify vehicles' response to the terrain. One motivation of this research is to refine and improve the TRIC algorithm. The research aims to reduce the cost of implementing the algorithm by reducing the number of sensors needed to collect the data required for the algorithm.

Previous work has also classified the operating modes of a vehicle defined by engine speed, vehicle speed and the accelerometer position. Vehicle operational mode information has been correlated to logistics variables such as fuel use, and is being examined for correlation with vehicle failure modes. The advantage of vehicle operational mode classification is that the required data already exists on the vehicle J1939 bus and does not require the addition of any new sensors. It has been proposed to use vehicle operating mode classification to in place of Terrain Classification to predict future maintenance and repair demands. A second motivation of the research is to compare this method of classification to the TRIC classifications to determine if there are any correlations between the two. The purpose of this comparison is to identify if terrain affects how a driver operates a vehicle and how the terrain affects the driver's fatigue levels. If it can be shown that operational modes provide the same information about vehicle wear and tear as information about the terrain over which the vehicles are

operated, then the Army can save money by not having to install the additional sensors, cabling, and data acquisition electronics required to implement the TRIC algorithm.

1.2 Background – Terrain Profiling Methods

A common method to measure the profile of a terrain or test course is using a profilometer like the one shown in Figure 1. The profilometer measures "roughness" by directly measuring the terrain profile from the surface using ultrasonic sensors, lasers and/or gyroscopes [2, 4]. The measurements are made spatially, at normal distance increments, rather than temporally, at normal time increments, as is commonly done in many data collection experiments [4]. A Fourier transform is performed on the data, and a power like spectrum called the wave number spectrum (WNS) is produced, which is analogous to the power-spectrum density (PSD) in the time domain. Some authors interchange the use of WNS and PSD. When used in this thesis, they are interchangeable when referring to data collected and analyzed as a function of distance. By doing an inverse Fourier transform, a test course with the same root-mean-square (RMS) value as the real course could be digitally synthesized for use in computer terrain modelling [4].

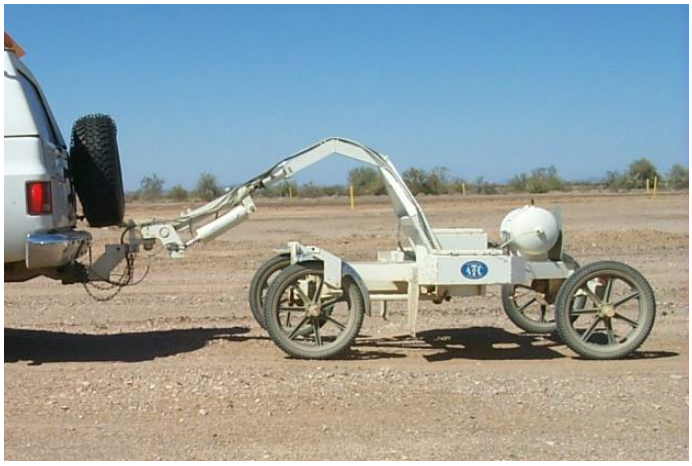

Figure 1: Profilometer being towed by vehicle Source: Hatton, K. (2011, March 2). ***Terrain Regime Identification and Classification (TRIC)***

The WNS can also be used to analyze the condition of the road by comparing the terrain's RMS to a known classified course with the same RMS. The problem with this method, as shown by Steinwolf and Connon, is that two courses that have the same RMS value can vary greatly in their terrain features and can produce different WNS [2], [4]. Figure 2 shows the WNS of two different courses that have the same RMS value, but different terrain features. The blue curve represents one course, the green course represents the other, and where they overlap is shown in purple.

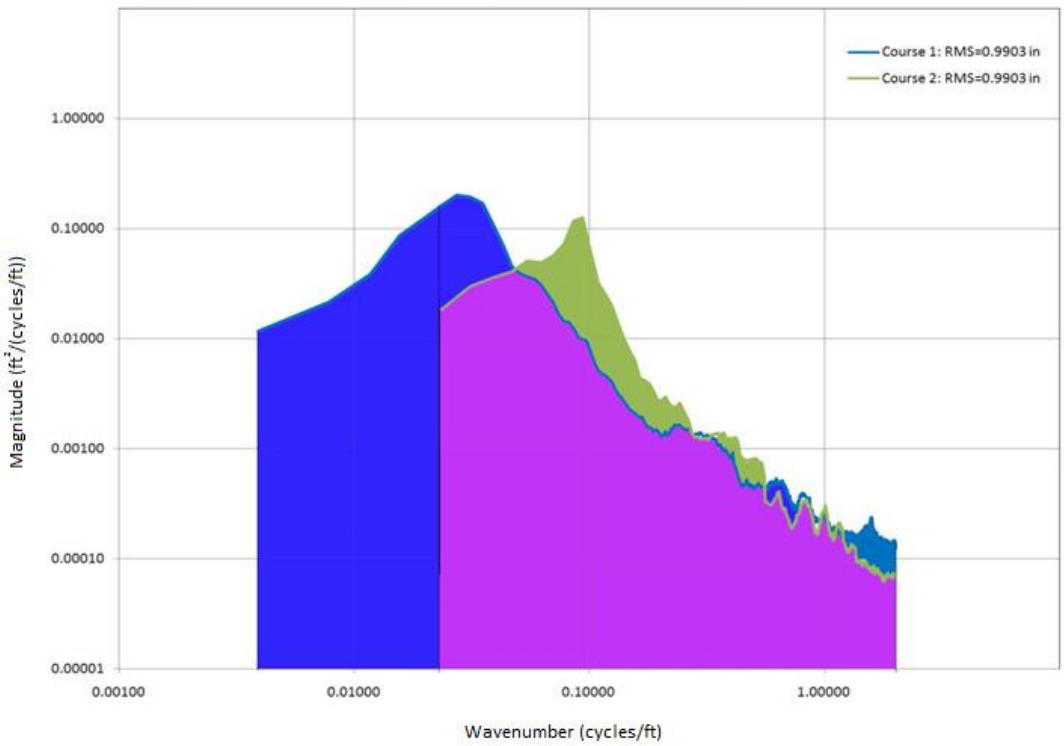

Figure 2: The WNS of two courses with identical RMS values, but very different terrain features (AMSAA) Source: Hatton, K. (2011, March 2). ***Terrain Regime Identification and Classification (TRIC)***

The two curves have completely different shapes, indicating different spatial distributions of the terrain effects on the vehicle. This could mean that one terrain has several small bumps and the other has only a few larger bumps. The green curve is even missing a large section of the long wavelength data because this method often filters out some of that data, or doesn't even account for it (since the spectrum is plotted with the logarithm of the wavenumber, the missing long wavelength data is most likely the result of a difference in the wavenumber resolution of the respective Fourier transforms). Loss of this data results in a loss of information about damage to some subsystems [2]. Different types of terrain can have different effects on the vehicle when it comes to damage accumulation.

For example, a vehicle that spends a lot of time on steep but smooth terrain may acquire more damage to the engine and transmission and have worse fuel economy than a vehicle that spends a lot of time on bumpy, flat terrain, which may accumulate more damage to the suspension and steering. The two terrains could have identical RMS values. Another problem is that the profilometer is large, expensive, and must be towed, making it difficult for in-operation vehicles to collect the necessary data [3].

A popular method for classifying terrain roughness is the roughness index from the International Organization for Standardization (ISO) [5]. The index uses the WNS, or the PSD[1], to classify the terrain on an A-H scale where A is excellent condition and H is very poor condition. The index can be seen in Figure 3.

[1] For constant velocity the WNS and the PSD of the time series data are the same with a simple scaling of the frequency axis.

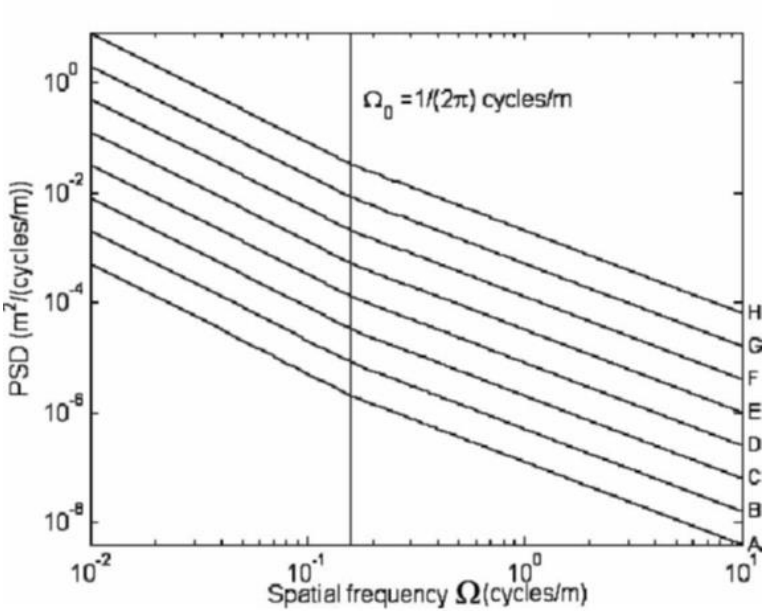

Figure 3: Road Roughness classification index from the ISO, where Ω is the wavenumber, also known as the spatial frequency Source: González, A., O'Brien, E., Li, Y. and Cashell, K. (2008, June). The use of vehicle acceleration measurements to estimate road roughness. ***Vehicle System Dynamics.***

A recent method in terrain profiling suggested by Gonzalez involves using the accelerometers already installed on many luxury vehicles [5]. The PSD from the unsprung accelerometers is obtained from exciting a vehicle with a theoretical known profile and with a known PSD. A transfer function between the road and the unsprung accelerometers is then obtained using Equation (1.1).

$$H(\Omega) = \frac{\mathrm{PSD}_{acc}(\Omega)}{\mathrm{PSD}_{road}(\Omega)} \tag{1.1}$$

where the numerator is the PSD of the unsprung accelerometers and the denominator is the theoretical PSD of the road.

The transfer function can then be applied to a PSD obtained from the accelerometers of that vehicle to determine the PSD of any road the vehicle is driven on. From the road PSD, the road can be given a roughness classification according to the ISO standards [5]. However, a terrain profile cannot be obtained from this method because when a PSD is generated, phase information is lost. To reconstruct a time series from a spatial frequency series, the phase information is needed because it contains the temporal content. Thus, the terrain elevation information cannot be backed out from the PSD of the road.

Another method of terrain profiling makes use of force measurements in addition to motion measurements used by other methods. This method was developed by the Nevada Automotive Test Center (NATC) for use on a test vehicle they developed called the Dynamic Force Measurement Vehicle (DFMV) [6]. A triaxial load cell is placed at the end of each axle of the DFMV. This load cell makes sure that the forces on one side are equal and opposite of the forces on the other side of the axle. The case shown in Figure 4 has $F1 = F2$, where $F1$ is the sum of all the forces on the left side of the axle (forces from the sprung mass, which is the vehicle body and vehicle load, and the suspension, represented by the sprung dashpot) and $F2$ is the sum of all the forces on the right side of the axle (the tire). This will make sure that vibrations from the suspension or the body of the vehicle are isolated from the accelerometer located on the tire side of the load cell. The accelerometer is then only measuring the vertical displacements from the tire's interaction with the ground. The accelerometer data can be integrated twice to obtain vertical displacement (elevation), and then the elevation can be plotted against distance travelled

[6]. The accelerometer measurements are taken at all four tires and samples are taken at equal intervals of distance, not time, just like the other methods of profiling. Then, like other methods, a WNS (or PSD) can be taken from the accelerometer data which can be used for comparison to other courses or to digitally synthesize a course.

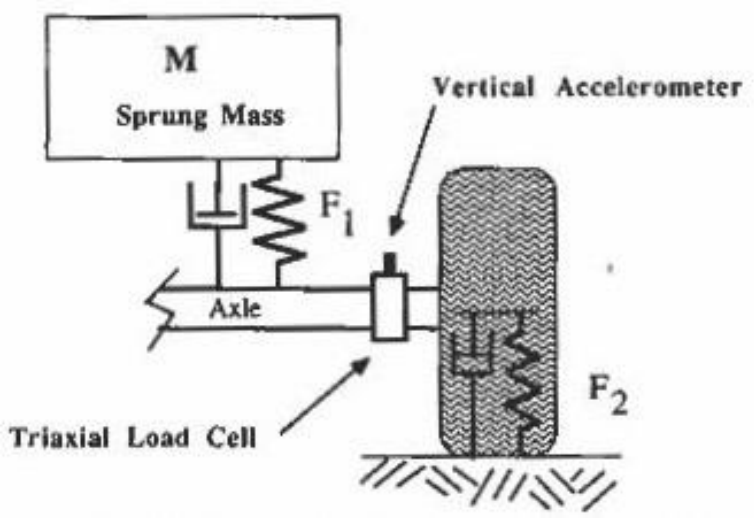

Figure 4: Quarter car model for the DFMV showing the locations of the load cell and the accelerometers, as well as showing the locations of the forces that are balanced out Source: Ashmore, S. and Hodges, H. (1990, October 28). ***Dynamic Force Measurement Vehicle (DFMV) and Its Application to Measuring and Monitoring Road Roughness.***

Currently, there is no standard that is widely used to classify terrain harshness. In addition to the ISO classification discussed above, one system used by the Army classifies terrain using a profilometer, like the one described above. The terrain is classified as primary, secondary and off-road. The classifications are based on the elevation profile of the terrain. The problem with the classification system is that the amplitudes that define each regime heavily overlap with the others [3]. The current methods are most concerned with improving road conditions or synthesizing courses digitally to use for testing. They

are also concerned with how vehicles perform on certain terrain profiles. However, they are usually not concerned with how and where on the vehicle the terrain causes wear or on how the terrain affects overall fuel economy. Current methods do not address classifying terrain across a broad spectrum of wavelengths (types of terrain) in real time or in identifying what kinds of terrain vehicles are operating on. Another concern with profiling methods is that the data is always post-processed and cannot be done in real-time in the field.

To help solve these problems, AMSAA developed the TRIC algorithm. The goal of TRIC is to provide a terrain harshness classification system in real time that includes medium and long terrain wavelengths as well as short wavelengths for vehicles deployed in the field. Because the TRIC classification is based on the vehicles' response to the terrain, it creates a classification system that is more focused on vehicle wear and in determining the amount of time spent on a particular terrain. This gives more information than a system that just profiles the terrain or that just classifies the short wavelengths – the "roughness." It is designed to be used in the field by in-operation vehicles as well as on test courses.

1.3 Thesis Outline

Chapter Two of this thesis gives a detailed explanation of the TRIC algorithm and examines the methods used to reproduce the TRIC algorithm based on data provided by AMSAA from runs done on test courses by two different vehicles. The results of the reproduction are discussed.

Chapter Three discusses the possibility of using an accelerometer placed above the suspension inside the body of the vehicle rather than below the suspension on the axis. This would allow for a less expensive setup when gathering data required to classify terrain harshness. Data for this analysis was obtained from vehicles with accelerometers placed on the axle and above the suspension in the body of the vehicle and which were driven over test courses in Nevada. Chapter Four discusses a method for classifying vehicle operational modes using a k-means clustering algorithm. Results of the algorithm applied to data from test runs done on courses in Nevada and from vehicle runs done in State College, Pennsylvania are presented. Chapter Five presents a method for correlating the TRIC classifications to the operational mode classifications. Results from the correlations done on the same data used in Chapter Four are presented. Finally, Chapter Six summarizes the research and results of the analysis. Suggestions for future work and continuation of this research are also presented.

Chapter 2 - Reproduction of the TRIC Algorithm

2.1 AMSAA Terrain Regime Identification and Classification (TRIC)

The TRIC algorithm divides the terrain into short wavelengths, medium wavelengths, and long wavelengths. Each of these wavelength bands are classified into low amplitude, medium amplitude, and high amplitude. This gives 27 different classifications of terrain harshness, as shown in the cube in Figure 5.

The short wavelength terrain can be thought of as repetitive small bumps in the terrain that causes vertical displacement and vibration in the vehicle. Short wavelength terrain measurements are made from a z-axis accelerometer, which measures the acceleration of the up and down motions of the vehicle. It ranges from smooth terrain, such as a paved road, to rough terrain, such as a rocky road or a Belgium block style of road, like the roads shown in Figure 6. The definition of smooth and bumpy depends on the type of vehicle and is discussed more in Section 2.3. On the vehicle, it can affect the condition of the fastened joints, the electronics and human comfort and safety.

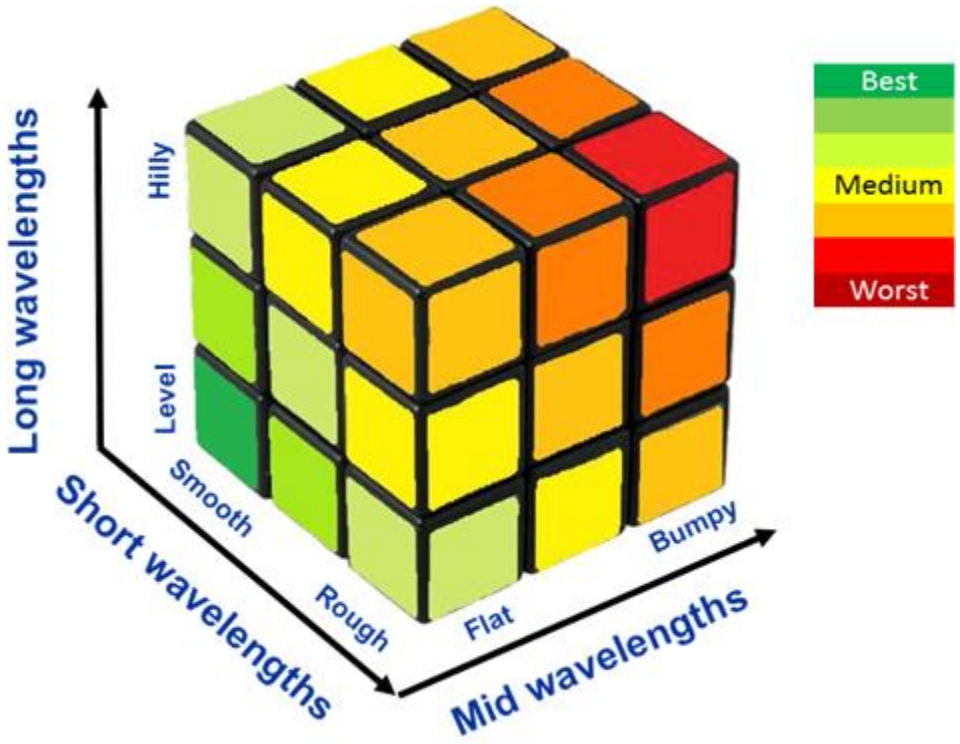

Figure 5: Cube showing the 27 different levels of harshness that the TRIC algorithm classifies Source: Hatton, K. (2011, March 2). ***Terrain Regime Identification and Classification (TRIC)***

Figure 6: Example of short wavelength terrain with smooth terrain on the left and rough terrain on the right Source: Hatton, K. (2011, March 2). ***Terrain Regime Identification and Classification (TRIC)***

The TRIC algorithm places the accelerometer on the axis of the vehicle so that it is below the suspension. The data is collected at a sampling frequency of 60 Hz – 100 Hz (or is decimated down to that frequency) and a low-pass filter of 25-30 Hz is applied. Then, the RMS values of the acceleration data are determined for 20 second blocks of time as well

as the average speed over the same 20 second blocks of time. These values are plotted against each other and speed and vehicle-specific thresholds are applied to classify each point of averaged data as low, medium or high amplitude. Low amplitudes are assigned a value of 1, medium amplitudes are assigned a value of 2 and high amplitudes are assigned a value of 3. Figure 7 shows the flow diagram of the short wavelength portion of the algorithm.

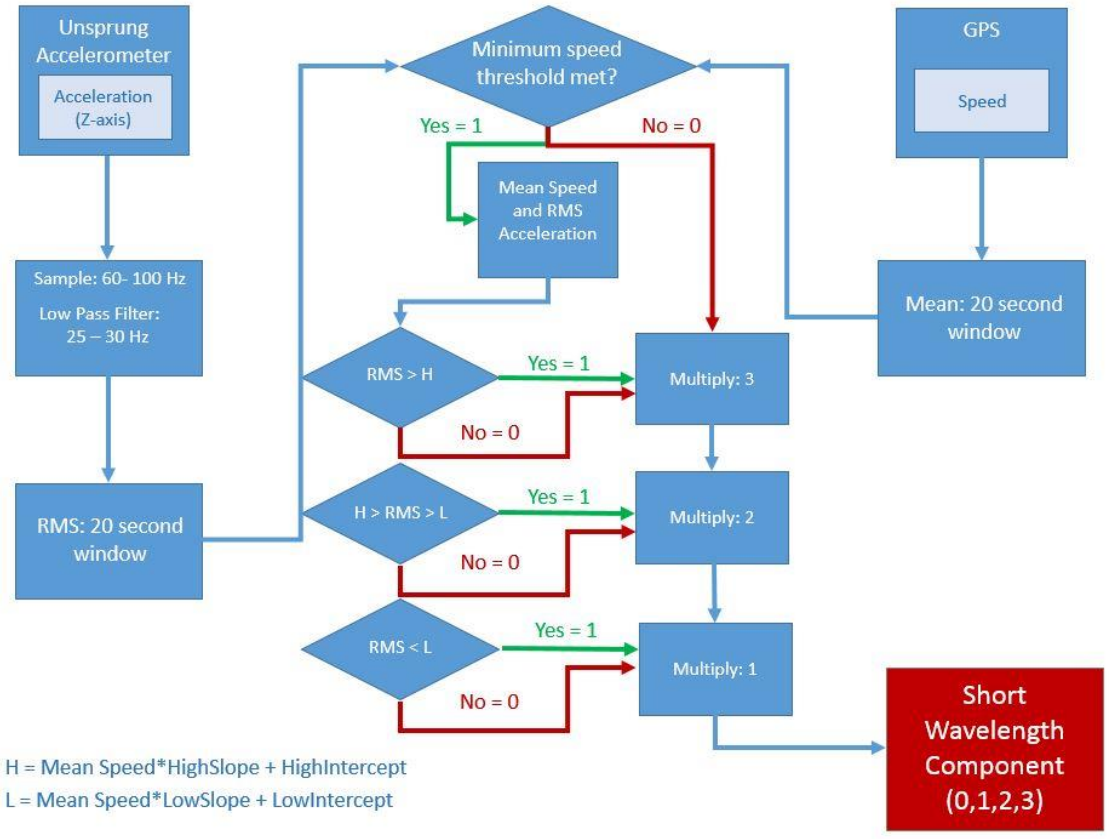

Figure 7: Flow diagram of the short wavelength portion of the TRIC algorithm, the thresholds HighSlope, HighIntercept, LowSlope and LowIntercept are specific to each vehicle

Next, the medium wavelength terrain can be thought of as larger repetitive bumps than the short wavelength data, and may impart a noticeable rocking motion instead of simple

vertical acceleration. The measurements are made from a gyroscope, but only the pitch and roll measurements are needed for the algorithm; yaw is not used. The definitions of pitch, roll and yaw are shown in Figure 8.

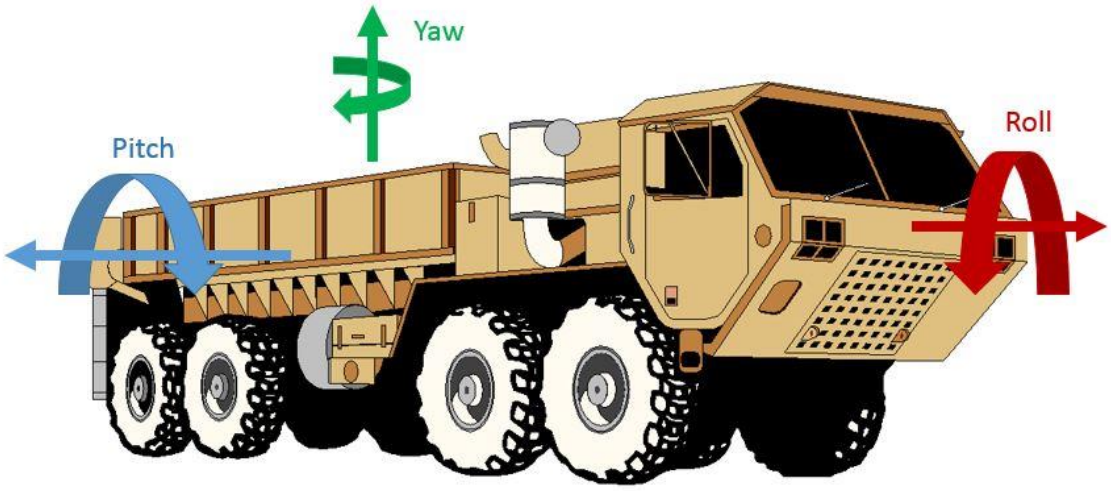

Figure 8: Pitch is defined as rotation around the axis running from the left to the right of the vehicle, roll is defined as rotation around the axis running from the front to the back of the vehicle and yaw is defined as rotation around the axis running from the top to the bottom of the vehicle Source: http://wmfclipart.com/images/Army/10-ton-truck.WMF.html, obtained, arrows added by author

Terrain ranges from flat terrain such as a paved road (the same as the short wavelength data) to bumpy terrain such as a path containing large boulders as shown in Figure 9. The definition of what is flat and what is bumpy is determined by the specific vehicle being tested. The suspension and steering of the vehicle can be worn by this type of terrain and it also affects the stability of the vehicle as well as human comfort.

Figure 9: Example of medium wavelength terrain with flat terrain on the left and bumpy terrain on the right Source: Hatton, K. (2011, March 2). ***Terrain Regime Identification and Classification (TRIC)***

The pitch and roll data is taken from the gyroscope, and the difference between each sample is taken giving $\Delta Pitch$ and $\Delta Roll$ values. Then a pitch and roll vector magnitude is computed, which represents the rate of angular change, using a standard vector magnitude formula as shown in Equation (2.1).

$$|PR| = \sqrt{\Delta Pitch^2 + \Delta Roll^2} \tag{2.1}$$

where $|PR|$ is the pitch and roll vector magnitude. It is assumed that the faster the angular rate of change, the bumpier the terrain [3].

Once the vector magnitude is calculated, the mean is calculated over 20 second periods and vehicle specific thresholds are applied to determine low, medium and high amplitude. The thresholds are note speed dependent. Low amplitudes are assigned a value of 10, medium amplitudes are assigned a value of 20, and high amplitudes are assigned a value of 30. Figure 10 shows the flow diagram for the medium wavelength classifications.

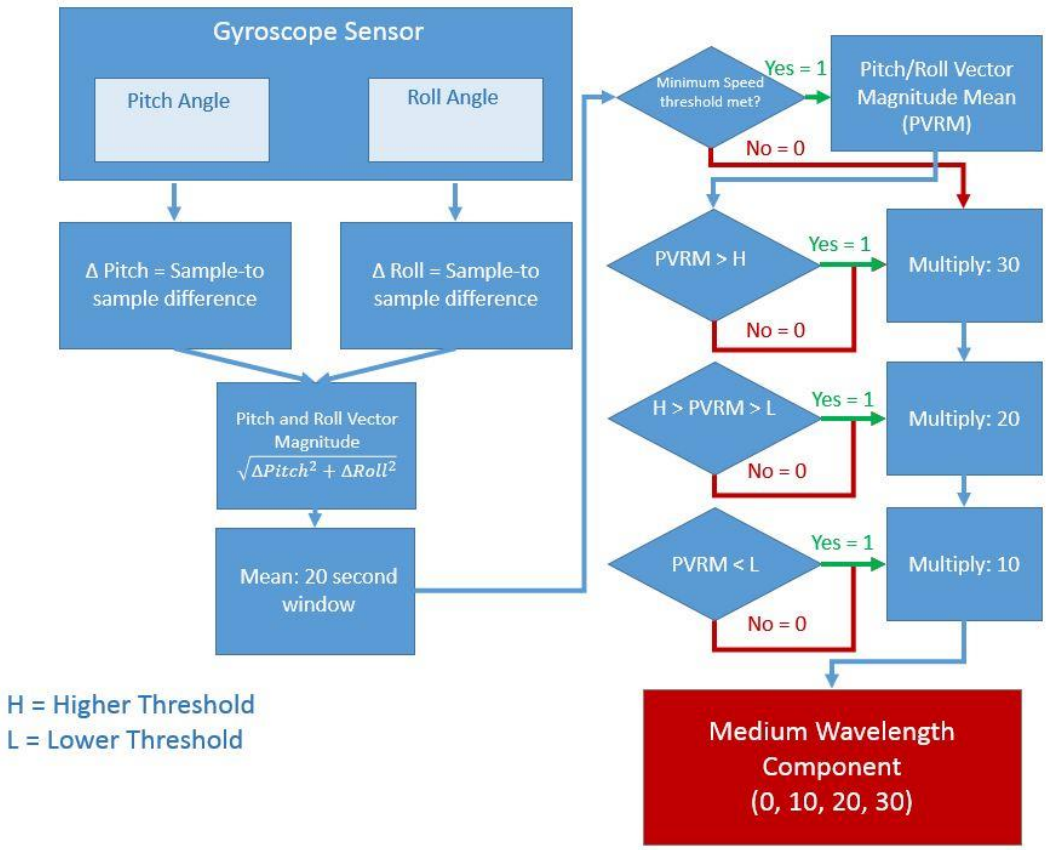

Figure 10: Flow diagram of the medium wavelength portion of the TRIC algorithm, the higher and lower thresholds are specific to each vehicle

Finally, the long wavelength terrain is defined as the hilliness of the terrain. The data is taken from the speed and altitude measurements from a GPS device mounted on the vehicle. The terrain is classified from flat to hilly, as seen in Figure 11. Terrain is considered flat whenever the percent grade is less than 2% and is considered hilly or steep when the percent grade is greater than 8%. These classifications are independent of vehicle type. Large, steep elevation changes in the terrain can affect the life of the engine, transmission and brakes of the vehicle. It can also negatively affect the fuel economy.

Figure 11: Example of long wavelength terrain with flat terrain on the left and hilly terrain on the right Source: Hatton, K. (2011, March 2). ***Terrain Regime Identification and Classification (TRIC)***

The vehicle speed data is taken from the GPS and numerically integrated to obtain the distance traveled. Change in distance over 20 second periods are calculated. The altitude data is also taken from the GPS and the 20 second change in altitude values are calculated. The change in altitude divided by the run gives the percent grade. The run is not the same as the distance traveled but can be calculated with simple geometry shown in Figure 12 and Equation (2.2). The thresholds are applied to obtain low, medium and high amplitude harshness values. The low amplitudes are assigned a value of 100, the medium amplitudes are assigned a value of 200 and the high amplitudes are assigned a value of 300. The flow diagram for the long wavelength classifications can be seen in Figure 13.

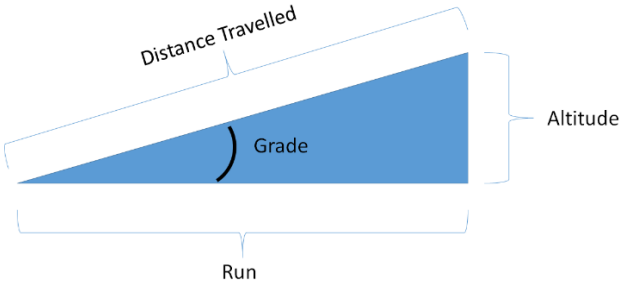

Figure 12: Simple geometry diagram to relate distance travelled, the run, and the altitude

$$Grade = \frac{Altitude}{\sqrt{Distance\ Travelled^2 - Altitude^2}} \tag{2.2}$$

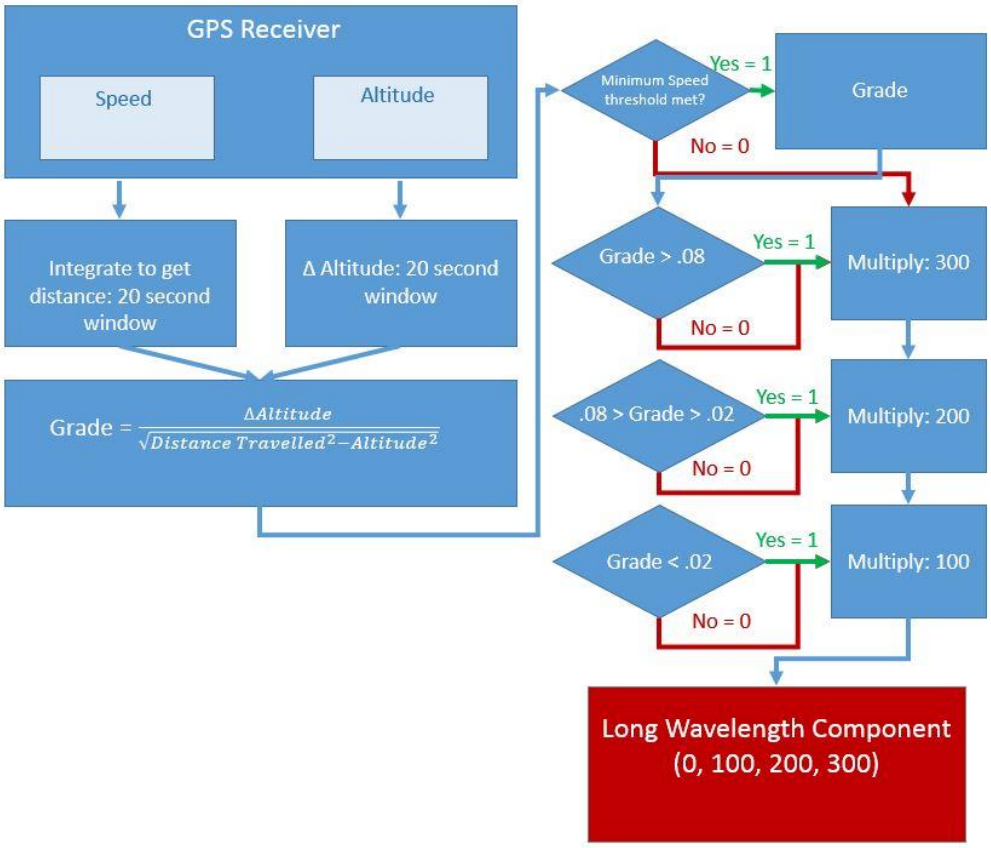

Figure 13: Flow diagram of the long wavelength portion of the TRIC algorithm

Once the harshness values for each classification of wavelength are determined, they are added together to obtain the overall harshness of the terrain. The final portion of the TRIC algorithm flow diagram is shown in Figure 14.

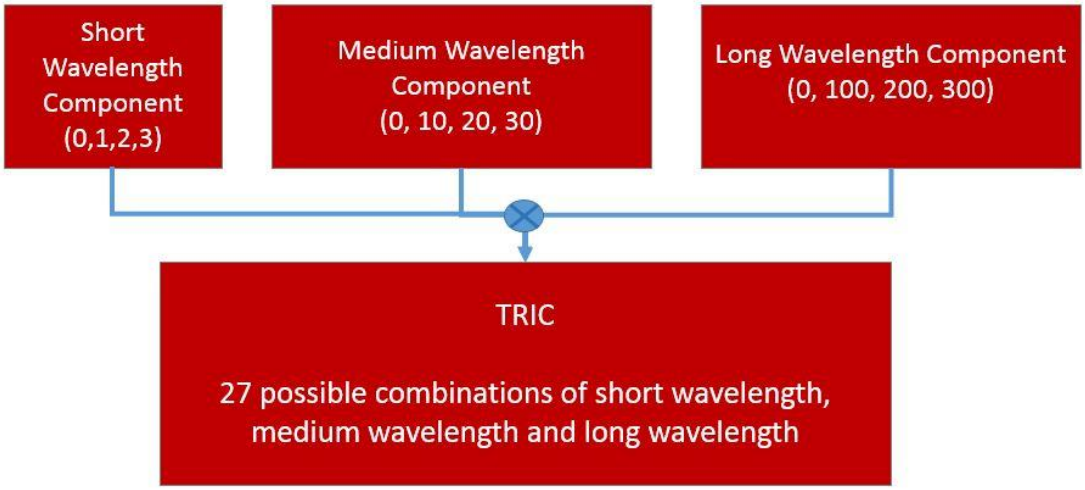

Figure 14: Final portion of the flow diagram for the TRIC classifications

Once the values are added together, there are 27 different combinations of short, medium and long wavelength classifications. It is also important to mention that there is a minimum speed threshold of 5 mph. Any time that the average speed over a 20 second period is below 5 mph, a value of 0 or a classification of "idle" is given, which gives 28 different possible harshness values. The values can be color-coded and overlaid on a map of the course in order to give a visual representation of the harshness of the terrain. Examples of such maps are shown later in Section 2.3. In order to not bog down the map with 28 different colors, "degenerate" TRIC classifications are used, as shown in the cube in Figure 5, to reduce the total number of colors down to seven. An eighth color of black is used to represent the idle classification.

As a note, the thresholds for the each vehicle were obtained by AMSAA using test courses with known features. Vehicle responses were recorded for certain sections on various courses at the Aberdeen Proving Ground and Yuma Proving Ground. These sections have been classified as primary, secondary, and off-road for the roughness (corresponding to low, medium and high amplitudes in the TRIC algorithm). A similar approach was used for the bumpiness where certain sections had features that caused different degrees of pitch and roll on the vehicle. The thresholds were set to match the classifications on the courses. Such a system was not needed for hilliness, or long-wavelength. For all vehicles, less than a 2% grade is considered flat or low steepness, a 2% to 8% grade is considered medium steepness, and anything greater than an 8% grade is considered high steepness [3].

Because the thresholds are based on vehicle response to the terrain, they are different for each vehicle, because each vehicle has a different response to terrain. The exception is the long wavelength classifications, being because independent of vehicle size or type, the steepness of the terrain will not change. For the short wavelength, the weight of the vehicle and the type and size of the tires are two variables that will affect the measurements at the accelerometers. For the medium wavelength, the size of the vehicle, specifically the wheelbase, will affect the thresholds. A larger vehicle with a larger wheelbase will not have the same pitch and roll magnitude (angular rate of change) as a smaller vehicle with a smaller wheelbase. A method being developed could convert a PSD in the time domain to a WNS in the spatial domain. The WNS for vehicles of different

sizes could then be compared and thresholds could be determined from the comparisons rather than relying on running different size vehicles over test courses with known profiles multiple times. The WNS could also potentially be used to define the division between medium and long wavelengths for each vehicle. This method is briefly discussed in Appendix A.

Important factors that affect the vehicle's response to the terrain are tire pressure and the weight of the load. AMSAA performed several tests using vehicles with different tire pressures and different load weights. Difference in the data from the sensors used for TRIC were compared, and thresholds were determined that resulted in the lowest errors across all the tests [3]. New thresholds could be determined for different weight loads and tire pressures, but with so many variations and combinations possible, a huge database would need to be produced for the algorithm to be implemented in the field.

2.2 Methods

Data supplied to ARL by AMSAA included the information from several military vehicle runs on various test courses. Four data sets were used involving four different courses. The two Army vehicles used in the tests were the Oshkosh Heavy Expanded Mobility Tactical Truck (HEMTT) and the Oshkosh Family of Medium Tactical Vehicles (FMTV). The HEMTT, an eight-wheeled vehicle, diesel-powered, with off-road capabilities, is primarily used to transport heavy cargoes [7]. The FMTV, a six-wheeled all-terrain vehicle, is also primarily used for a variety of combat missions including troop transport, resupply, and hauling [8]. The two vehicles are shown in Figure 15. The tests took place

on the Churchville, Perryman and Belgian Block courses at the Aberdeen Proving Ground in Maryland and on the road between the Churchville course and the Perryman course. The data included raw data from an unsprung accelerometer, the pitch and roll data from a gyroscope, and the speed and altitude data from a GPS. The data also included the filtered, 20-second RMS unsprung accelerometer data, the 20 second average of the pitch and roll vector magnitude and the 20-seond grade of the slope.

Figure 15: Above: the Oshkosh HEMTT Below: the Oshkosh FMTV Source: oshkoshdefense.com, May 2014

The first task of this research was to reproduce the TRIC algorithm in MATLAB. The raw data from AMSAA was used as the basis for reproducing the algorithm. The raw data was

passed into the algorithm written in MATLAB to obtain the 20 second values. If the time length of a data set was not evenly divided into 20 second periods (meaning there would be a remainder), the average of the remainder was taken for the final data point. These values were then compared to the AMSAA 20-second averages with the expectation that they should be the same. The MATLAB code for the algorithm can be found in Appendix C.

An immediate challenge encountered when reproducing the algorithm was whether to do a sliding average of 20 second periods with no overlapping, or to do some percentage of overlapping in the average. Examination of the AMSAA averages showed that an average with no overlapping was done. This was determined by the number of points in the data after averaging. With this information, the AMSAA algorithm could be reproduced. A more accurate picture of the terrain harshness could be obtained by averaging with an overlap between the 20 second periods. A future version of the algorithm may include averaging done with overlapping to better represent the terrain.

After it was determined that no overlapping was used in the averaging, the number of data points in the reproduced averages did not always match the number of points in the AMSAA averages, sometimes they were off by one point. When the number of data points was off after averaging, it was always that the AMSAA averages had one more data point. It was unclear why this was the case.

Another problem was that the data averages from the reproduced algorithm were not the same as the corresponding values from the AMSAA averages. The reason for the

difference is not known. Because of this, the TRIC classifications did not match between those produced by the reproduced algorithm and those supplied by AMSAA. Table 1 shows the error percentage between the two sets of classifications for the four tests.

Table 1: Percentage of classification points that matched between the reproduced TRIC algorithm and the classifications supplied by AMSAA. Each of the four courses are listed with the vehicle used to collect the data.

Perryman, FMTV (Loaded)			**Perryman to Churchville, HEMTT**		
SWL Match (%)	MWL Match (%)	LWL Match (%)	SWL Match (%)	MWL Match (%)	LWL Match (%)
100	89.7	99.5	96.7	86.81	85.71
Churchville, HEMTT (Unloaded)			**Belgian Block, HEMTT**		
SWL Match (%)	MWL Match (%)	LWL Match (%)	SWL Match (%)	MWL Match (%)	LWL Match (%)
62.84	88.51	33.1	74.22	73.19	79.38

The long wavelength classifications for the Churchville HEMTT test were different from the AMSAA classifications. This is because the AMSAA classifications did not use the absolute value of the slope. This means that any negative slope would be classified as low amplitude. If we assume that negative slopes will cause as much wear on the vehicle as corresponding positive slopes, the absolute value of the slope should be used when calculating the long wavelength classifications. There were some points in this test that were classified by AMSAA as having slopes with a percent grade above 30%, which are not present on the Churchville course [9]. The other three tests did not show as much error in the long wavelength classifications because they are flatter courses and so using the negative slopes did not have as much of an effect.

A representative from AMSAA was contacted to address the differences of the reproduced classifications and the AMSAA classifications. The received response was to not trust the provided classifications, to process the data through the reproduced algorithm, and use those classifications [10]. Based on the response, it was decided to continue analysis with the reproduced algorithm and not pursue causes for difference any further.

In the following section, the process of the TRIC algorithm is shown, as an example, using raw data from the Churchville test put through the reproduced algorithm.

2.3 Algorithm Reproduction Analysis Using AMSAA Data Sets

2.3.1 Churchville Course with HEMTT vehicle

The Churchville Test Area is located at the Aberdeen Proving Grounds in Maryland. The course used for this test can be seen in Figure 16. The course consists of dirt roads with steep inclines and tight turns. It is designed to stress a variety of the vehicle subsystems including the engine, drivetrain and suspension [10]. This type of terrain was selected to give a full spectrum of harshness classifications for all three wavelength types in the TRIC algorithm.

Figure 16: Left: Churchville Test area where the black line represents the path taken for this test Right: Sample Image of terrain and road of the course

Figure 17 shows the map overlaid with the path of the vehicle, color coded with the TRIC classifications, and shows the overall harshness. Dark green shows the sections of the course that are smooth, flat and level. Red shows the harshest sections of the course, which are rough, bumpy and steep. On this course, the other colors show that there is every classification of terrain in between the two extremes.

The maps can be broken down further to show the amplitudes of just the short wavelengths, the medium wavelengths, or the long wavelengths. These maps can be useful if one is just looking for the harshness of a certain type of terrain. Figures 18 through 20 show these maps for the short, medium and long wavelengths respectively.

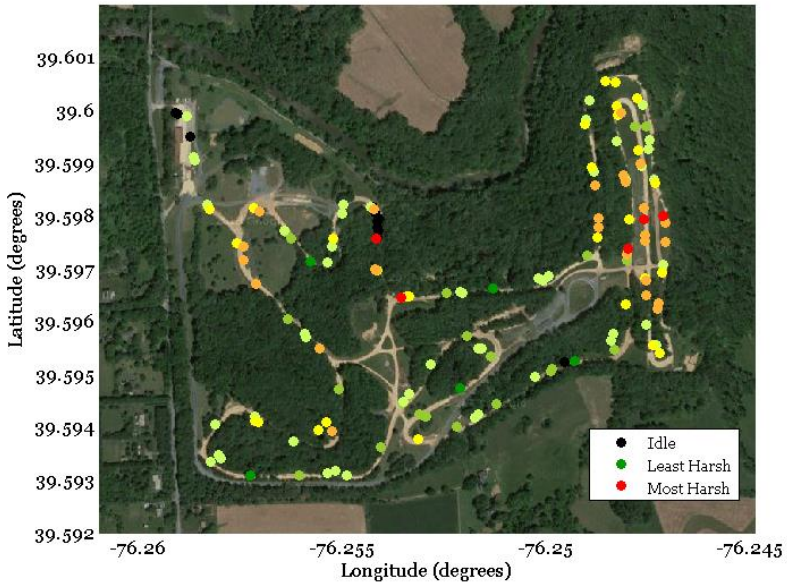

Figure 17: Map of the Churchville course overlaid with the terrain classifications obtained from a HEMTT vehicle

The maps can be broken down further to show the amplitudes of just the short wavelengths, the medium wavelengths, or the long wavelengths. These maps can be useful if one is just looking for the harshness of a certain type of terrain. Figures 18 through 20 show these maps for the short, medium and long wavelengths respectively.

Figure 18, the short wavelength map, shows that the course is mainly smooth. The eastern (right) section of the course is a little rougher with a small section near the middle that is very rough. Figure 19, the medium wavelength map, shows that the course has a medium level of bumpiness throughout, with only small sections that are flat or very bumpy. And finally, Figure 20, the long wavelength map, shows that the course is not very level. Several sections had a percent grade between 2% and 8% while several other sections had a percent grade above 8%.

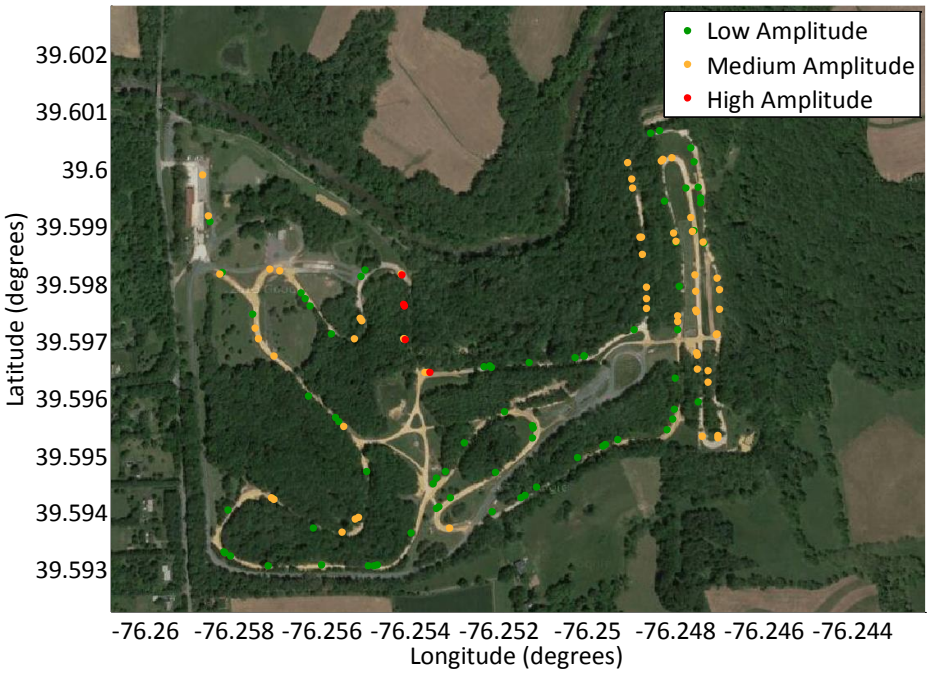

Figure 18: Short wavelength classifications for the Churchville course driven by a HEMTT

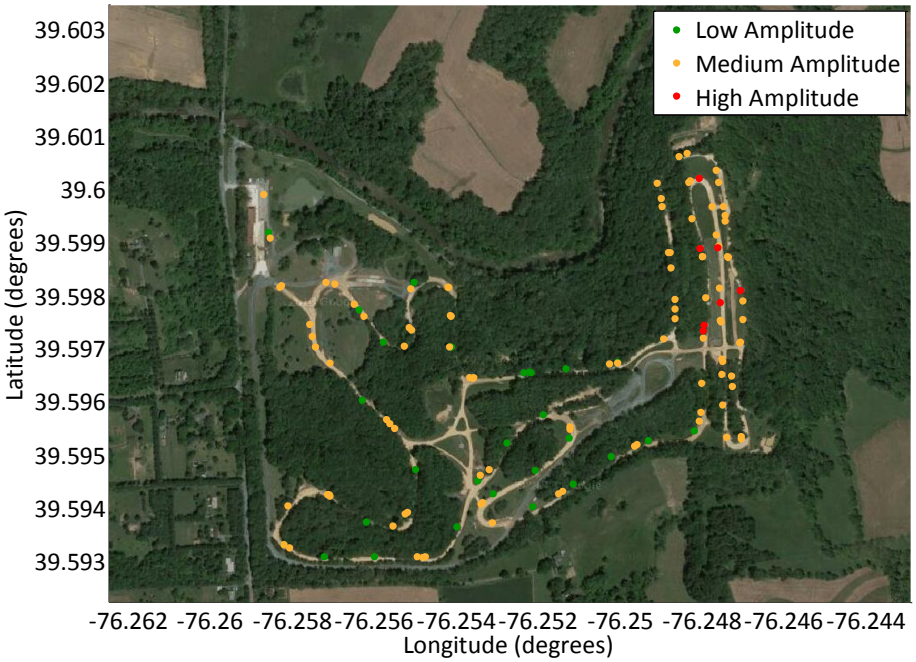

Figure 19: Medium wavelength classifications for the Churchville course driven by a HEMTT

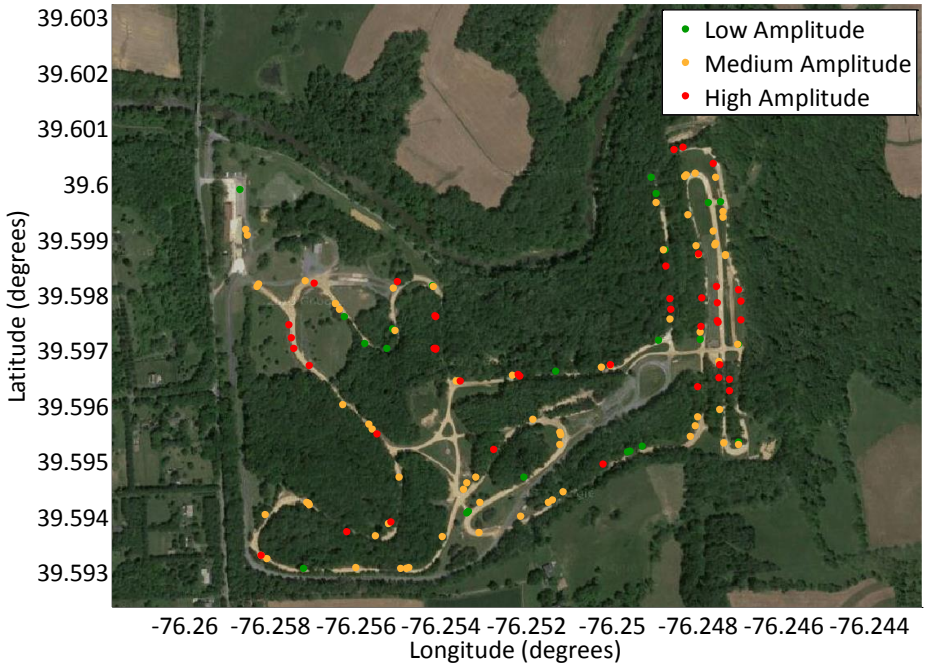

Figure 20: Long wavelength classifications for the Churchville course driven by a HEMTT

Figure 21 shows scatter plots of the short, medium and long wavelength averaged data from the reproduced algorithm. The maps are more useful in visualizing the course, but these scatter plots can be more useful in visualizing how fast the vehicle is travelling over certain types of terrain. In all three wavelength regimes, the data is plotted against speed, but only the short wavelength classification is dependent on speed. The threshold lines are shown as solid lines on the plot. These thresholds were determined experimentally by

AMSAA and are specific to the HEMTT[2]. The short wavelength upper and lower threshold lines are given respectively by the equations:

$$a = .0325s^{-1} * \mathrm{S} - .105g \tag{2.3}$$

and

$$a = .01s^{-1} * \mathrm{S} + .026g \tag{2.4}$$

where *a* is the z-axis acceleration in units of *g's*, S is vehicle speed, and where a unit of *g* is equal to $9.8\frac{m}{s^2}$. The medium wavelength upper and lower thresholds are given respectively by $|PR| = .09095°$ and $|PR| = .03570°$, where $|PR|$ is the pitch and roll vector magnitude in units of degrees. The long wavelength upper and lower thresholds are given respectively by $Grade = .08$ and $Grade = .02$, where *Grade* is the grade of the slope and is unitless. The grade multiplied by 100 gives the percent grade. For any wavelength, if the vehicle speed is below 5 mph, the point is classified as idle and is not given a low, mid or high amplitude classification. The data points are color-coded according to their TRIC

[2] The long wavelength thresholds are the same regardless of the type of vehicle. The short wavelength threshold will depend on the dynamics of the vehicle, and will be different for each vehicle. The medium wavelength thresholds should change based on vehicle size because mid-sized bumps on the terrain will be experienced differently by vehicles with different wheelbase sizes. See Appendix A.

classification. Green data points represent low amplitude, orange data points represent mid amplitude and red data points represent high amplitude.

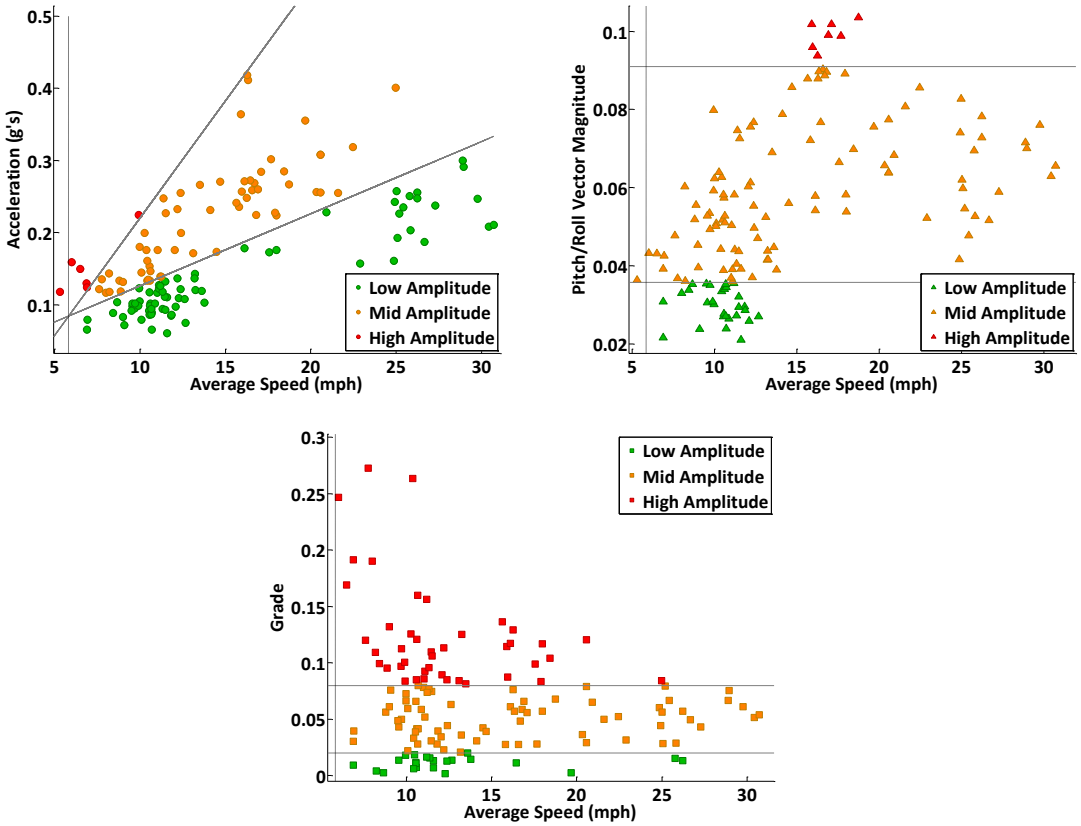

Figure 21: Top Left: Reproduced short wavelength data; Top Right: Reproduced mid wavelength data; Left: Reproduced long wavelength data

2.3.2 Summary of Reproduction Results

Despite the differences between the classifications from the reproduced algorithm and the classifications provided by AMSAA (see Table 1), it was determined that the reproduced algorithm followed the TRIC algorithm flow diagrams shown in Figures 7, 10, 13 and 14, and could be used for the remainder of this research. The reproduced classifications were determined to be more accurate based on a few errors in the AMSSA classifications,

specifically not using the absolute value of the slope and having some percent grade values above 30%, which are not present on the Churchville course.

An example of the TRIC algorithm was shown using raw data supplied by AMSSA from a HEMTT operating on the Churchville test course. A map was produced which displayed the overall terrain classifications as well as classifications for the short, medium and long wavelengths. Scatter plots were also shown that displayed the TRIC classifications in relation to the thresholds. They also showed how terrain was classified at certain speeds.

Chapter 3 - Comparison of Sprung and Unsprung Accelerometers

3.1 Background

To record the data necessary for the TRIC algorithm, AMSAA developed a Health and Usage Monitoring System (HUMS) called the System Health and Reliability Computer (SHARC), shown in Figure 22.

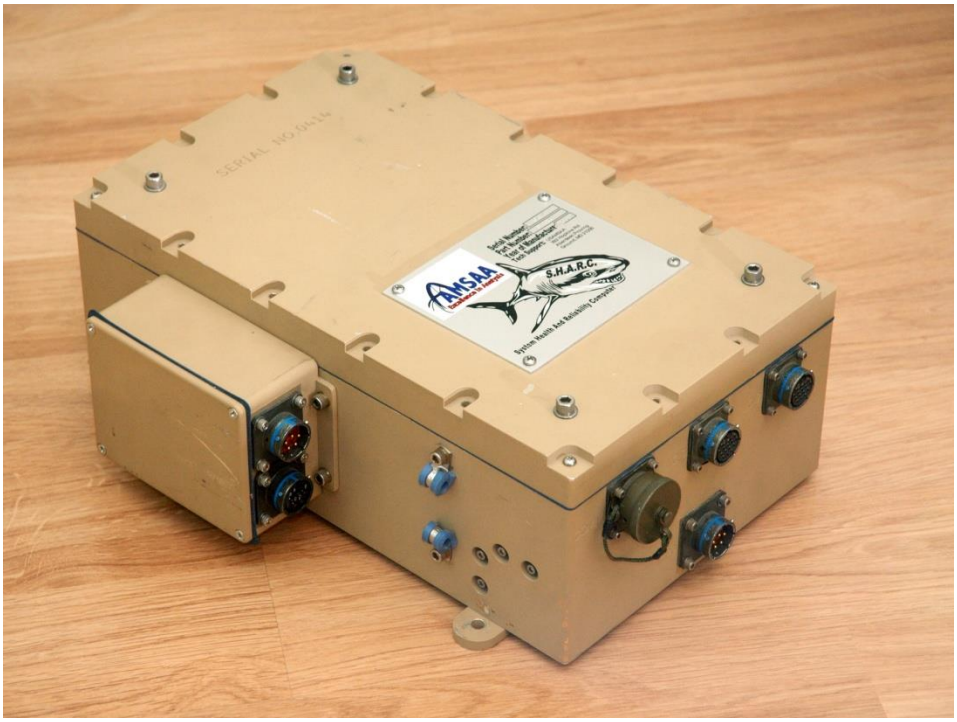

Figure 22: System Health and Reliability Computer (SHARC) used by AMSAA to collect the data required for the TRIC algorithm Source: Hatton, K. (2011, March 2). ***Terrain Regime Identification and Classification (TRIC)***

The SHARC includes all the sensors required for the TRIC algorithm, including GPS, a triaxial accelerometer and a triaxial gyroscope. The SHARC also includes a connection to the vehicle bus capable of monitoring the engine, transmission, braking system and other electronic systems in the vehicle, as well as inputs for four analog sensors. The various sensors allow the SHARC to generate vehicle usage patterns, terrain classification, and record fault codes from the engine and transmission [3].

In the TRIC algorithm, there is the requirement that the triaxial accelerometer be placed below the suspension on the axle of the vehicle. The accelerometer is called unsprung in this case because the signal received is not affected by the vehicle suspension. However, it would save costs if the accelerometer could be placed in the body of the vehicle with the rest of the sensor package. In this case the accelerometer would be called sprung because the signal would be influenced by the vehicle suspension. This research is interested in determining if the same TRIC classifications could be obtained from sprung accelerometer data and unsprung accelerometer data.

3.2 Data and Analysis

To make the comparisons between the sprung and unsprung accelerometers, data were used from seven tests performed at the Nevada Automotive Test Center (NATC) in the Fall of 2012. The tests were done with an Oshkosh Medium Tactical Vehicle Replacement (MTVR). The MTVR is a six-wheeled off-road vehicle and is used for a variety of tasks by both the U.S. Marines and the U.S. Navy [11]. Figure 23 shows an MTVR. NATC has a variety of paved and off-road courses that are used to test a variety of aspects and systems

of a vehicle. The data were collected at NATC to be used for different health monitoring and usage purposes, and not just for terrain classification. However, these data were chosen because the MTVR vehicles were equipped with the right sensors, including both sprung and unsprung accelerometers, to be able to implement the TRIC algorithm. Three unsprung accelerometers were placed on the vehicle: one on the front axle, one on the middle axle and one on the rear axle of the vehicle. The sprung accelerometer was placed at the front of the vehicle, inside the cab. Data were also used from a test done in the hills around State College, PA. This test was done with an MTVR and was collected by Penn State ARL.

Figure 23: U.S. Marines and U.S. Navy MTVR Source: oshkoshdefense.com, July 2014

3.2.1 Unsprung Accelerometer Locations

One advantage of using unsprung accelerometers is that the location where it is mounted shouldn't matter. Before looking at the comparison of the unsprung and sprung accelerometers, it is useful to compare the different unsprung accelerometers (front, mid

and rear). To compare data for each of seven tests done at NATC, the data from each accelerometer was run through the TRIC algorithm, and a percentage was calculated for how many classifications matched between each combination of accelerometers. Of the seven tests, three were done on gravel and four were done on paved roads. This is significant because it is more likely that the terrain classifications coming from the three accelerometers taking data over paved road would have a better match than the classifications from the accelerometers taking data on a rough road. After the first analysis, the percentages of classifications that were the same between accelerometers were much lower than expected. Once it was realized that there was a DC offset between the channels, and the data were corrected, the match greatly improved and was close to what was expected. Table 2 shows the results of the comparisons.

Table 2: The percentage of matching TRIC classifications between pairs of accelerometers

	Accelerometer Locations		
Paved Test Name	Front and Mid (% same)	Front and Rear (% same)	Mid and Rear (% same)
Acceleration	99.38%	100.00%	98.38%
Cruising	100.00%	100.00%	100.00%
Sort	100.00%	100.00%	100.00%
Warm Up	98.50%	98.50%	100.00%
Average	**99.47%**	**99.63%**	**99.60%**
Gravel Test Name			
Cruising	92.63%	93.68%	94.74%
Coast Down	84.06%	81.16%	84.41%
Steering	96.08%	92.15%	96.08%
Average	**90.92%**	**89.00%**	**91.74%**
Overall Average	**95.81%**	**95.07%**	**96.23%**

As expected, the classifications between the different channels on paved roads were almost exactly the same. The classifications on the gravel were the same between channels 90% of the time on average. This is not a huge difference, but the difference is significant enough that when doing the TRIC classification, an accelerometer location on the vehicle should be chosen and the same location used for all vehicles of that make. This will make the results consistent across all tests.

To verify the results, a test was done by Penn State ARL in the hills around State College, PA. The test was done on a paved road with loose gravel and dirt on the surface. The vehicle used in this test was the MTVR. The accelerometers were mounted in approximately the same places as they were in the previous test. In this test the classifications from front accelerometer only matched the middle accelerometer classifications 73% of the time, the front matched the rear only 55% of the time, and the middle matched the rear 82% of the time. The reason for the larger differences in this test are not known, but could have to do with the gravel, dirt, paved road combination, rather than just a paved road or just a gravel road as was the case above. Another reason may be that in the NATC test the MTVR was loaded, but in the ARL test the MTVR was not loaded. However, the conclusion remains the same that when outfitting a fleet of vehicles with sensors, the location of the accelerometer should be consistent.

3.2.2 Sprung Versus Unsprung Comparison

The first test examined here was a coast down test done on a gravel course with a loaded MTVR. This test gave good variation between high, medium and low amplitudes in the

TRIC classification of the short-wave length regime. The duration of the test was approximately 23 minutes. Figure 24 shows the path the MTVR took overlaid on a Google Maps image and Figure 25 shows the TRIC classifications for the short-wavelengths, also overlaid on a map. Figure 25 is here to give an idea of the bumpiness of the course. In Figure 25, each colored point on the map represents the RMS of 20 seconds of data. Because the sprung accelerometer was placed in the front of the vehicle, the data from the unsprung accelerometer located at the front was used for the TRIC algorithm because it was the closest to the sprung accelerometer.

To first compare the unsprung and sprung accelerometers, the time series of each set of data were plotted together and can be seen in Figure 26. The RMS value of the unsprung data is .2430 g's and the RMS value of the sprung data is .0367 g's or 6.6 times smaller than the unsprung. The suspension acts as a low-pass filter, attenuating the amplitudes of the higher frequencies. The general shape of both data series are similar, but different because of noise from the vehicle that gets into the sprung measurements. This was expected behavior because of the effects of the suspension.

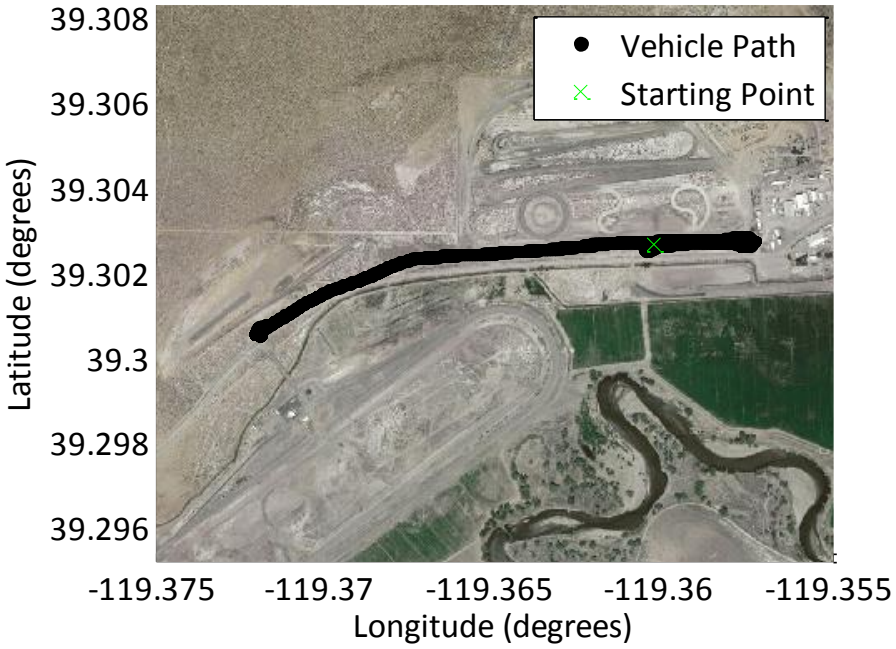

Figure 24: Path that MTVR took on a gravel course at the NATC. Data from this run will be used to compare unsprung and sprung accelerometers.

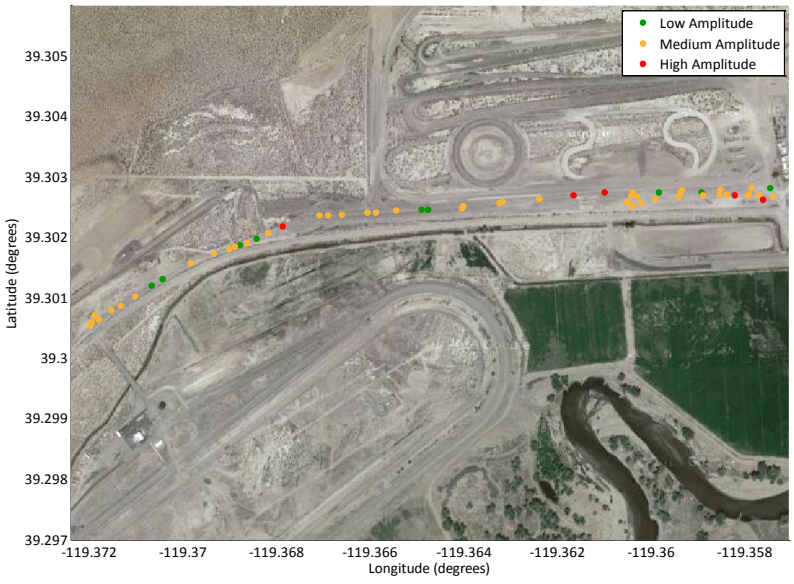

Figure 25: Short wavelength classifications of the gravel coast down test driven by an MTVR

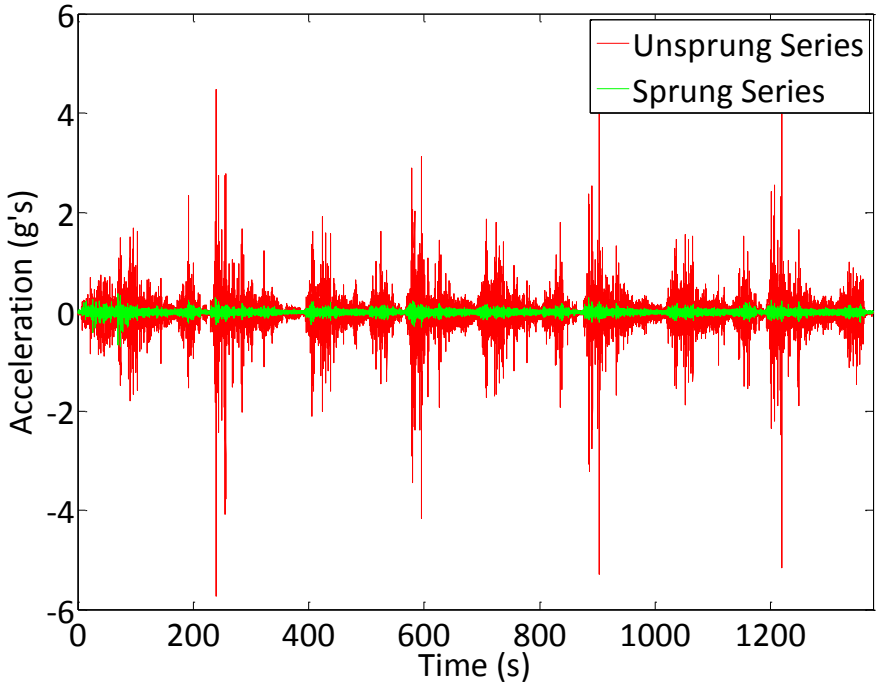

Figure 26: Time series of the unsprung and sprung accelerometer data from the MTVR coast down test on gravel course

The next step in comparison was to look at the spectral density of both the unsprung and sprung accelerometer data. An average spectral density was calculated for each set of data with a Hanning window applied to each time series record and a 50% overlap was done between each record. A 30 Hz low pass filter was applied to both sets of data. Figure 27 shows the plot of the spectral densities. The spectral densities are very similar until about 2 Hz, at which point the magnitude of the sprung spectral density drops significantly below the magnitude of the unsprung spectral density. This is because the suspension system of the truck acts as a low pass filter, attenuating the amplitudes of the higher frequencies that could cause system damage and operator and passenger discomfort. With a few exceptions, the shapes of the two spectral densities are very similar, giving

hope that the same TRIC classifications could be extracted from the sprung data as from the unsprung data. The peaks seen at 5 Hz and 35 Hz in the sprung PSD are from noise in the body of the truck that never made it to through the suspension to the unspung accelerometer.

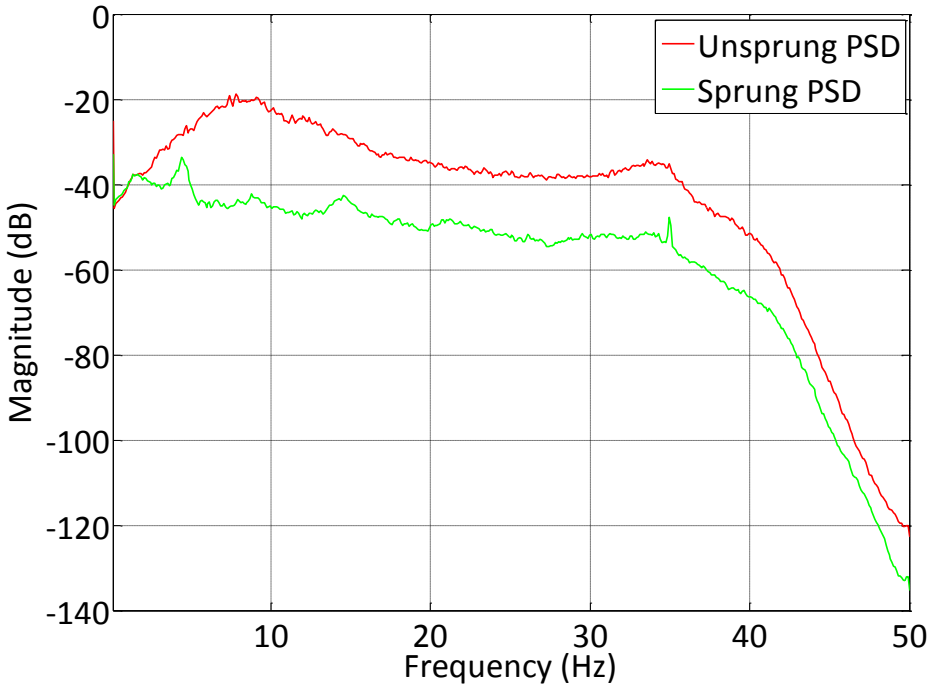

Figure 27: Spectral densities of the unsprung and sprung data from the MTVR coast down test on a gravel course

The next step was to obtain a transfer function between the unsprung and sprung data for several of the test courses. Normally, if the suspension was being analyzed, the transfer function between the unsprung mass and the sprung mass would be calculated. Since this research is interested in obtaining an estimate of the unsprung accelerometer data from the sprung accelerometer data, the transfer function from the sprung mass to the unsprung mass is computed. This avoids having to do an inversion once the transfer

function is computed. Once computed, the transfer functions can then be averaged and the average can be applied to the sprung accelerometer data collected by any vehicle of the same type and passed through the TRIC algorithm. To help negate noise, a transfer-function estimate in the frequency domain was obtained using the cross-spectrum of the sprung and unsprung data. Equation (3.1) shows the equation for the transfer function estimate.

$$H(f) = \frac{\bar{G}_{XY}}{\bar{G}_{XX}} \tag{3.1}$$

where *H(f)* is the transfer function, $\bar{G}_{XY}$ is the average cross-spectrum from the sprung to the unsprung accelerometer, and $\bar{G}_{XX}$ is the average auto-spectrum of the sprung accelerometer data. With this method, noise is averaged out in the output channel and it also preserves the phase relationship between the input and output channels, which is necessary to be able to reconstruct the unsprung accelerometer data [12]. Figure 28 shows a plot of the transfer function estimate for the MTVR on the gravel course.

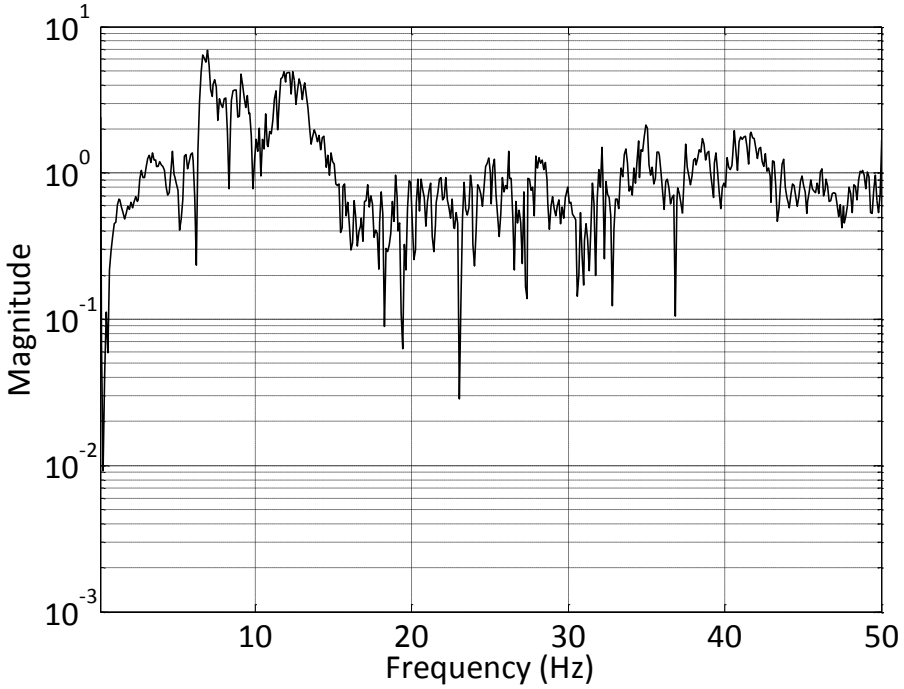

Figure 28: Transfer function estimate between the unsprung and sprung accelerometer channels for the MTVR on an NATC Gravel course

To see how good the transfer estimate was, the coherence function was calculated. Coherence gives a measure of how strongly two signals are related, or how coherent they are. It gives a value between zero and one. Two signals that are completely coherent will have a value of one. If the coherence is not close to one, then the estimate of the transfer function is probably poor. Equation 3.2 gives the equation used to calculate the coherence [12].

$$\gamma^2 = \frac{(\bar{G}_{XY})^* \bar{G}_{XY}}{\bar{G}_{XX}\bar{G}_{YY}} \tag{3.2}$$

where γ^2 is the coherence value.

Figure 29 shows the coherence of the unsprung and sprung accelerometers of the MTVR on the gravel course.

Unfortunately, the coherence values turned out to be pretty poor across the whole spectrum. The reason that the coherence is so poor is that the sprung accelerometer is receiving signals from other sources on the truck that don't make it to the unsprung accelerometer, such as vibrations caused by the engine. The coherence indicates that the transfer function estimate would probably not be very accurate. However, it was decided to apply the transfer function to the sprung accelerometer data in an attempt to reproduce the unsprung data, and then compare it to the original unsprung data and see what kind of results could be obtained.

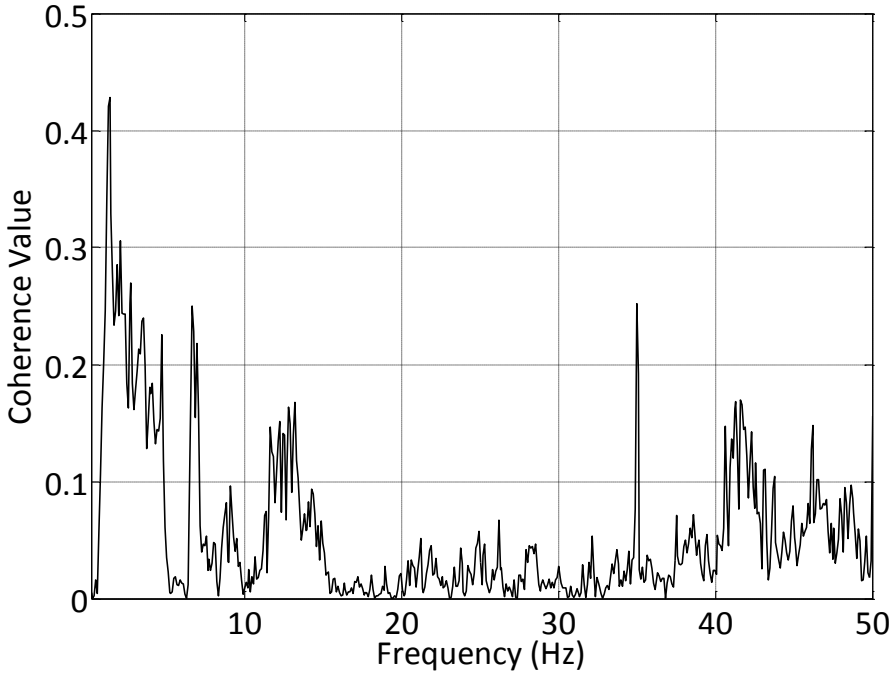

Figure 29: Coherence values from 0-50 Hz of the unsprung and sprung accelerometers mounted on a MTVR driven over a gravel course

The unsprung data can be reproduced a couple of different ways. The first way is to take the transfer function estimate in the frequency domain, take the inverse fast Fourier transform (FFT), and convolve that with the sprung time series data. The other way, considered to be simpler, is to take the FFT of the sprung series time data, multiply that with the frequency domain transfer function estimate, and take the inverse FFT of the result to give a reproduced unsprung time series data. Since the transfer function estimate was obtained by averaging the spectral densities, it will be shorter in record length than the FFT of the sprung time series data. This can be handled in a couple of different ways so that the transfer function estimate can be multiplied with the spectrum of the sprung data.

The first way is to interpolate the transfer function estimate to make it the same length as the sprung data spectrum. This is done by taking the inverse FFT of the transfer function, appending as many zeroes as needed, taking the FFT of the appended time series transfer function, and multiplying this with the sprung spectrum. The problem with this approach is that the complete sprung time series is needed before the transfer function estimate can be used to reproduce the unsprung data. This is fine if there is no interest in real-time classification, as it could only be used to classify the terrain after the run is complete.

The second way to handle multiplying the frequency domain transfer function to the spectrum of the sprung data is to multiply the transfer function to sections of the sprung series spectrum of the same length as the transfer function. Two similar methods can be

applied to avoid discontinuity at the end of each record where the transfer function is applied. The two methods are the overlap-add and the overlap-save [13, 14].

The overlap-add method divides the data into records of length K, usually a power of two for efficiency. The inverse FFT of the transfer function estimate, *h*, can be taken and zero padded to interpolate the transfer function if needed. The FFT of the record of length K is taken, multiplied with the transfer function estimate, and the inverse FFT is taken of the result to obtain the unsprung accelerometer estimate. If *h* is an arbitrary length P, then when a convolution in the time domain between *h* and the record of length K is taken, the result will be of length K+P-1. Because of this, the final P-1 points of the unsprung estimate are added with the first P-1 points of the next unsprung estimate record. This avoids inaccuracies that arise from the circular convolution. The overlap-save method is similar, but the first P-1 points of each unsprung estimate are dropped and the record is appended to the previous record. It is the first P-1 points that are inaccurate when a circular convolution is done [13, 14]. Both methods are accurate and give the same result, but the overlap-save is more intuitive and the algorithm is slightly easier to code, and is the method that was used for this research. When using either the overlap-save or the overlap-add method, processing can be done in real time and the terrain classification can be transmitted immediately to researchers. When done on many vehicles operating in the field, correlations can be made to terrain harshness and vehicle system damage. Figure 30 shows a chart of the overlap-add method.

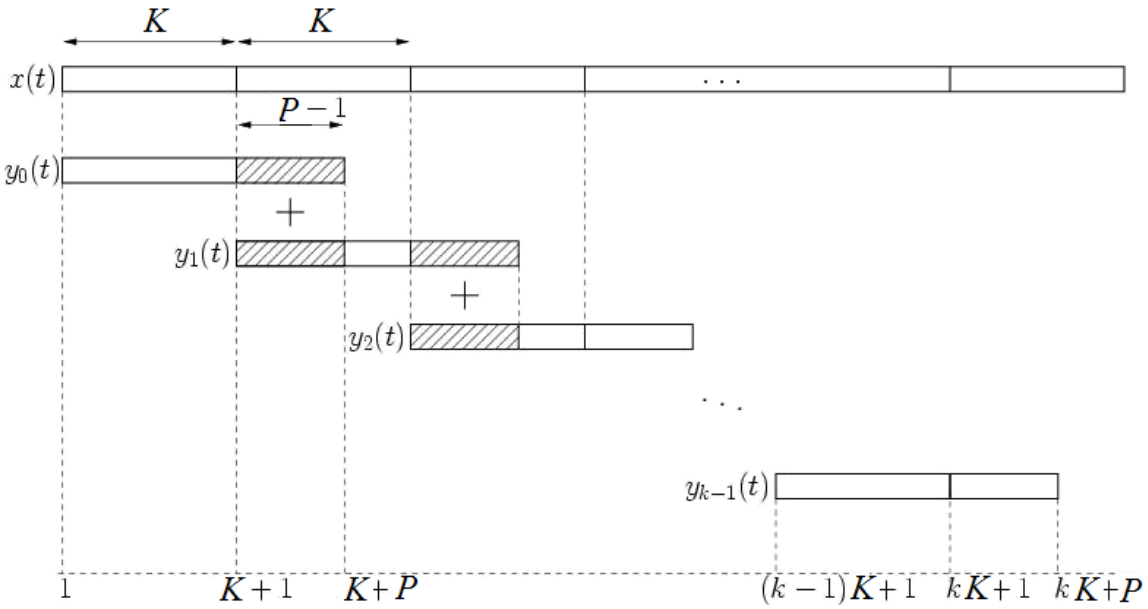

Figure 30: Diagram of the overlap-add method where little k denotes the number of records and little l denotes the beginning value of *x(t)*
Source:
http://commons.wikimedia.org/wiki/File:Depiction_of_overlap-add_algorithm.png#mediaviewer/File:Depiction_of_overlap-add_algorithm.png,
letters changed to reflect notation in this thesis

The overlap-save method was used to obtain an estimate of the unsprung spectral density with the MTVR data from the gravel coast down test. The spectral density of the unsprung data estimate was taken and compared to the spectral density of the original unsprung data to check for accuracy. Figure 31 shows the results of the spectral densities.

Unfortunately, the transfer function estimate was very poor. Comparing Figure 31 to Figure 27, it can be seen that the spectral density of the unsprung data estimate resembles the sprung data spectral density in magintude, except in the region of 0-15 Hz and around 35 Hz where the values of the coherence function are slightly higher. A plot of the time series, shown in Figure 32, shows that the amplitudes of the unsprung estimate data are much smaller than the amplitudes of the original unsprung data.

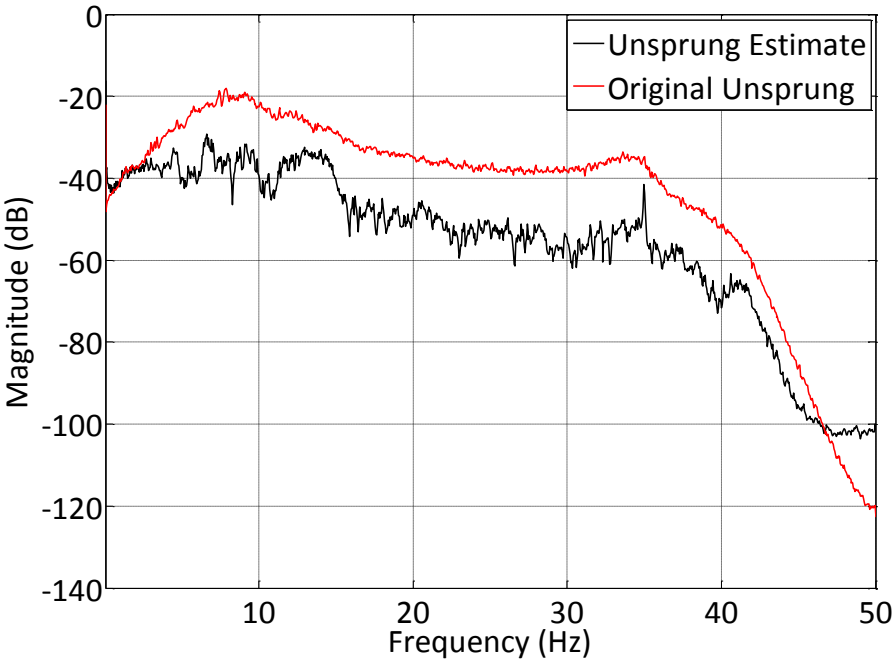

Figure 31: Spectral densities of the unsprung data and the unsprung data estimate from MTVR data on a gravel course doing a coast down test.

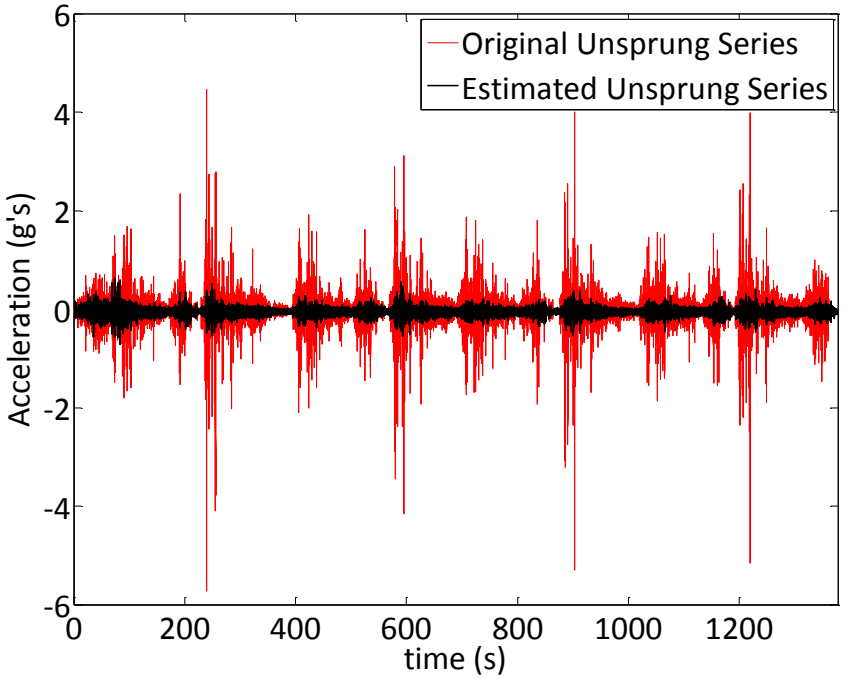

Figure 32: Original unsprung data time series plotted with the estimate of the unsprung series obtained using the transfer function series

The RMS value of the unsprung data estimate is .085 g's which is 2.3 times bigger than the RMS of the sprung data, but is still 2.9 times smaller than the RMS of the original unsprung data. This means that the transfer function did have an effect on the sprung data, but the effect was not large enough to accurately represent the unsprung data.

To help improve the match between the measured and estimated unsprung spectra, a gain was applied to the transfer function estimate that would make the RMS value of the original unsprung data equal to the RMS value of the estimated unsprung data. Since the TRIC algorithm uses 20 second RMS values, doing this should help the transfer function estimate give better terrain classification results.

The sum of the spectral density values, over a range of frequencies, multiplied by the frequency spacing, Δf, (this is equivalent to integration of a continuous function) will be equal to the mean-square value of the time series [12]:

$$\frac{1}{N}\sum_{n=0}^{N-1} x_n^2 = \sum_{m=0}^{N-1} S_{XX} \cdot \Delta f \tag{3.3}$$

where x_n is the time series, N is the length of x_n, and S_{XX} is the spectral density. The square root (because the *root* mean square needs to be the same for the time series) of this 'integration' performed on the original unsprung data spectral density can be divided by the square root of the integration of the estimated unsprung data spectral density to give the gain value to be applied to the transfer function. If the gain values for the tests were similar, then the average gain value could be used to give an improved estimate of the terrain classifications.

For the gravel coast down test used in used for the analysis in Section 3.2.2, the gain applied to the transfer function estimate was √8.19 or 2.86. Figure 33 shows the PSD of the unsprung estimate from the transfer function with the gain applied overlaid on top of the original unsprung PSD. Looking back at Figure 31, it can be seen that the transfer function with the gain applied gives a much better estimate of the unsprung data.

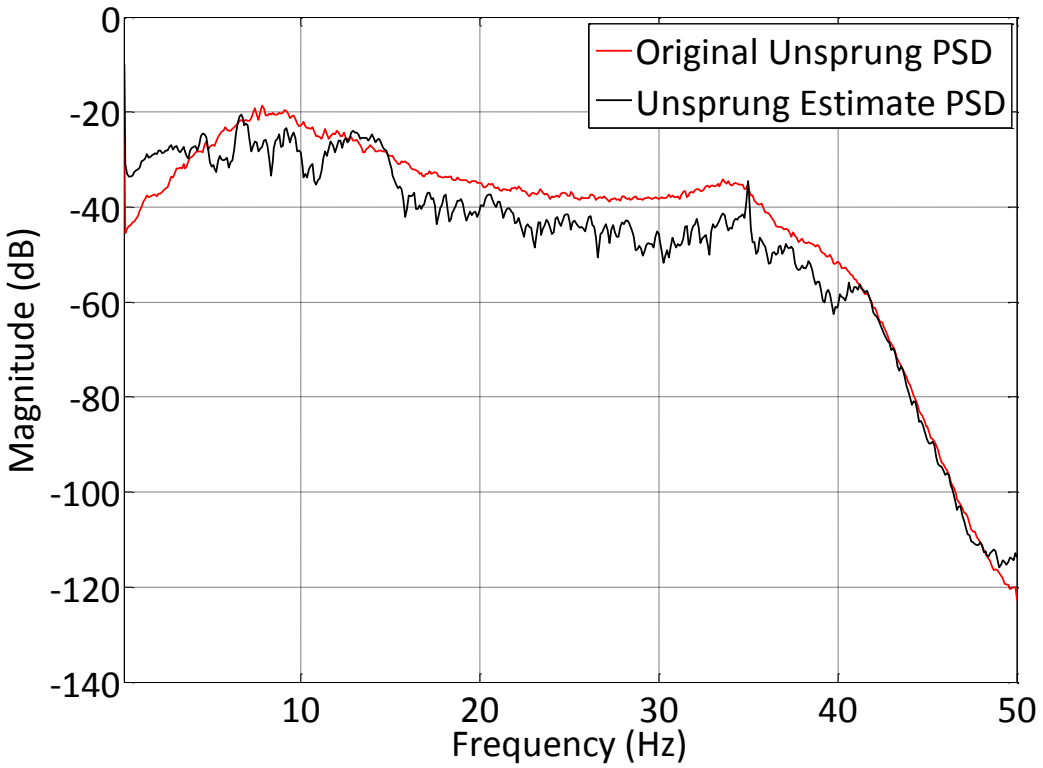

Figure 33: PSD of the unsprung data estimate obtained from applying a gain of 2.86 to the transfer function estimate

Next, the short wavelength terrain classification was performed for both the original unsprung and the estimated unsprung data. Figure 34 shows the scatter plots of the classifications, as well as the scatter plot for the classifications from the sprung accelerometer.

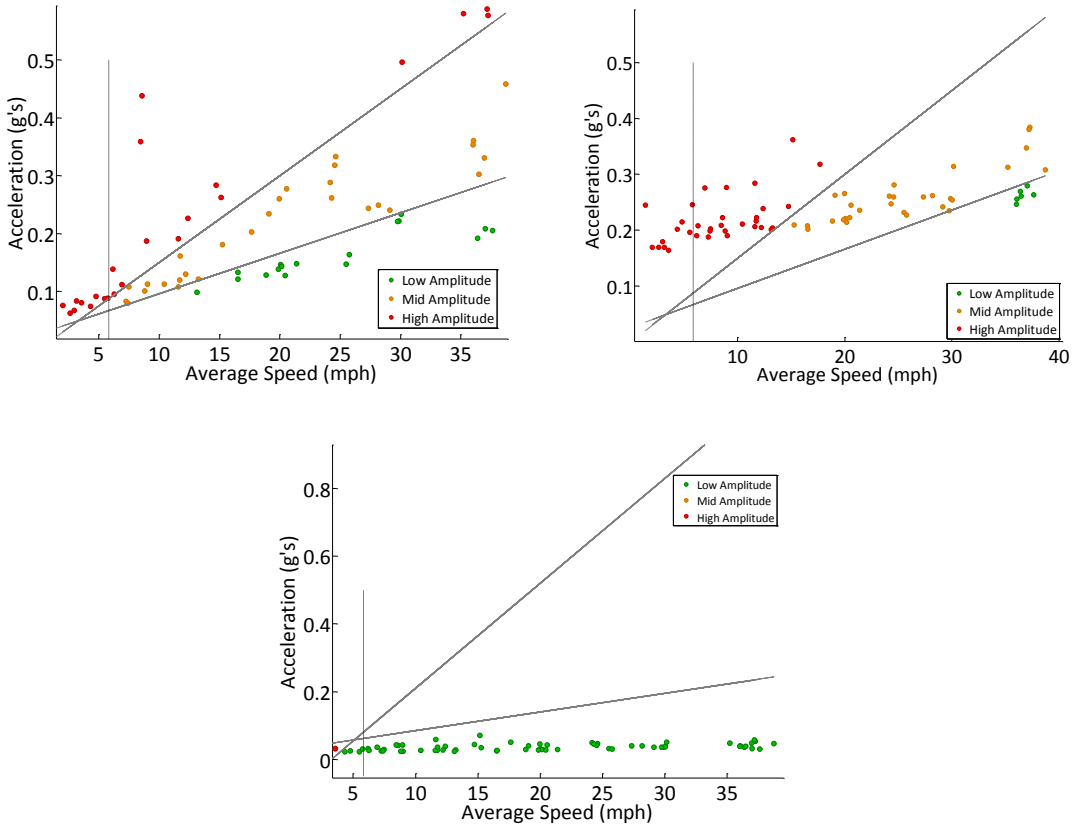

Figure 34: Top left shows the short wavelength terrain classifications for the original unsprung data, top right shows the classifications for the unsprung data estimate obtained after the gain was applied to the transfer function, and the bottom is the classification for the unsprung data estimate with no gain applied to the transfer function

In this case when the transfer function with the gain is used, the data is classified higher than it should be, especially at the lower speeds, and classified lower than it should be at the higher speeds. Before the gain was applied to the transfer function, only 39.13% of the classifications were the same. After the gain was applied, 52.17% of the classifications were the same. While the results are better than the classifications obtained from the original sprung data, they are still fair at best. However, if one was limited to using only a sprung accelerometer while classifying terrain, it is recommended to apply the gain to the transfer function estimate to give better results.

When the gain was applied to the transfer functions of tests on paved roads, the accuracy of the classifications went down. This is because on the paved roads the classifications were mostly low amplitude. The transfer functions generally lowered the RMS amplitudes, so the classifications remained the same – low amplitude. For the three gravel tests, applying the gain to the transfer function estimate improved the classification accuracy, sometimes by a large amount and sometimes by a small amount. The amount of the gain applied to the transfer function estimate for each of the seven tests and the classification accuracy before and after the gain was applied can be seen in Table 3.

Table 3: Accuracies of the terrain classifications for the unsprung data estimates before and after a gain was applied to the transfer function estimate, as well as the gain values

Paved Tests	Gain Applied to Transfer Function	Accuracy before gain	Accuracy after gain
Acceleration	2.53	99.38%	85.19%
Cruising	2.42	99.23%	74.05%
Sort	1.14	93.68%	91.58%
Warm Up	2.07	98.50%	95.49%
Gravel Tests			
Cruising	3.43	3.16%	69.47%
Coast Down	2.86	39.13%	52.17%
Steering	1.90	82.35%	84.31%

It is important to remember when applying the transfer function estimate that it may classify roads with low harshness, such as a paved road, as a road with medium harshness. On average in the NATC tests, less than 10% of the points that should have been classified low amplitude were classified as mid amplitude.

The above method of obtaining the transfer function estimated was also done on data obtained by Penn State ARL in the hills around State College, PA. The sprung accelerometer was attached to the frame of the vehicle above the suspension at each of the three axles. The test was done on a paved road with gravel and dirt on the surface. Figure 35 shows the route of the vehicle on a map.

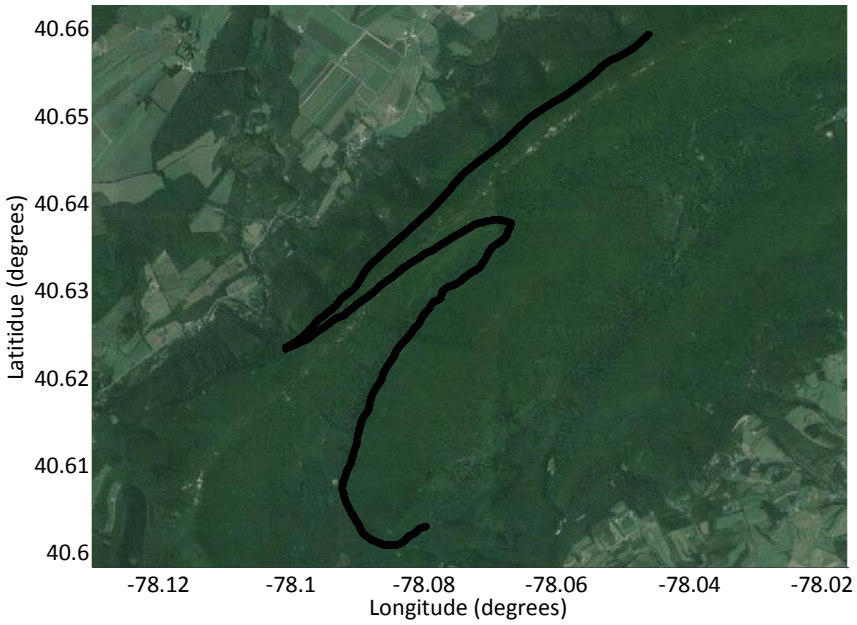

Figure 35: Path of the MTVR in the hills near State College, PA

First, the spectral densities were produced for each of the accelerometers, as shown in Figure 36. The spectral densities are similar to the ones in Figure 27, except where the sprung PSD rises up at around 30 Hz. This indicates that some noise between 20-30 Hz was getting into the sprung accelerometer but not into the unsprung accelerometer.

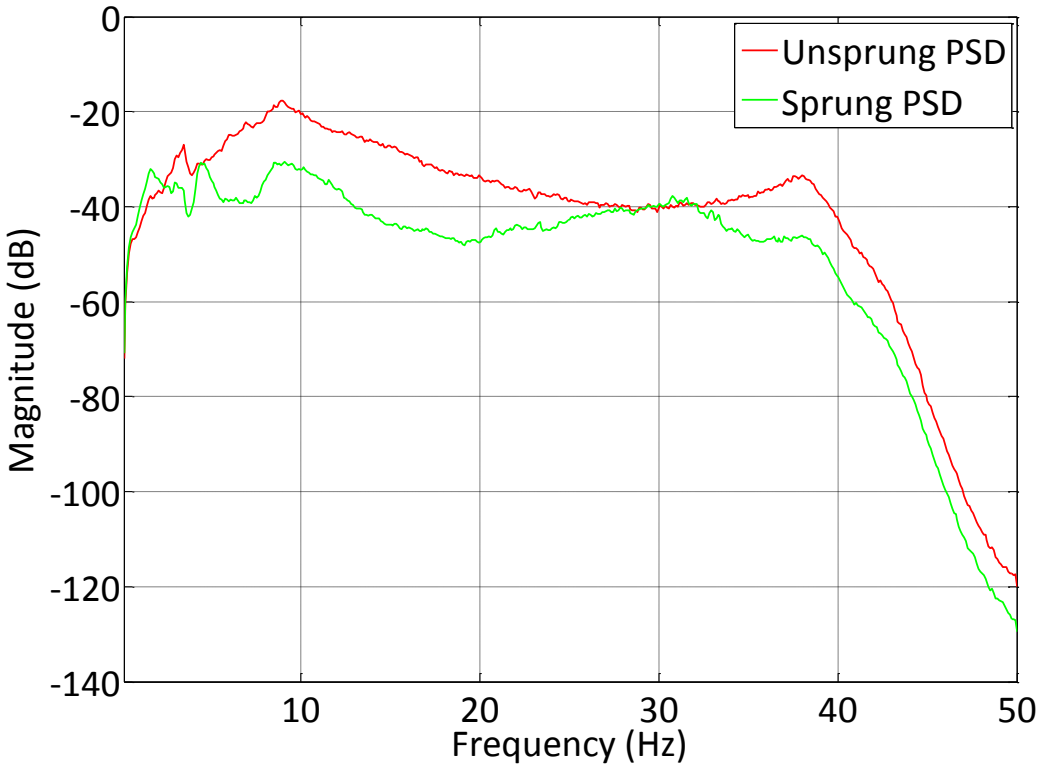

Figure 36: Spectral density for the sprung and unsprung accelerometers used on the MTVR in the State College test

Next, the coherence was generated. It is shown in Figure 37. The coherence values for this test are better than the coherence values of the NATC test (the closer the value is to one, the more coherent the two signals are at that frequency). The spot where the coherence is worst is in the same frequency range that the sprung PSD had a "hump." The improvement in coherence values probably has to do with the location where the sprung accelerometer was mounted. In the NATC test, the sprung accelerometer was mounted in the cab of the vehicle, which was not over the axle where the unsprung accelerometer was located. In the State College test, the accelerometer was mounted on the frame above the

suspension. With these improved results, the transfer function estimate should be improved as well.

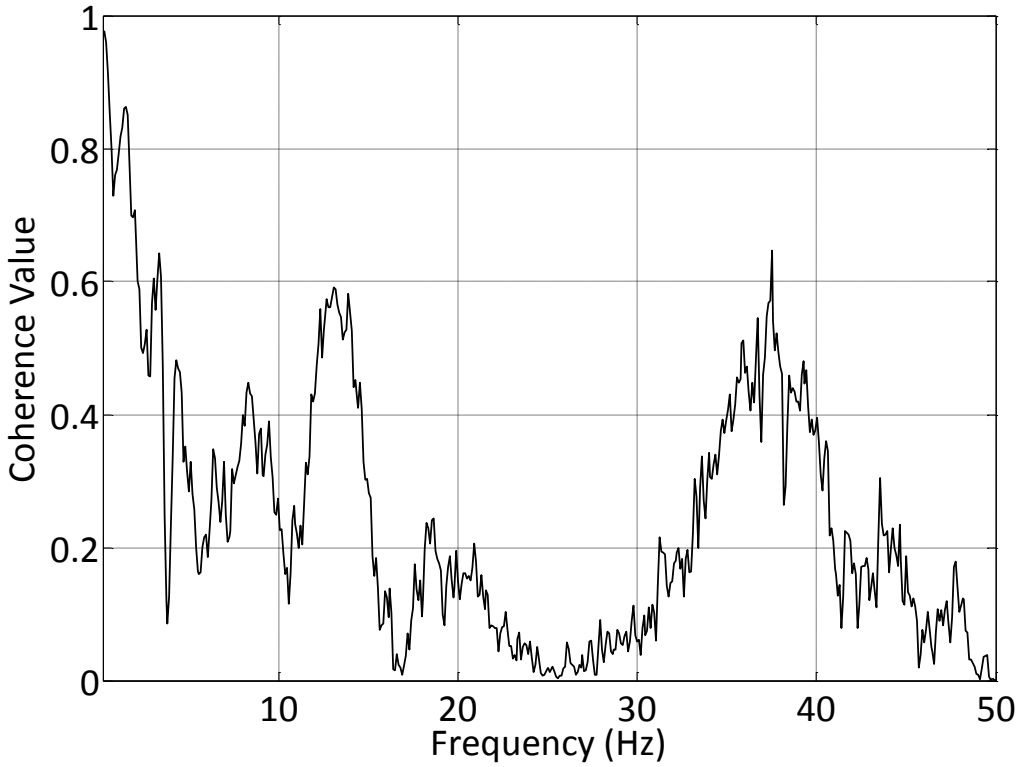

Figure 37: Coherence values for the MTVR test in the State College

Next, the transfer function was produced, and it is shown in Figure 38. This is the transfer function with the gain applied to equalize the RMS values of the unsprung data and the unsprung estimate, as was discussed in the last test. Again, there is the problem between 20-30 Hz, but hopefully it will not affect the TRIC classifications significantly.

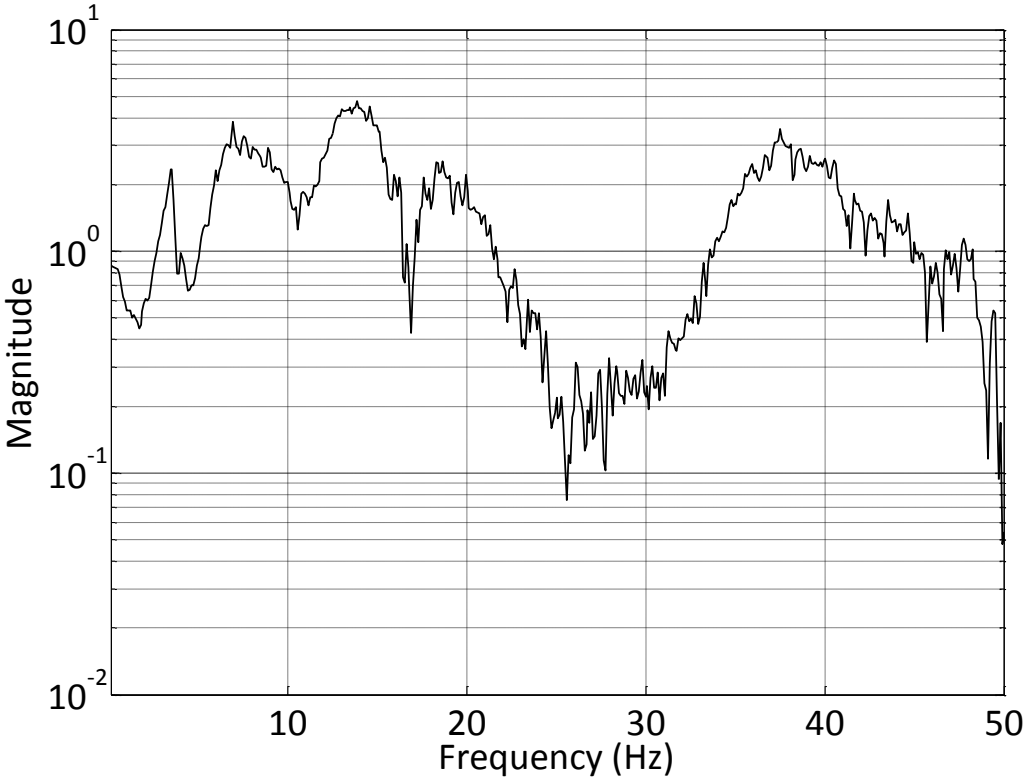

Figure 38: Transfer function estimate between the sprung accelerometer and the unsprung accelerometer for the MTVR State College test

Figure 39 shows the spectral densities of the unsprung accelerometer data and the unsprung estimate data. The unsprung estimate PSD shows improvement from the PSD, and is very close in shape and magnitude to the measured unsprung PSD. Referring back to Figure 33, it can be seen that the transfer function estimate did a better job in estimating the unsprung data from the sprung data in this test than in the NATC test.

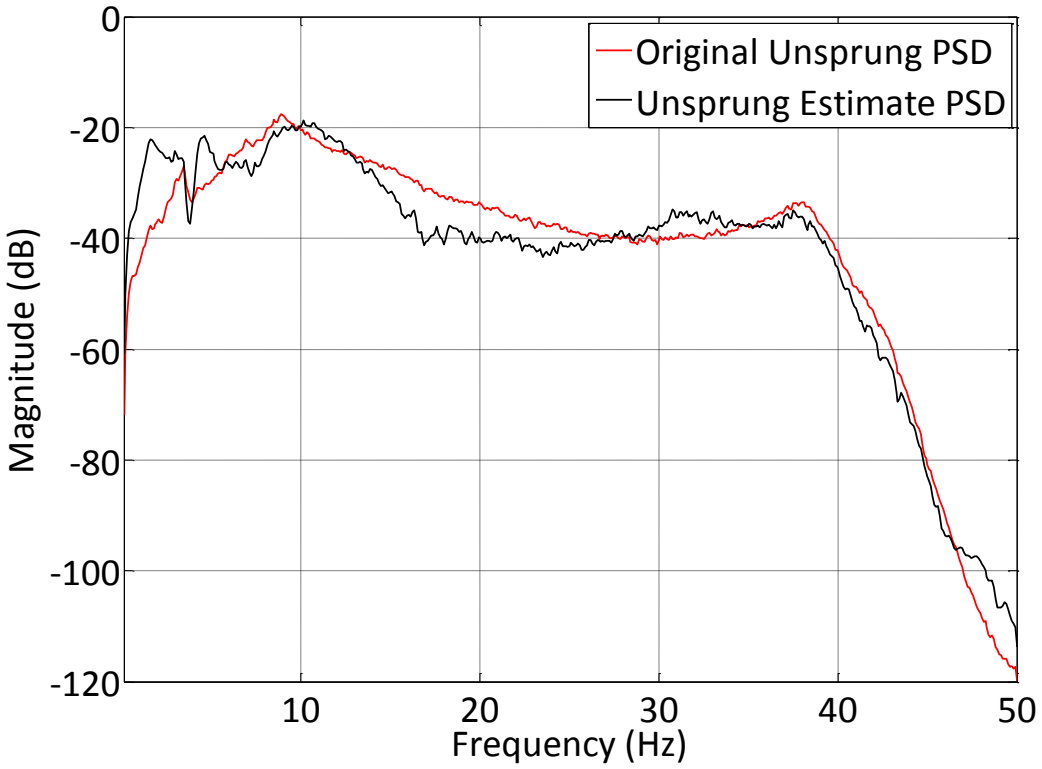

Figure 39: Spectral densities for the unsprung accelerometer and the unsprung estimate from the sprung accelerometer used on the MTVR in the State College test

Finally, the unsprung estimate data was processed with the TRIC algorithm. Figure 40 shows the scatter plots of the short wavelength TRIC classifications for the original unsprung data, the unsprung estimate data and the original sprung data. The classifications from the original unsprung data matched the classifications from the unsprung estimate data 97% of the time. This shows a large improvement over the NATC test, which only matched 52% of the time. This test shows that when the sprung accelerometer is mounted to the frame of the vehicle, rather than in the cab, the transfer function estimate becomes a more viable option.

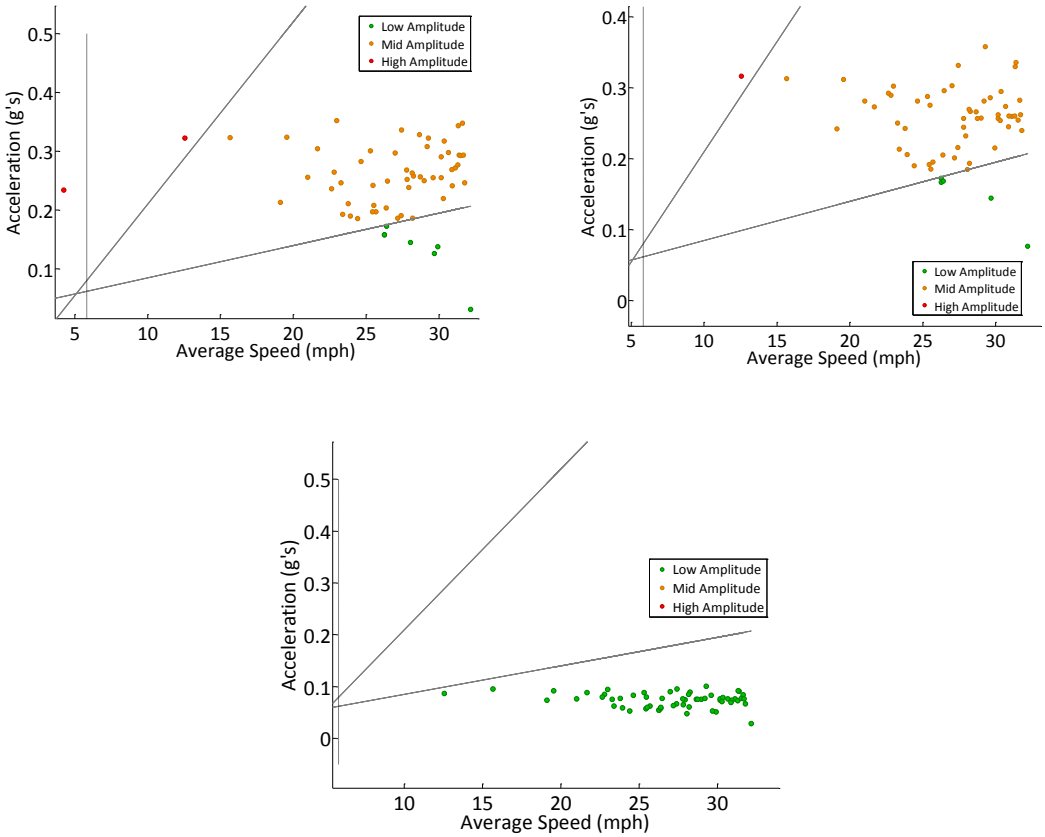

Figure 40: Top Left: Classifications from the original unsprung accelerometer; Top Right: Classifications from the unsprung estimate; Bottom: Classifications from the sprung accelerometer

3.3 Transfer Function Estimate Conclusions

The transfer function estimates were obtained for seven different MTVR test runs on various NATC courses, including the coast down test on the gravel course discussed in the previous section. In these tests the unsprung accelerometer was mounted on the front axle and the sprung accelerometer was mounted in the cab of the vehicle. The results for all seven sets of data were similar to results of the gravel course test run shown. While some had better coherence functions than others, all had poor coherence. This led to

estimated usprung data that was too low in amplitude across the whole range of frequencies. To help improve the transfer function estimates, a gain was applied to each of the seven transfer functions. The gain was chosen so that the RMS of the original unsprung data and the unsprung data estimates were the same. The results of applying the gain improved results from some courses but not for others. For paved tests, the accuracy of the classifications decreased after the gain was applied. This is largely due to the fact that the data for paved road tests are mostly classified as low amplitude and remain so with the estimated unsprung data. For the gravel tests, there was improvement with the gain applied to the transfer function, but the accuracy of the classifications was still fair at best.

Another test was done by Penn State ARL that showed even better that the transfer function estimate is a viable option for estimating the unsprung accelerometer data from the sprung accelerometer. This test was done on paved road with a dirt and gravel covering in the hills near State College, PA. In this test, the sprung accelerometer was mounted on the frame above the suspension rather than in the cab. The classifications from the unsprung estimate data matched the classifications from the original unsprung data 97%, a 77% increase over using the original sprung data.

Another way that the sprung accelerometer may be used to replace the unsprung accelerometer is to define the thresholds the same way that the thresholds were defined using the unsprung data. The vehicles could be driven on courses with known harshness levels and the thresholds for sprung accelerometer data could be set so that all data

classified on course sections classified by the Army as off-road would be classified as high-amplitude by TRIC, sections classified by the Army as secondary would be classified as mid-amplitude by TRIC, and sections classified as primary by the Army would be classified as low-amplitude by TRIC.

It may be possible to obtain a better transfer function if parameters of the vehicle are known, such as the mass of the vehicle, the spring and damping properties of the suspension, and the spring properties of the tire (tire pressure). Figure 41 shows what is known as a quarter car model. It is the vehicle on one side of one axle modeled as a mass, spring and damper system.

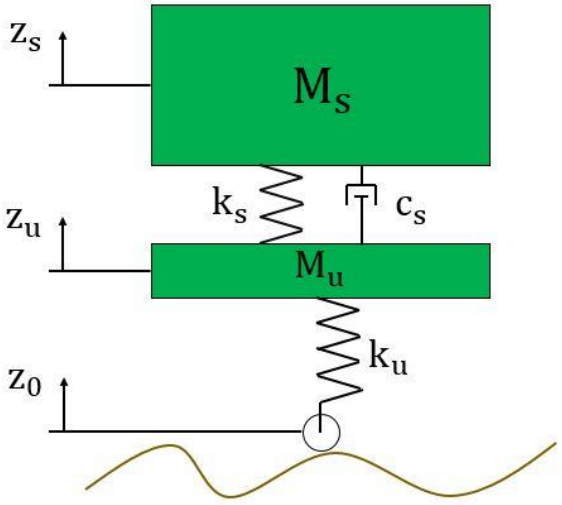

Figure 41: A quarter car model showing the vehicle modeled as a mass, spring and damper system on one side of one axle Source: adapted from Balogh. L. Road Surface Estimation for Control System of Active Suspensions

The equations of motion [15] for this system are,

$$M_s \cdot \ddot{z}_s = c_s \cdot (\dot{z}_u - \dot{z}_s) + k_s \cdot (z_u - z_s) \tag{3.4}$$

$$M_u \cdot \ddot{z}_u = k_u \cdot (z_0 - z_u) - c_s \cdot (\dot{z}_u - \dot{z}_s) - k_s \cdot (z_u - z_s) \tag{3.5}$$

where M is mass, k is spring stiffness, c is the damping coefficient and z is displacement. The subscripts s and u denote sprung and unsprung mass.

Because the motion of M_u depends on the motion of M_s, each unsprung location on the truck will have a different acceleration, since M_s represents the portion of the overall vehicle mass contributing to the motion at that location and is not the same at each location. This explains why the tests resulted in different classifications at each of the three unsprung accelerometers. Future work may look into where the best location for the unsprung accelerometer is.

A Laplace transform of Equation 3.4 can be taken in order to obtain the transfer function between the sprung and unsprung accelerometers [15].

$$M_s \cdot Z_s \cdot s^2 = c_s \cdot (Z_u - Z_s) \cdot s + k_s \cdot (Z_u - Z_s) \tag{3.6}$$

where s is the complex angular frequency. The transfer function can be obtained by solving Equation 3.6 for Z_u/Z_s.

$$T_{su} = \frac{\frac{c_s}{M_s} \cdot s + \frac{k_s}{M_s}}{s^2 + \frac{c_s}{M_s} \cdot s + \frac{k_s}{M_s}} \tag{3.7}$$

Also,

$$\omega_0 = \sqrt{\frac{k_s}{M_s}} \tag{3.8}$$

$$2 \cdot \omega_0 \cdot \xi = \frac{c_s}{M_s} \tag{3.9}$$

where ω_0 is the natural frequency of the sprung mass and ξ is the damping ratio [16]. Plugging Equations 3.8 and 3.9 into Equation 3.7 gives,

$$T_{su} = \frac{2 \cdot \omega_0 \cdot \xi \cdot s + {\omega_0}^2}{s^2 + 2 \cdot \omega_0 \cdot \xi \cdot s + {\omega_0}^2} \tag{3.10}$$

Unfortunately, the natural frequency and the damping ratio are not known for the data sets used in this research. Future work may include determining these factors for the vehicles to obtain a better transfer function that can remove the sensitivity to vehicle weight, tire pressure, and suspension properties. Because of the transfer function's dependence on M_s, it would only be good for one location on the vehicle. This is further motivation to determine where the best location would be.

Chapter 4 - Vehicle Operational Modes

4.1 Methods

The Applied Research Laboratory (ARL) at Penn State has used a method of clustering vehicle operational data into groups for maintenance and fuel conservation purposes. These vehicle operational modes may be correlated to the TRIC classifications. The purpose of this chapter is to examine how well this clustering process groups operational variables. If successful, this classification could be used with the TRIC algorithm to see how terrain affects operator response. This could be useful in helping to reduce driver fatigue and vehicle system damage.

These modes are obtained by calculating the 20 second average values of vehicle parameters such as engine speed, vehicle speed and accelerator position that are dependent on how the driver is operating the vehicle. The data is then passed through a k-means clustering algorithm that outputs defined clusters of vehicle parameters that can then be used to define the modes of operation of the vehicle. Many more vehicle operating parameters are available on the vehicle data bus, but the engine speed, vehicle speed and accelerometer position were down-selected as the most relevant parameters for relating operating modes to fuel economy, and possibly maintenance.

K-means clustering takes data with n variables and partitions the data into k clusters. The algorithm works by choosing k initial cluster centers. Each iteration of the algorithm takes each point in the data set and, using the Euclidean distance (n-dimensional), assigns that point to the closest cluster center point. Then, the mean of all the points is taken for each group, and each center point is updated to be that mean. Once there are no more changes to the center points, the algorithm is considered to have converged [17]. Figure 42 shows the flow diagram for the k-means algorithm.

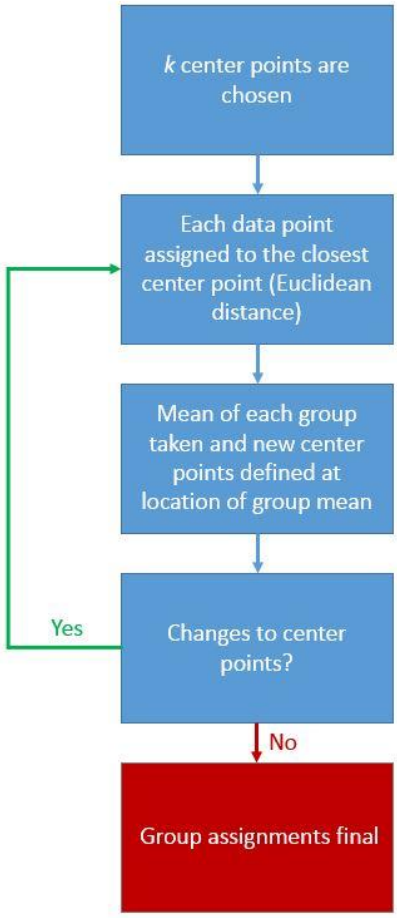

Figure 42: Flow diagram for the k-means algorithm

Because *k* is a user input, considerations must be made when choosing its value. In this case, there are three variables (engine speed, accelerator position, and vehicle speed) going into the operational modes. One method of choosing the value of *k* follows. If a scatter plot were done of the variables, there would be three axes, one for each variable. If each axis were divided into three equal portions, this would create 27 cubic regions on the plot, which could be potential clusters. Three divisions of each variable gives a variety of vehicle operational modes, and is the same number of potential classifications from the TRIC algorithm. Because the TRIC algorithm really has 28 classifications because of the idle classification, it was decided to use $k = 28$ for the k-means algorithm.

MATLAB has a built-in k-means function that makes this process very simple. The syntax in MATLAB is `IDX = kmeans(X,k)` where X is an n-by-p matrix with rows being the data and the columns representing each variable. Then *k* is the number of desired clusters and IDX is an n-by-1 vector with the cluster number for each point [18].

To help in visualizing the operational modes, an ARL engineer wrote a program in the Python language that randomly assigns a color to each operational mode defined by the k-means clustering. It then places colored points on a map in Google Earth along the path of the vehicle. Each dot represents 20 seconds of averaged data. This program inspired the same technique for displaying the TRIC classifications like the map seen in Figure 17.

Instead of using the Python code to display the maps, a MATLAB program was written supplemented with an open source function called `plot_google_map` obtained from the MathWorks website [19]. This code plots the points overlaid on a Google Map image with

axes defined by latitude and longitude specified in the code. The advantages of the Python program utilizing Google Earth is that elevation detail in the terrain can be viewed. The advantage of using the MATLAB code with Google Maps is that all other code done for this research was done in MATLAB, making it easy to transition from one MATLAB program to another.

The maps by themselves can only show how often a vehicle is changing modes. This is because there is no meaning behind the number that the k-means algorithm assigns to each group (i.e. it is not in ascending harshness like is the case in the TRIC algorithm classifications) and the colors assigned to each group number are generated randomly as well. However, there are some other ways to visualize what is happening and obtain more information about the operational modes.

First, a 3D scatter plot can be produced with each of the operational mode variables as one of the axes. Each point can be color-coded to the color assigned to its corresponding k-means group number. This can give a good idea of what region each color represents. Second, a histogram can be used to see how the data points fall within each k-means group number. Each bar is color-coded to match the color assigned to the corresponding k-means group number. Similar to the histogram, a color-coded pie graph can also be used.

In large data sets, it can be difficult to see on the 3D scatter plot what region the points lie in. Another useful way to more concisely see where the vehicle is operating in each group number is that each variable of the data can be placed in a column of a spreadsheet and the group numbers can be placed in a fourth column. Then the data can be sorted by group

number from lowest to highest, as long as the data is sorted with the group numbers it will give an easier way to see where the vehicle is operating in each group. The pie chart or the histogram can then be referred to to see what color each group belongs to (see next section for examples).

4.2 Data and Analysis - NATC Oval Paved Track with MTVR Data

Data obtained from the NATC tests with the MTVR were used to make maps with the vehicle operational modes. ARL collected data from a HEMTT driven around and near State College, PA that were used for this portion of the analysis. The three operational variables used in these tests are engine speed, measured in revolutions per minute (RPM), accelerator position, measured as a percentage of the maximum depression of the pedal, and the vehicle speed, measured in miles per hour (mph).

The first test examined here was done on a paved, oval track with a loaded MTVR. The test lasted approximately one hour and three minutes. Figure 43 shows the map of the course overlaid after the data had been passed through the k-means algorithm. The figure shows that the modes are frequently changing. This could be for a variety of reasons, one of which could be changes in terrain features that will be looked at in the next chapter. Another reason could be the nature of the test. It could be that the purpose of the test was to change speeds frequently and/or accelerate or decelerate frequently. It is also more difficult to tell where the changes are taking place when the course is a loop and the test involves several laps. In that case, from the map alone, it would be difficult to tell where the operational mode changes take place. The histogram and the pie chart, see Figures 44

and 45 respectively, show which group the colors belong to and how often they occur. The 3D scatter plot seen in Figure 46 can be used to see the variable values for each group.

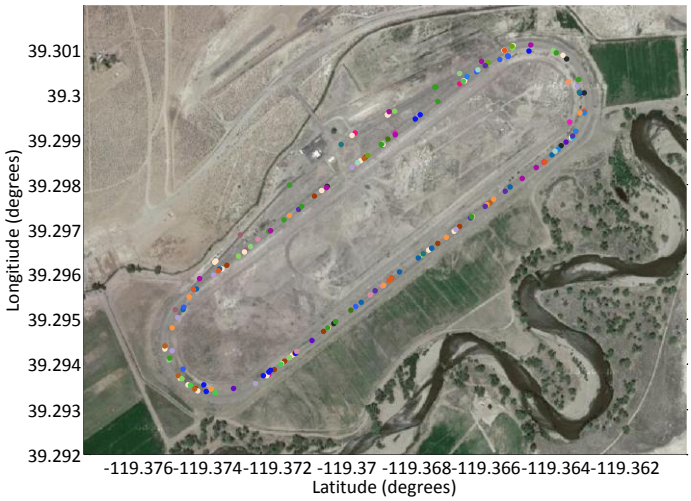

Figure 43: Map displaying the color-coded vehicle operational modes, sorted into groups by the k-means algorithm in MATLAB

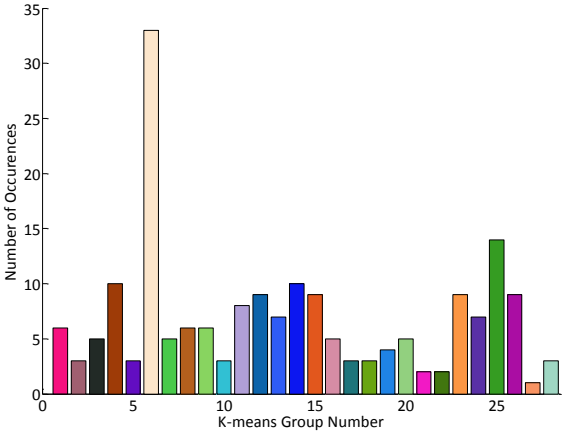

Figure 44: Histogram showing how often each of the color-coded groups generated by the k-means algorithm occur

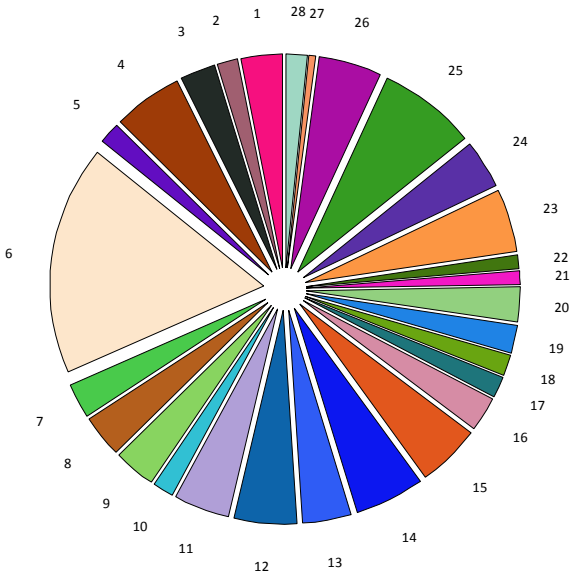

Figure 45: Pie chart showing how often each of the color-coded groups occur

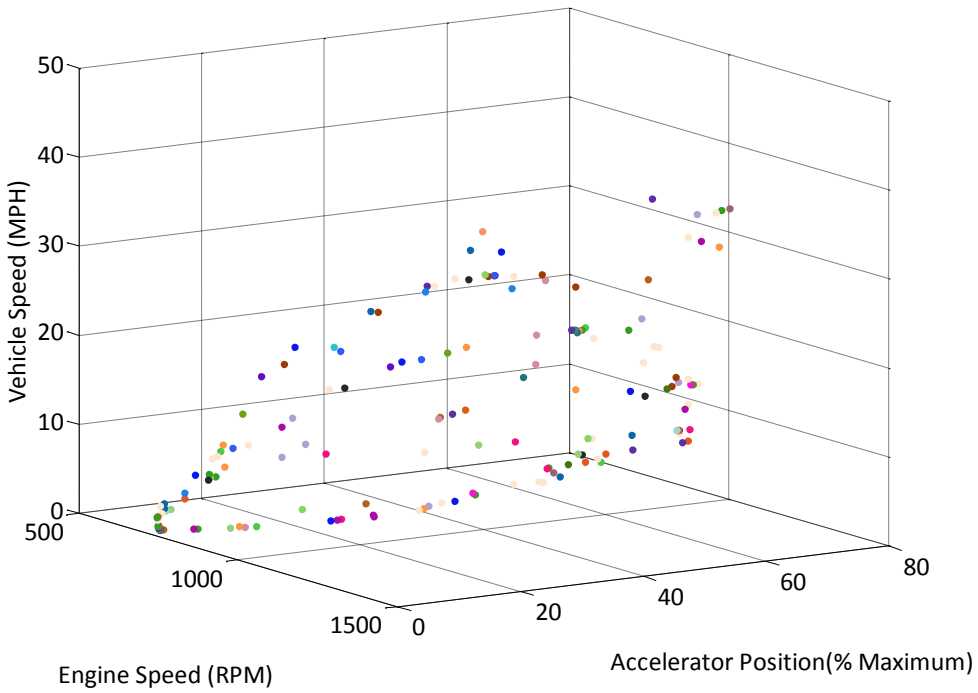

Figure 46: Color-coded 3D scatter plot of the three operational mode variables

It can be difficult to tell on a 3D scatter plot where each point lies, especially when the plot cannot be interacted with and rotated. For this reason, it can also be useful to have a table sorted by group number. Table 4 shows three groups from this data set and Table 5 shows the mean and standard deviation for each variable in each of the three groups.

The small standards deviation values indicate that the groups are tightly spaced. The number of operational mode groups of the vehicle is arbitrary, and more or less could be used by changing the value of *k*.

Table 4: Three of the vehicle mode groups and the operation variables

Engine Speed (RPM)	Accelerator Position (% Max)	Vehicle Speed (mph)	k-means Group #
800	11	1	1
778	3	4	1
811	11	1	1
808	3	7	1
774	3	5	1
793	11	1	1
1269	31	14	2
1273	18	17	2
1281	19	17	2
1177	41	9	3
1159	37	7	3
1174	21	12	3
1181	37	19	3
1177	42	9	3

Table 5: Mean and standard deviation for engine speed, accelerator position and vehicle speed in the first three vehicle operational mode groups

	Engine Speed		Accelerator Position		Vehicle Speed	
Group Number	Mean (RPM)	Standard Deviation (RPM)	Mean (%)	Standard Deviation (%)	Mean (mph)	Standard Deviation (mph)
1	794	15	7	4	3	3
2	1274	6	22	7	16	2
3	1174	9	35	8	11	5

4.3 K-means Algorithm Conclusions

It was shown that the k-means algorithm can successfully group the vehicle operational variables into modes by checking the standard deviation of the variables in a group. In the data examined, the standard deviations were small, meaning the data was grouped tightly. The number of modes desired is arbitrary and decided by the user. These modes by themselves can be used to improve fuel economy and reduce system damage by looking at how often the vehicle is changing modes and by examining time spent in a particular mode.

The following chapter explores a method for correlating the TRIC classifications to the vehicle operational modes which could be used to help understand operator response to a terrain. This could be helpful in reducing driver fatigue and system damage even further.

Chapter 5- Terrain Classifications and Vehicle Operational Modes Comparison

The vehicle modes discussed and analyzed in the previous chapter are called operational modes because they depend on how the vehicle is being operated by the driver. A question that has arisen is how terrain affects operation of a vehicle. This helps vehicle operators be more aware of how terrain can affect their driving and allow them to adjust their driving. It also helps in identifying improvements to vehicles that will increase operator comfort and increase performance over harsh terrain. Knowledge of terrain effects on vehicle operation and performance will help with planning for maintenance, repair and logistics because the terrain will affect failures in the suspension system, tire wear and the fuel use.

5.1 Methods of Comparison

To compare the terrain classifications to vehicle operational modes, a correlation between the two sets needs to be made over time. A histogram is a useful tool for showing how often specific data occurs over time. A two-dimensional histogram can be used to show how often specific data in one set occurs at the time as specific data of another set. The two-dimensional histogram can be used to show how often data with a specific TRIC classification occurs at the same time as data grouped into a specific vehicle operational

mode. A spreadsheet can also be used to correlate the two classifications. The rows would represent the TRIC classifications or the vehicle operational modes, and the columns would represent the other. Each time a specific TRIC classification occurs at the same time as a specific vehicle operational mode, a value of one would be added to the corresponding cell in the spreadsheet. The hope in using these methods is that specific TRIC classifications will strongly correlate to specific vehicle operational modes.

MATLAB was used to make a two-dimensional histogram of the TRIC classifications and vehicle operational modes. These histograms are shown in the following section. MATLAB does not have a built-in function for a two-dimensional histogram, but the built-in function `accumarray` can serve the same purpose. The syntax for the `accumarray` function is `A = accumarray(subs,val,sz)` where `subs` is a matrix containing subscript values, `val` is a vector of elements, `sz` is the size of the output matrix, `A`, in the format of `[a,b]`. The matrix `A` sums the elements from the `val` vector according to the subscripts in the subs matrix [20]. For the case of the two-dimensional histogram, `subs` is a 2 column matrix with one column as the TRIC classifications and the other as the vehicle operational mode number from the k-means grouping. Each row of the matrix has the same time. In the case of this research, both classification systems use 20-second averages of the data, so they line up in time. But neither the TRIC algorithm nor the method used to obtain the vehicle operational modes must use 20 second averages. This depends on the preference of the user. If either the TRIC classifications of the vehicle operational modes do not use 20 second records, decimation can be used to make the

times the same. The vector `val` would be a 1-by-1 matrix with a value of 1, and `sz` is the number of classifications in the TRIC algorithm and the number of the k-means groups. The function goes through `subs` and finds each instance of each possible combination of the TRIC classification and the operational mode group number. For each instance it finds, it uses the value of `val`, one, and adds it to the corresponding subscripts of the matrix A. For example, for the (1,1) entry in the matrix `A`, the `accumarray` function finds all instances where both the TRIC algorithm and the k-means group number for the operational modes are both 1, and adds one for each instance. Then the contour function can be used to graphically represent the histogram. The TRIC classifications are plotted on one axis, the operational modes on the other axis, and the contours will show how many reoccurrences of each combination is present. A 3D surface plot can be used, but the contour is easier to see and read with static plots.

5.2 Terrain Classifications Compared to Vehicle Operational Modes

5.2.1 MTVR on NATC Paved Oval Track

To show this comparison, the same test from the last chapter, using a loaded MTVR on a paved oval track, is examined. In this test the terrain was uniform and not harsh. Figure 47 shows a map of the course overlaid with the color-coded TRIC classifications. To make things easier to read, only seven colors are used on this map.

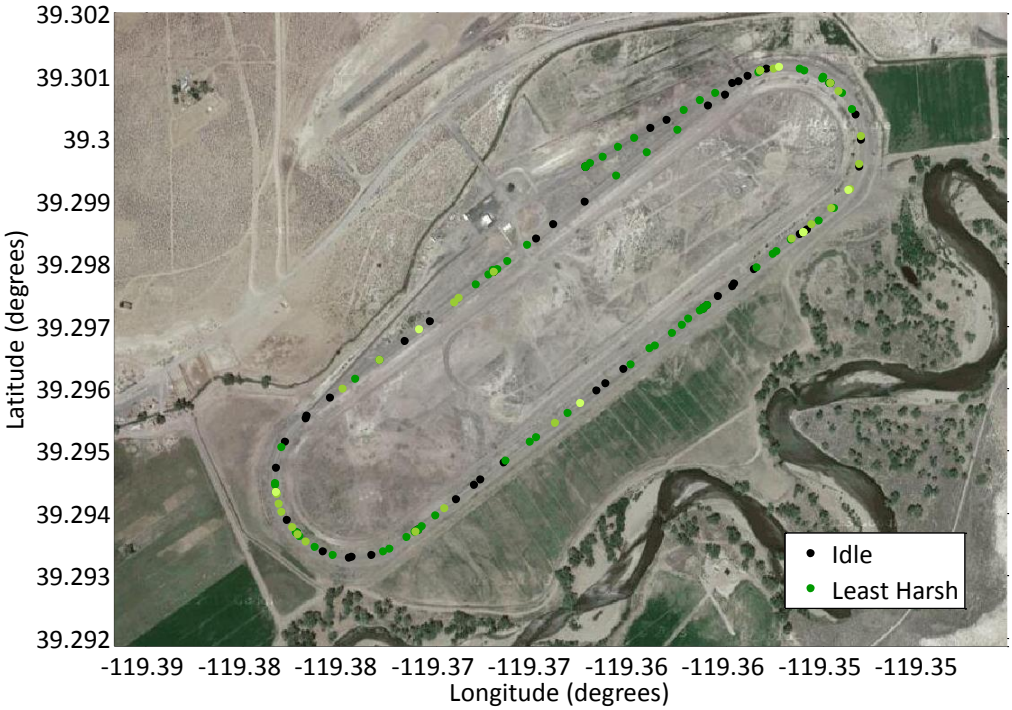

Figure 47: Map overlaid with the terrain classifications of a loaded MTVR vehicle travelling around an oval track

Figure 47 confirms that this terrain is uniform and not harsh. There are only 3 shades of green in the figure; any shade of green is considered low harshness. The classifications appear to change most frequently at or around the curves of the track. The black dots represent where the vehicle is not moving or where the speed is below 5 mph.

Rather than directly comparing maps of terrain classifications and vehicle operational modes, the two-dimensional histogram discussed in the last section is easier to read and to see correlations. Figure 48 shows a map with the color-coded vehicle operational modes (circles) overlaid with color-coded TRIC classifications (squares). The TRIC classifications

are slightly offset. Even with the offset, the plot is cluttered and it is difficult to make any correlations between the two sets of classifications.

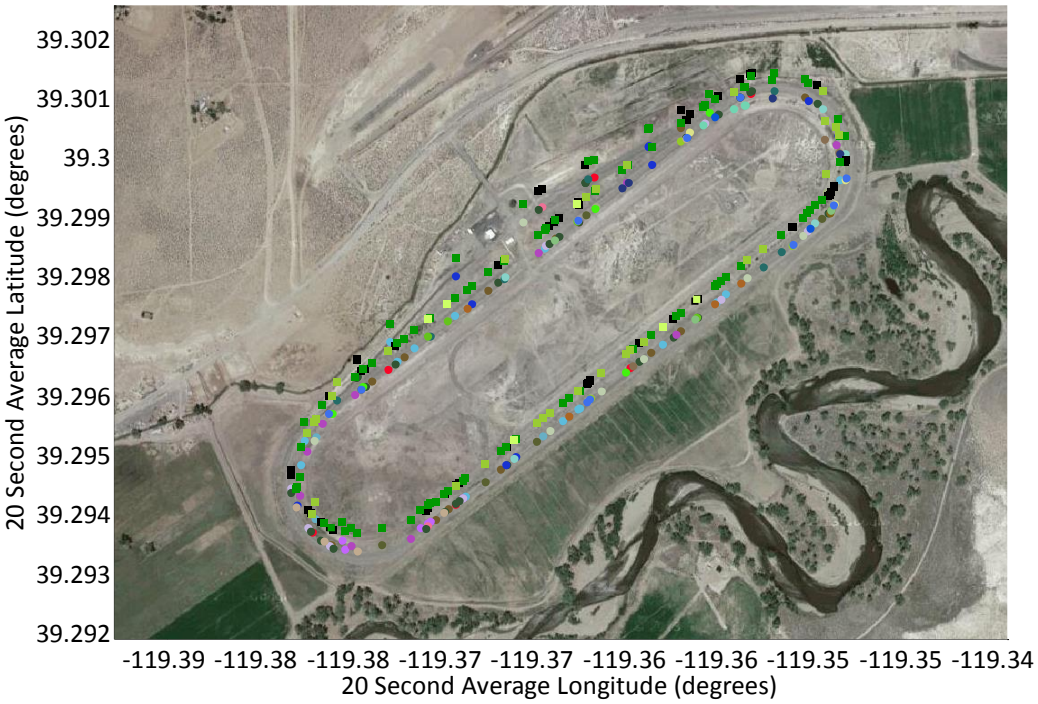

Figure 48: Map of NATC oval track overlaid with vehicle operational modes (circles) and TRIC classifications (squares)

Figure 49 shows the contour plot histogram of this test, and Figure 50 shows the 3D surface plot histogram. The terrain classifications and operational modes that have a high number of matches are circled in red on the contour plot. The circled groups correspond to TRIC group 1 with vehicle mode (VM) group 6, TRIC group 1 and VM group 15, TRIC group 5 and VM group 23 and TRIC group 11 and VM groups 12 and 13. Appendix D contains a table with all of the VM variable (engine speed, accelerator positon and vehicle speed) data and the assigned group number for each data point.

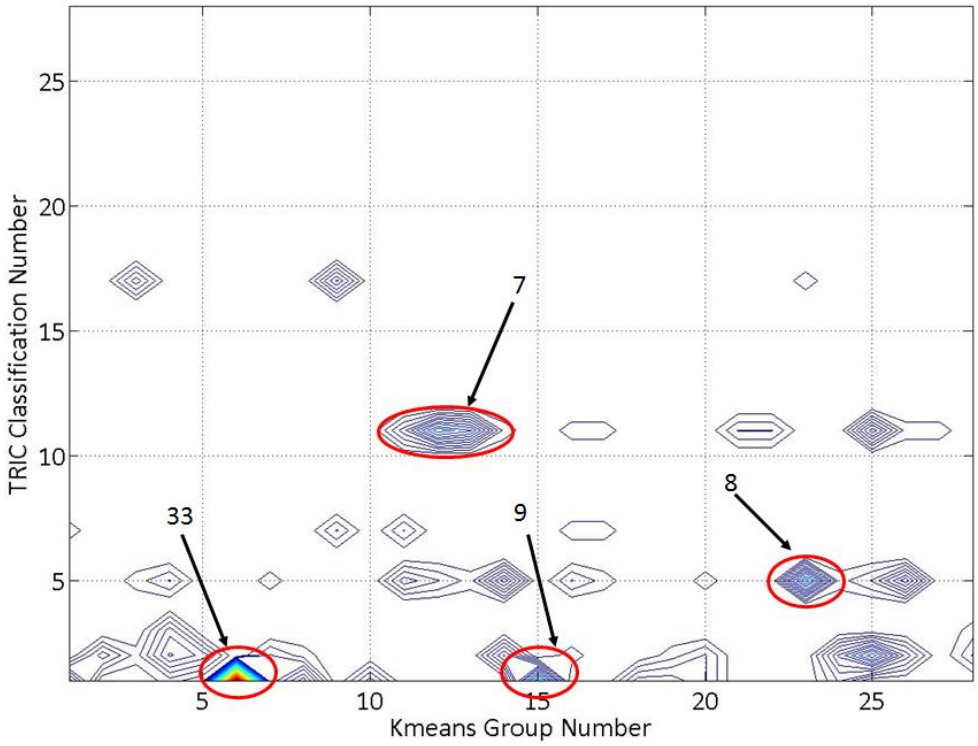

Figure 49: Contour histogram showing the correlation between the terrain classifications and the vehicle operational modes

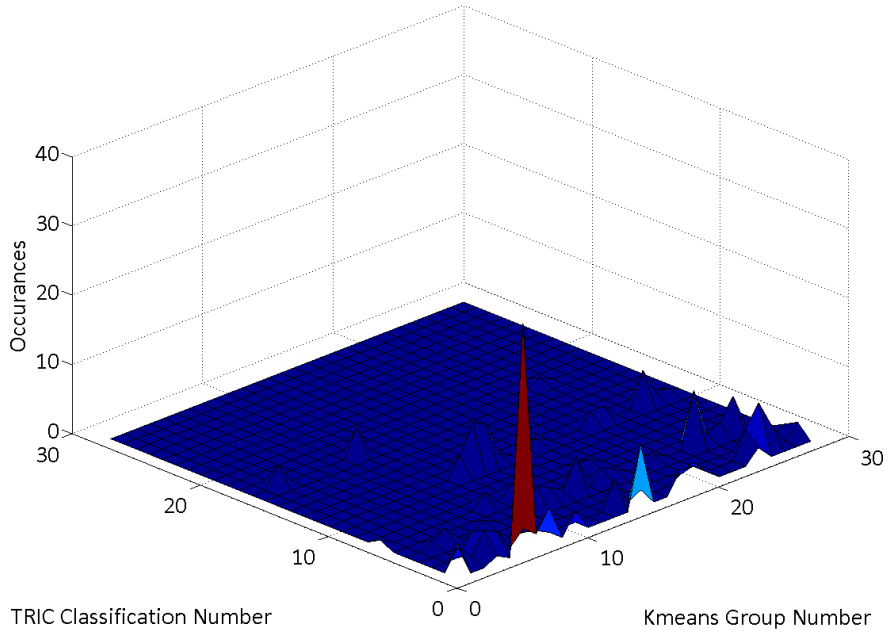

Figure 50: 3D surface plot histogram showing the correlation between the terrain classifications and the vehicle operational modes

In this data set, there are only seven of the possible 28 terrain classifications. Table 6 shows the classifications that are present and what type of terrain they are.

Table 6: Terrain classification groups for the MTVR on the oval paved track at NATC and a description of each terrain type

TRIC Classification Number	Short Wavelength Amplitude	Medium Wavelength Amplitude	Long Wavelength Amplitude
1	None	None	None
2	Low	Low	Low
3	Mid	Low	Low
5	Low	Mid	Low
7	Mid	Mid	Low
11	Low	High	Low
17	Mid	High	Low

A histogram of VM groups for just one TRIC group can show more detailed correlations. Figure 51 shows a histogram of the VM groups for the seven TRIC classifications from this data set. TRIC group 1 is very strongly correlated with VM group 6. Both groups correspond to the vehicle idling. While these groups are not significant for vehicle maintenance, it shows that the k-means algorithm is grouping the operational variables into groups that make sense. Table 7 summarizes the strongest correlations. It shows the TRIC group number and the VM group number that are correlated, the percent of the total number of that specific TRIC classification that was matched with the correlated VM group, and the percent of the total number of that specific VM group that was matched with the correlated TRIC classification.

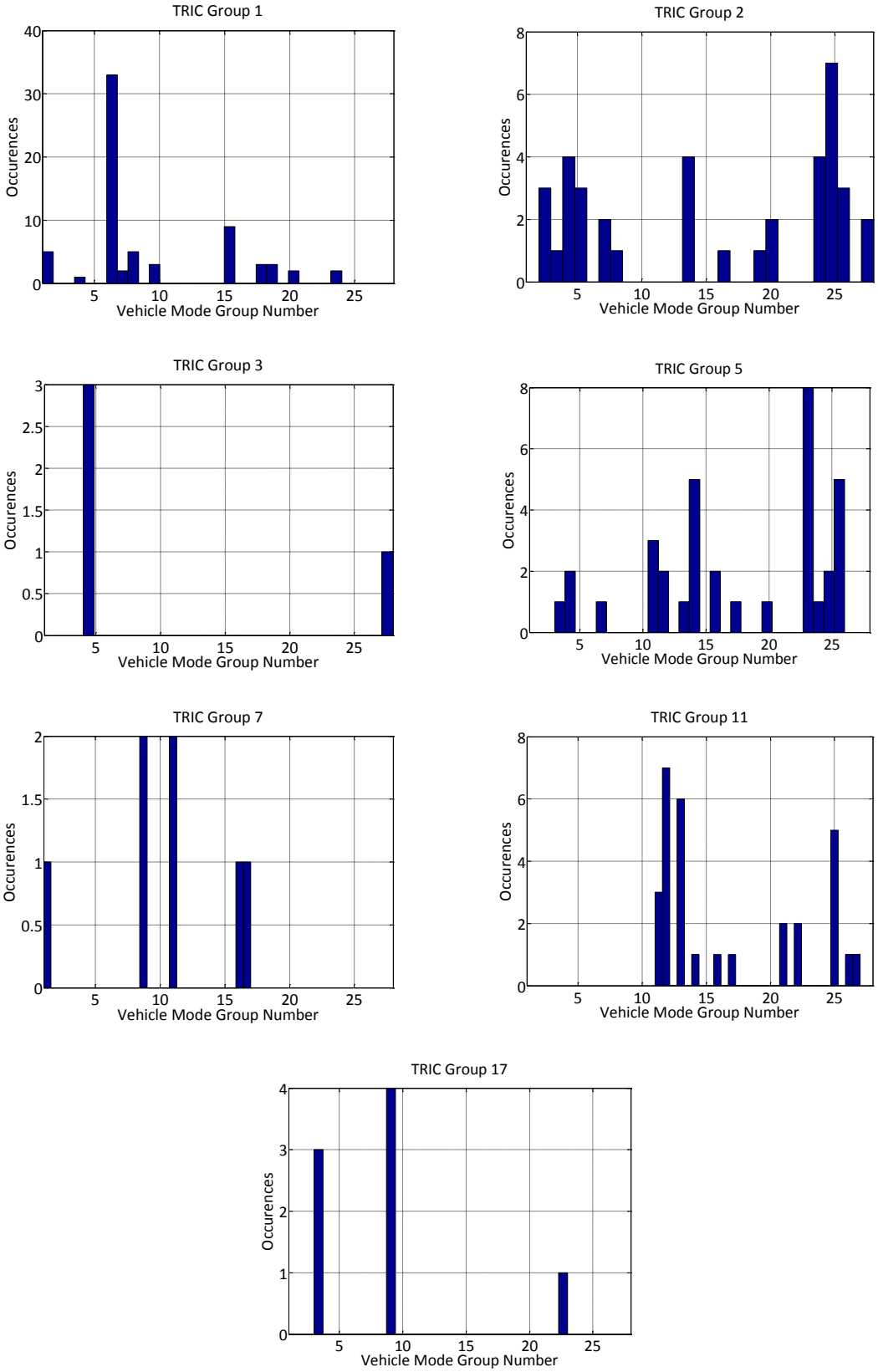

Figure 51: Histogram of the VM groups for each TRIC group for the MTVR test

Table 7: TRIC groups and VM groups that showed the strongest correlation for the MTVR test on the paved oval track

TRIC Group	Vehicle Mode Group	TRIC Classification Percentage	Vehicle Mode Classification Percentage
1	6	49%	100%
1	16	13%	100%
2	25	19%	50%
5	23	23%	89%
11	12	23%	78%
11	13	20%	66%

TRIC groups 2 and 5 were the most common classifications in this data set. They correspond to smooth, flat, level terrain and smooth, medium bumpiness, level terrain respectively. They correlated strongest with VM groups 23-26. These groups combined had an average engine speed of 1304 RPMs, 39% pedal depression, and speed of 20 mph. The VM groups with the highest average speeds were groups 12 and 13 (30 mph and 27 mph respectively) and they both correlated strongest with TRIC group 11. TRIC group 11 is classified as smooth, very bumpy and level. This correlation shows that speed may affect the medium wavelength classifications, and this may need to be taken into account in the TRIC algorithm thresholds in the future.

Looking back at Table 7, one of the most noticeable things is that a majority of the classifications from a vehicle mode group matched with the same terrain classification group. But the opposite was not true. This is most likely due to the fact that the number of groups from the k-means algorithm is user defined. There will be 28 groups no matter

what. However, the terrain classification groups are parameter defined, meaning the data has to meet a criteriion to be classified into one group or another. Even though there are 28 different classifications, some groups may not have any data points. So there will always be 28 groups in the vehicle modes that each have at least one data point, but in the terrain classifications some groups may be empty. From this data taken from simple terrain, it can be seen that there is correlation between the terrain classifications and the vehicle operational modes.

5.2.2 HEMTT on APG Churchville Course

The next data set contains more varied terrain. It was taken from a HEMTT operating on the Churchville test course at APG in Maryland. Referring back to Chapter 2, Figure 17 shows the terrain classifications over the map of the test course. This course includes low-harshness terrain that is smooth, flat and level all the way up to harsh terrain that is rough, bumpy and steep. Figure 52 shows a plot of the color-coded VM groups overlaid on a map of the test course. Figure 53 shows the color-coded histogram of the VM groups for the data. See Appendix D for a table of the VM variables and their classification.

To compare the TRIC classifications to the VM groups, a two-dimensional histogram was done. The contour plot is shown in Figure 54 and the surface plot is shown in Figure 55.

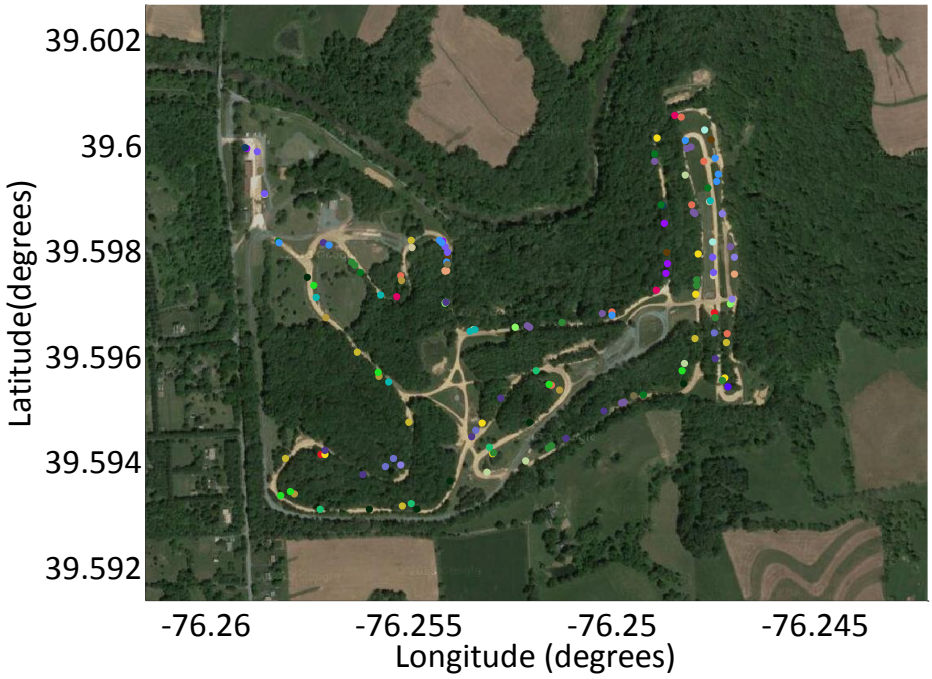

Figure 52: Vehicle operational modes for the HEMTT operating on the Churchville course at APG

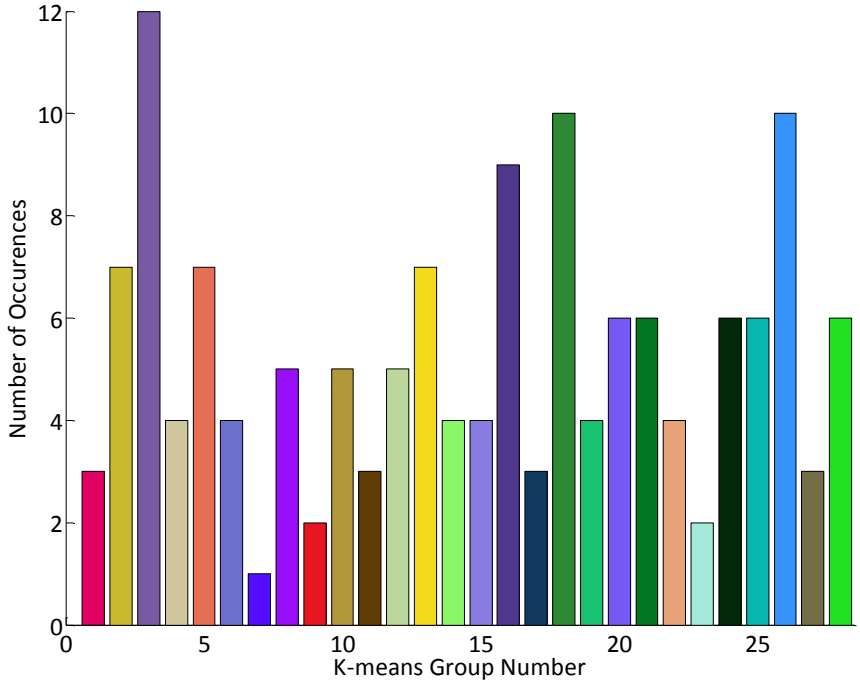

Figure 53: Color-coded histogram of the HEMTT Churchville VM modes

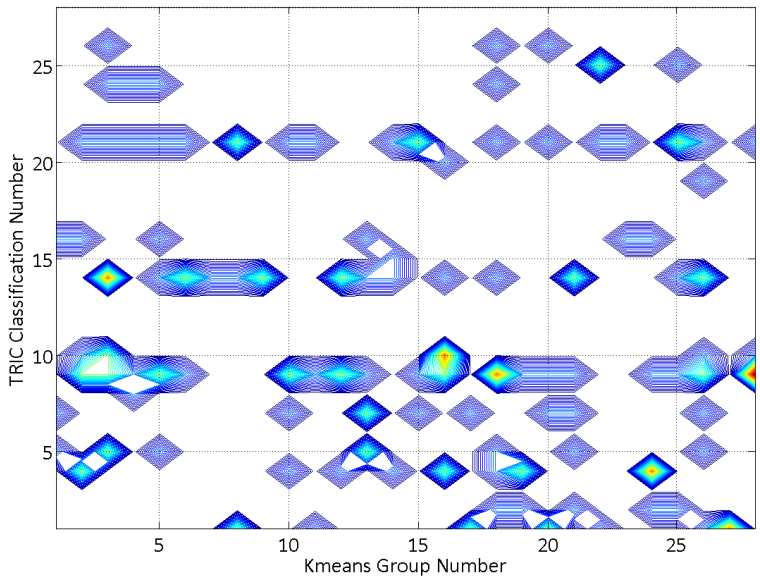

Figure 54: Contour plot histogram of the TRIC classifications and the VM groups for the Churchville HEMTT test

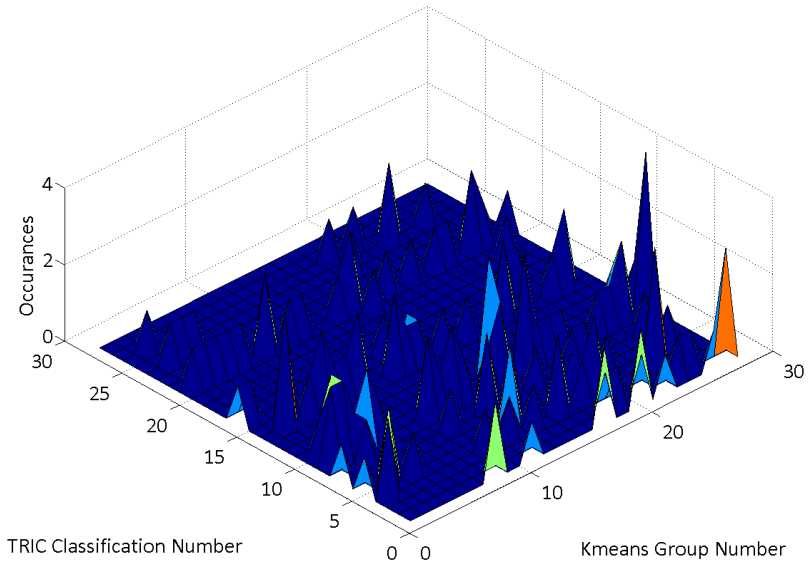

Figure 55: Surface plot histogram of the TRIC classifications and the VM groups for the Churchville HEMTT test

Because of the increase in TRIC classifications, the more varied terrain does not give as many matches between the two classifications as the previous test did. However, there are patterns present. Table 8 lists the TRIC classifications present in the data and the corresponding terrain type.

Table 8: TRIC classifications present in the Churchville HEMTT data and the corresponding terrain type

TRIC Classification Number	Short Wavelength Amplitude	Medium Wavelength Amplitude	Long Wavelength Amplitude
1	None	None	None
2	Low	Low	Low
4	Low	Low	Mid
5	Low	Mid	Low
7	Mid	Mid	Low
8	Mid	Low	Mid
9	Low	Mid	Mid
10	Low	Low	High
14	Mid	Mid	Mid
15	Mid	Low	High
16	Low	Mid	High
19	High	Mid	Mid
20	High	Low	High
21	Mid	Mid	High
24	Mid	High	Mid
25	High	Mid	High
26	Mid	High	High

Because of the large number of TRIC classifications in the data, only the VM group histograms for the specific TRIC groups that have the strongest correlation were produced. They can be seen in Figure 56. It is important to remember that the k-means assignments are random, so the VM groups for this data set are not the same as the VM groups of the data set in the previous section.

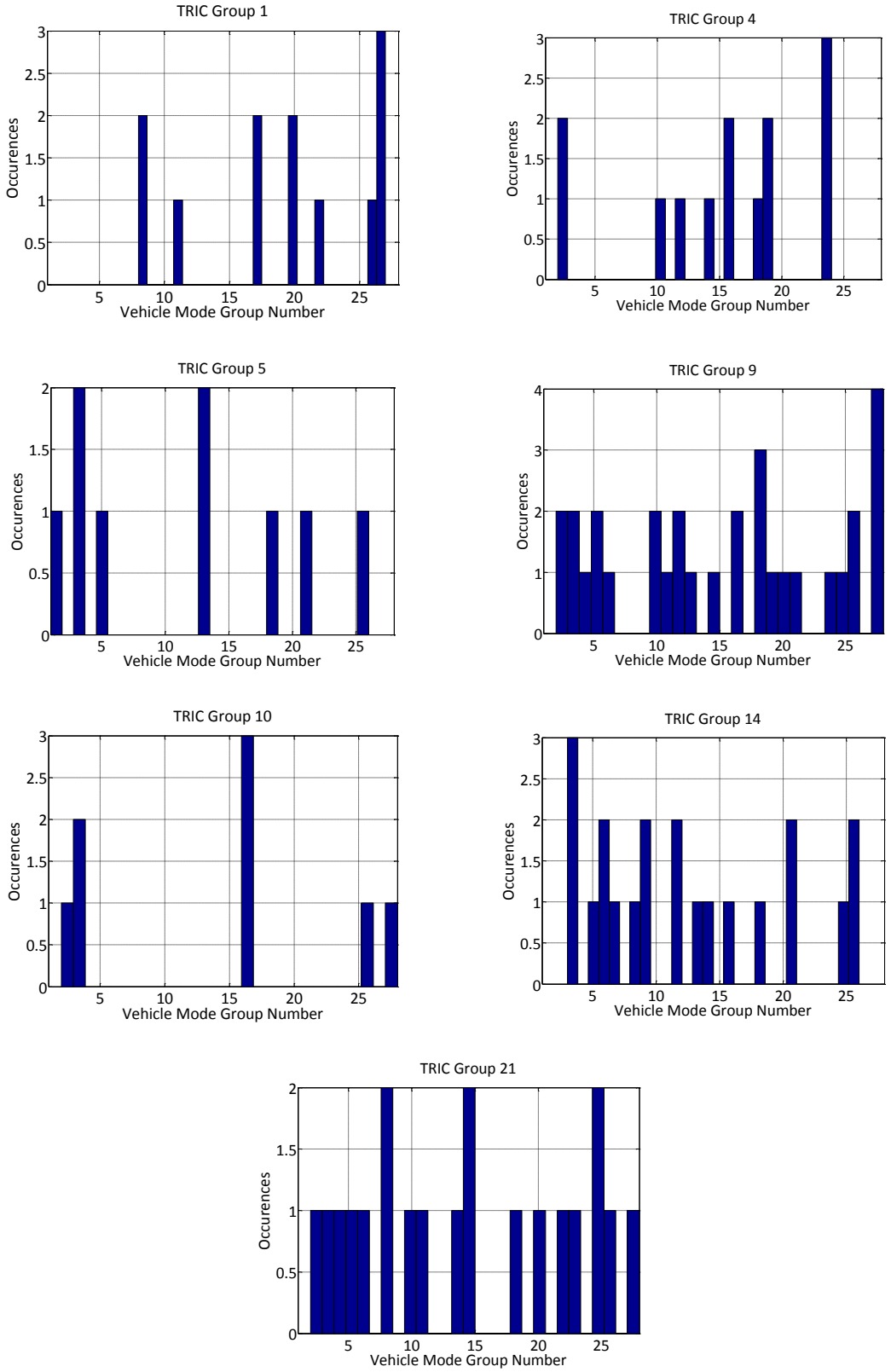

Figure 56: Histograms of the VM groups for select TRIC classifications from the Churchville HEMTT test

The data points were most commonly classified as TRIC group 9, 14, or 21. TRIC group 9 corresponds to smooth, partly bumpy, and partly steep terrain. Group 14 corresponds to terrain that is partly rough, partly bumpy, and partly steep. Group 21 is similar terrain, but is very steep. Table 9 shows which VM groups were most correlated with these TRIC groups and the averages for engine speed, accelerator position, and vehicle speed for the VM groups.

Table 9: TRIC groups with the most classifications, most correlated VM modes, and the averages VM variables for those groups for the Churchville HEMTT test

TRIC Group	Vehicle Operational Modes	Average Engine Speed (RPMs)	Average Accelerator Position (%Maximum)	Average Vehicle Speed (mph)
9	18 & 28	1622	56	20
14	3, 6, 9, 11 & 26	1423	64	14
21	8, 15 & 25	1395	27	11

In the groups shown in Table 9, as the terrain becomes harsher the vehicle speed drops and the RPMs decrease. In TRIC group 14, where the amplitude of all three terrain wavelengths was mid-level, the accelerator position was most depressed. This could indicate that the vehicle operator became aggressive as the terrain became a little harsher, but on terrain classified as TRIC 21 where the terrain was steep, the operator backed off.

This test, like the previous test, shows that correlations can be made between the TRIC classifications and the VM groups, even when the terrain is more varied.

5.2.3 Correlation Test Conclusions

Both the MTVR test on the oval paved course and the HEMTT test on the Churchville course showed that the TRIC classifications could be correlated to the VM groups. These correlations can be used to discover how operators are responding to the terrain and how that is affecting vehicle performance. It can also help in predicting maintenance and repair schedules by knowing how a vehicle is operating on different terrains.

A problem encountered was that specific TRIC groups correlated to several of the VM groups. This is because when using the k-means algorithm, k groups are filled, whereas the TRIC groups are only filled if the terrain meets the requirements. Future work could include some potential solutions to this problem. First, the k parameter (number of clusters) can be adjusted by the user manually. This approach may be time consuming and may be difficult to do correctly. Second, a clustering algorithm could be developed that sorts the data into a number of groups decided on by the algorithm, rather than defined by the user. Lastly, training data could be used. Training data is used to develop classifications for a certain number of classes. Once the classes are defined, the rules can be applied to new data, but the new data does not have to occupy all classes defined.

Chapter 6 - Summary and Conclusion

6.1 Summary and Conclusions

The Terrain Regime Identification Classification (TRIC) algorithm was developed by the Army Materiel Systems Analysis Activity (AMSAA) to better understand the effects of terrain on vehicle damage and predict maintenance schedules. TRIC does more than just profile the terrain, but classifies the vehicle's response to the terrain. The terrain is divided into three regimes – short wavelength, medium wavelength and long wavelength. The short wavelength regime includes short, small bumps such as gravel or dirt road. It is measured with the z-axis of an accelerometer. The medium wavelength regime includes larger bumps that will cause the vehicle to pitch (tip forward or backward) and roll (tip side-to-side). It is measured with the pitch and roll output of a gyroscope. The long wavelength regime includes the steepness of the terrain, or how hilly it is. It is measured from the speed and altitude output of a GPS device. Each regime is classified as either low, mid or high amplitudes based on vehicle specific thresholds. When combined, there are 27 different possible classifications of terrain harshness. As data is collected on a fleet of vehicles, terrain harshness can be correlated to maintenance and maintenance schedules can be better predicted.

The algorithm from AMSAA uses an accelerometer mounted to the axis of the vehicle, referred to as an unsprung accelerometer, to measure the up and down accelerations used

to classify the short wavelength terrain. It would be cheaper and more convenient to replace the unsprung accelerometer with one placed above the suspension, referred to as a sprung accelerometer. A process of computing and applying a transfer function to the data from a sprung accelerometer to estimate and the data from the unsprung accelerometer. This process was carried out on two different data sets. One data set was obtained from sensors mounted on a Medium Tactical Vehicle Replacement (MTVR) driven over a course at the Nevada Automotive Test Center (NATC). The other data set was obtained from sensors on an MTVR driven in the hills near State College, PA. In the NATC test, the sprung accelerometer was located inside the cab of the truck, and in the State College test the sprung accelerometer was mounted on the frame of the truck above the suspension. The results for the NATC data showed that terrain classifications are improved when the transfer function is applied to the sprung data, but the results were still poor; only 52% of the classifications were the same as the classifications from the unsprung accelerometer. Other NATC tests showed up to 84% of the classifications from the unsprung estimate data as being the same as the classifications from the unsprung accelerometer, but results were not consistent. The State College test showed that classifications from the estimated unsprung data matched the original unsprung data 97% of the time. This huge improvement from the NATC tests was because the accelerometer was mounted on the frame of the vehicle above the suspension rather than in the cab of the vehicle, improving the coherence between the two accelerometers.

A method of vehicle operational mode classification was explained. This method used a k-means clustering algorithm to group three operational mode variables into groups. These variables were engine speed, accelerator position and vehicle speed. Different methods of displaying these vehicle modes were produced. Data from a test done at NATC with an MTVR were used to produce a map overlaid with color-coded points representing the different vehicle modes. This type of display is useful for seeing when the vehicle is changing modes. A color-coded histogram and a pie chart were also produced for the data, showing how often each mode occurred. These can be used alongside a table of the groups and the data points belonging to each group to see how the vehicle is operating.

Once the vehicle operational mode groups were produced, they were compared to the classifications from the TRIC algorithm. This was done to see if there was any correlation between terrain and how a vehicle is operated. Knowing correlations can help vehicle operators adjust their driving on different terrains. It can also help with operator comfort, planning for maintenance, repair and logistics because the terrain will affect failures in the suspension system, tire wear and the fuel use. The correlations were done by using a two-dimensional histogram. Two tests were looked at in this case – an MTVR test on a paved oval track at NATC and a HEMTT test done on the Churchville course at APG. The MTVR test showed that there were correlations between the terrain classifications and the vehicle modes. A majority of the classifications from a vehicle mode group matched with the same terrain classification group, but the opposite was not true. This is because all of

the groups from the k-means algorithm were populated, but only seven of the TRIC classifications were populated. The HEMTT test also showed correlations between terrain classifications and vehicle modes. In this test, more of the TRIC classifications were populated because the terrain was more varied. However, the test still showed terrain classifications being correlated to more than one group. For the long wavelength terrain (steep hills), this would be expected because the TRIC algorithm does not distinguish between uphill and downhill, but the vehicle operational modes would. In terms of maintenance, this would not be of much importance, because on average there would be as many uphill roads as downhill. Despite this, there was little overlap between k-means groups correlating to more than one TRIC group.

6.2 Recommendations for Future Work

In this research, the tire, unsprung mass, suspension and sprung mass were not modelled in detail. A more detailed model may show better how to relate the data from the unsprung accelerometer to the sprung accelerometer. Knowing vehicle weight and tire pressure, as well as the spring and damping properties of the suspension, can reduce sensitivity to these factors and provide a more accurate transfer function. Thresholds can also be defined for the sprung accelerometer data (without the transfer function applied) the same way that they are defined for the unsprung accelerometer data. The Army currently has test courses defined as primary roads, secondary roads, and off-road for the short wavelength regime. The data from the sprung accelerometer can be compared to sections of courses defined as primary, secondary and off-road, and thresholds can be

determined so that the data from sections of primary road become low amplitude in TRIC, secondary roads become mid amplitude in TRIC, and data from off-road sections become high amplitude in TRIC.

A problem encountered while correlating vehicle operational modes to the TRIC classifications was that specific TRIC groups correlated to several of the vehicle mode groups. This is because all k-means groups are populated, whereas the TRIC groups are only populated if the terrain falls within the thresholds. Future work could include some potential solutions to this problem. First, the *k* parameter (number of clusters) can be adjusted by the user manually. This brute force approach may be time consuming and may be difficult to do correctly. Second, a clustering algorithm could be developed that sorts the data into a number of groups decided on by the algorithm, rather than defined by the user. This type of clustering is called unsupervised. Another approach would be to use training data to develop classifications for a certain number of classes. Once the classes are defined, the rules can be applied to new data, but the new data does not have to occupy all classes defined. Because there was no clear one-to-one correlation between the output of the TRIC algorithm and the vehicle operating modes determined through k-means clusters, another approach to consider is to use both the TRIC classifications and the vehicle modes in a classifier to predict maintenance requirements and fuel efficiency.

This research focused on the use of the TRIC algorithm with vehicles, specifically military vehicles. There is also interest in using the algorithm on robots primarily to see how the terrain is affecting battery power consumption. Figure 57 shows the short wavelength

classifications for a Tankbot robot run done on grass. In this preliminary study, the thresholds were done be eye looking at were natural divisions in the data may be. To use TRIC on the robots, a more refined way of defining the thresholds will need to be developed.

Figure 57: Short wavelength classifications from a Tankbot robot test done on grass. The thresholds were done by eye and a more accurate method of developing thresholds is being developed

Also, the Robot Operating System (ROS) used on many robots can only process programs in languages like Python and C++. An implementation of the TRIC algorithm in the Python language is currently in development.

6.3 Final Comments

The purpose of the TRIC algorithm is to help improve vehicle life, prevent damage and system failure, better predict maintenance schedules, and improve operator performance and comfort based on the interaction of the vehicle with the terrain it traverses. This research looked at improving the TRIC algorithm and in ways to make it more useful. Ultimately, before the TRIC algorithm can fulfil its purpose, more data will need to be collected from the field and correlations to component damage and system failure will need to take place. While this research did not correlate the TRIC classifications to vehicle maintenance, ways were presented in which the algorithm can be improved to better fulfill this purpose in the future.

References

[1] Wells, S. 2013. (2013, May). *Vibration-Based Sensor Design to Detect Lubrication Levels Contained Within Differential Gear Housings* (Master's thesis). Pennsylvania State University, University Park.

[2] Hatton, K. (2011, March 2). *Terrain Regime Identification and Classification (TRIC)* [PowerPoint]. Australian International Aerospace Congress HUMS 2011. Melbourne, Victoria, Australia.

[3] Hatton, K. (2011, March). *Terrain Regime Identification and Classification for Condition Based Maintenance*. Presented at the Fourteenth Australian International Aerospace Congress, Melbourne, Victoria, Australia.

[4] Steinwolf, A. and Connon, W. (2005, February). *Limitations of the Fourier Transform for Describing Test Course Profiles.* Sound & Vibration, pp. 12-17.

[5] González, A., O'Brien, E., Li, Y. and Cashell, K. (2008, June). The use of vehicle acceleration measurements to estimate road roughness. *Vehicle System Dynamics*, 46, pp. 483-499.

[6] Ashmore, S. and Hodges, H. (1990, October 28). *Dynamic Force Measurement Vehicle (DFMV) and Its Application to Measuring and Monitoring Road Roughness.* Presented at the conference of Vehicle, Tire, Pavement Interface, Santa Barbara, CA.

[7] Chriss, C. (1998-2014). US Military HEMTT. *Olive-Drab.* Retrieved July 22, 2014, from http://olive-drab.com/idphoto/id_photos_hemtt.php

[8] Chriss, C. (1998-2014). FMTV: Family of Medium Tactical Vehicles. *Olive-Drab.* Retrieved July 22, 2014, from http://olive-drab.com/idphoto/id_photos_fmtv.php

[9] Churchville Test Area. (n.d.) In Wikipedia. Retrieved May 2014, from http://en.wikipedia.org/wiki/Churchville_Test_Area

[10] Hershey, C. (personal communication, August 5, 2013)

[11] Chriss, C. (1998-2014). US Navy & USMC Medium Tactical Vehicle Replacement (MTVR) 7 ton 6x6. *Olive-Drab.* Retrieved July 22, 2014, from http://olive-drab.com/idphoto/id_photos_mtvr.php

[12] Gabrielson, T. *Chapters 1 and 2*. Acoustics 516 Class Notes. Pennsylvania State University, University Park. Print.

[13] Oppenheim, A., Schafer, R., and Buck, J. (1999). *Discrete-Time Signal Processing*. Upper Saddle River, NJ: Pearson Prentice Hall.

[14] Shynk, J. (1992, January). *Frequency-Domain and Multirate Adaptive Filtering.* IEEE SP Magazine, pp. 14-37.

[15] Balough, L., Mészáros-Komáromy, Á., Palkovics, L. *Road Surface Estimation for Control System of Active Suspensions.*

[16] Jazar, R. (2014). *Vehicle Dynamics: Theory and Application*. New York, NY: Springer

[17] Wagstaff, K., Cardie, C., Rogers, S., and Schroedl, S. (2001). *Constrained K-means Clustering with Background Knowledge.* Proceedings of the Eigthteenth International Conference on Machine Learning. Pp. 577-584.

[18] The MathWorks, Inc. (1994-2014). kmeans. *MathWorks.* Retrieved July 15, 2014, from http://www.mathworks.com/help/stats/kmeans.html?s_tid=doc_12b

[19] Bar-Yehuda, Z. (2010). plot_google_map.m. Retrieved 2014, February 27 from http://www.mathworks.com/matlabcentral/fileexchange/27627-plot-google-map

[20] The MathWorks, Inc. (1994-2014). accumarray. *MathWorks.* Retrieved May, 2014, from http://www.mathworks.com/help/matlab/ref/accumarray.html

[21] Serena, Paolo. "Depiction of overlap-add algorithm." (original, released for free use) - en wikipedia, derived from w:en:File:Oa idea.jpg by User: Paolostar, Paolo Serena, University of Parma (Italy). Licensed under Public domain via Wikimedia Commons - http://commons.wikimedia.org/wiki/File:Depiction_of_overlap-add_algorithm.png#mediaviewer/File:Depiction_of_overlap-add_algorithm.png

Appendix A- Wavenumber Spectrum Conversion

Because the TRIC algorithm is based on the wavelengths of terrain to make classifications, it may be useful to use wavenumber spectrums (WNS) to define thresholds and divisions between the wavelength regimes. A WNS is produced analogously to a power spectral density (PSD). The big difference being that the data is sampled in equal divisions of distance, rather than in equal divisions of time as is done when transforming to the frequency domain. When it is not convenient or possible to sample data spatially, it can be difficult to obtain the WNS.

There is a simple method that can be used to obtain a WNS from the PSD, for a vehicle traveling at a constant speed. In a paper from 2000, William Connon describes this method for converting a WNS to a PSD [19], but it can be done in reverse as well. This process in reverse is described here. First, the wavenumbers can be obtained using Equation (A.1):

$$k = f/v \tag{A.1}$$

Where k is the wavenumber in cycles/ft, f is the frequency in Hz, or cycles/sec and v is the constant speed of the vehicle in ft/sec. Then, the wavenumber spectrum can be obtained by using Equation (A.2):

$$G_{wn} = G_f * v \tag{A.2}$$

Where G_{wn} represents the wavenumber spectrum values, G_f represents the power spectral density values and v is again the constant speed of the vehicle. The units of G_{wn} and G_f depend on the measurement of interest. If one was wanting to convert the PSD from a z-axis accelerometer to a WNS, the units of G_{wn} would be g^2/(cycles/ft) and the units of G_f would be g^2/Hz. Once G_{wn} and k are obtained, the WNS can be plotted. Figures 58 and 59 show the PSD and the resulting WNS from a warm up test done with an MTVR on a paved oval track at NATC. This test was chosen because the vehicle's speed was constant for most of the test. One-thousand seconds of data where the speed was constant, about 74 ft/sec, were used to produce the PSD and resulting WNS.

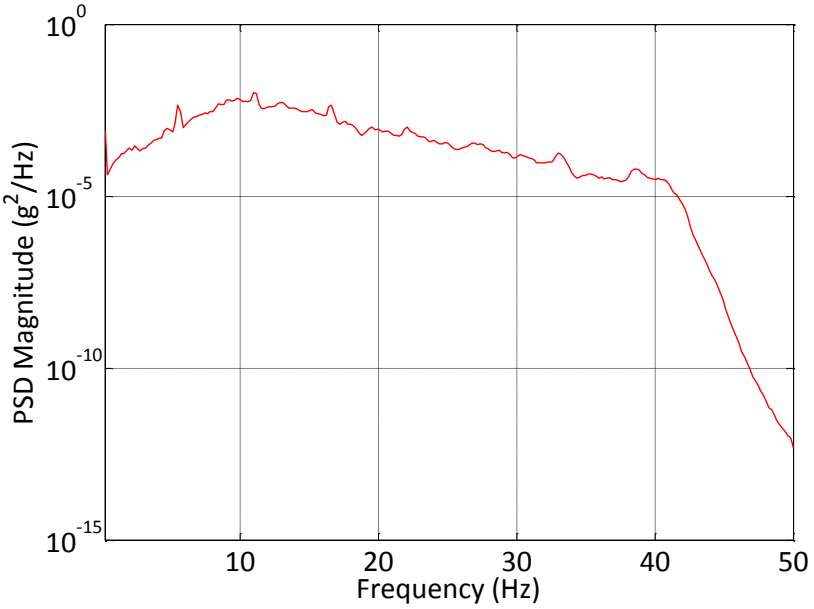

Figure 58: PSD of 1000 seconds of a warm up test with an MTVR at a constant speed of about 74 ft/sec

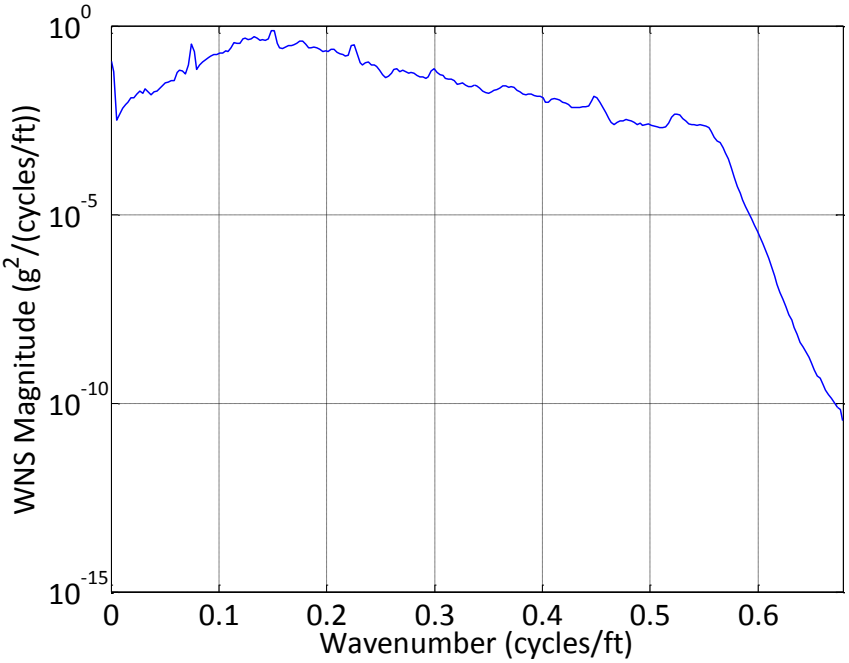

Figure 59: WNS of 1000 seconds of a warm up test with an MTVR at a constant speed of about 74 ft/sec

As can be seen, the shape of the two plots are identical, but the magnitudes are different. This can also be done for medium wavelength data, the pitch and roll vector magnitude, as well as the long wavelength data, the grade.

Appendix B -Terrain Variables Sorted Using K-means Algorithm

Because the k-means algorithm was used to group the vehicle operational data of engine speed, accelerator position and vehicle speed, there was interest in seeing how the k-means algorithm would sort the variables used in the TRIC algorithm. The three variables, z-axis acceleration 20 second RMS values, pitch and roll vector magnitude 20 second averages, and terrain grade over 20 second periods, were placed in a matrix and run through the k-means algorithm, with *k*, the number of groups, being equal to 28. The resulting k-means group assignments were made into a three-dimensional histogram along with the TRIC classifications as was done in the last section.

B.1 Tests and Analysis

The resulting contour histogram for the paved oval track used in the last section can be seen in Figure 60. The figure shows some strong correlation between the idle modes (TRIC group 1 and k-means groups 22 and 23) which is to be expected. Figure 61 shows histograms for the k-means groups for each specific TRIC group.

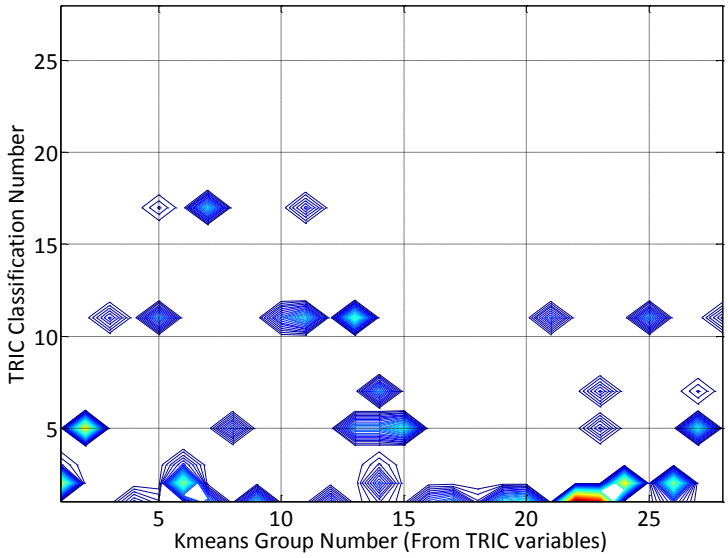

Figure 60: Histogram of terrain classification number and the k-means group number from the TRIC variables, rather than from the vehicle operation variables as was done in the last section

Like before, because only seven TRIC classifications are made with this data, and all 28 k-means groups are filled, each TRIC group corresponds with several k-means groups. However, there are only a few cases where the k-means groups are correlated to more than one TRIC group. This mostly happens in TRIC group 5, which has terrain features similar to TRIC groups 2, 7, and 11. This test shows that k-means algorithm can also be used to classify terrain, and can even break the TRIC classifications into smaller sub-groups, depending on the value of *k*.

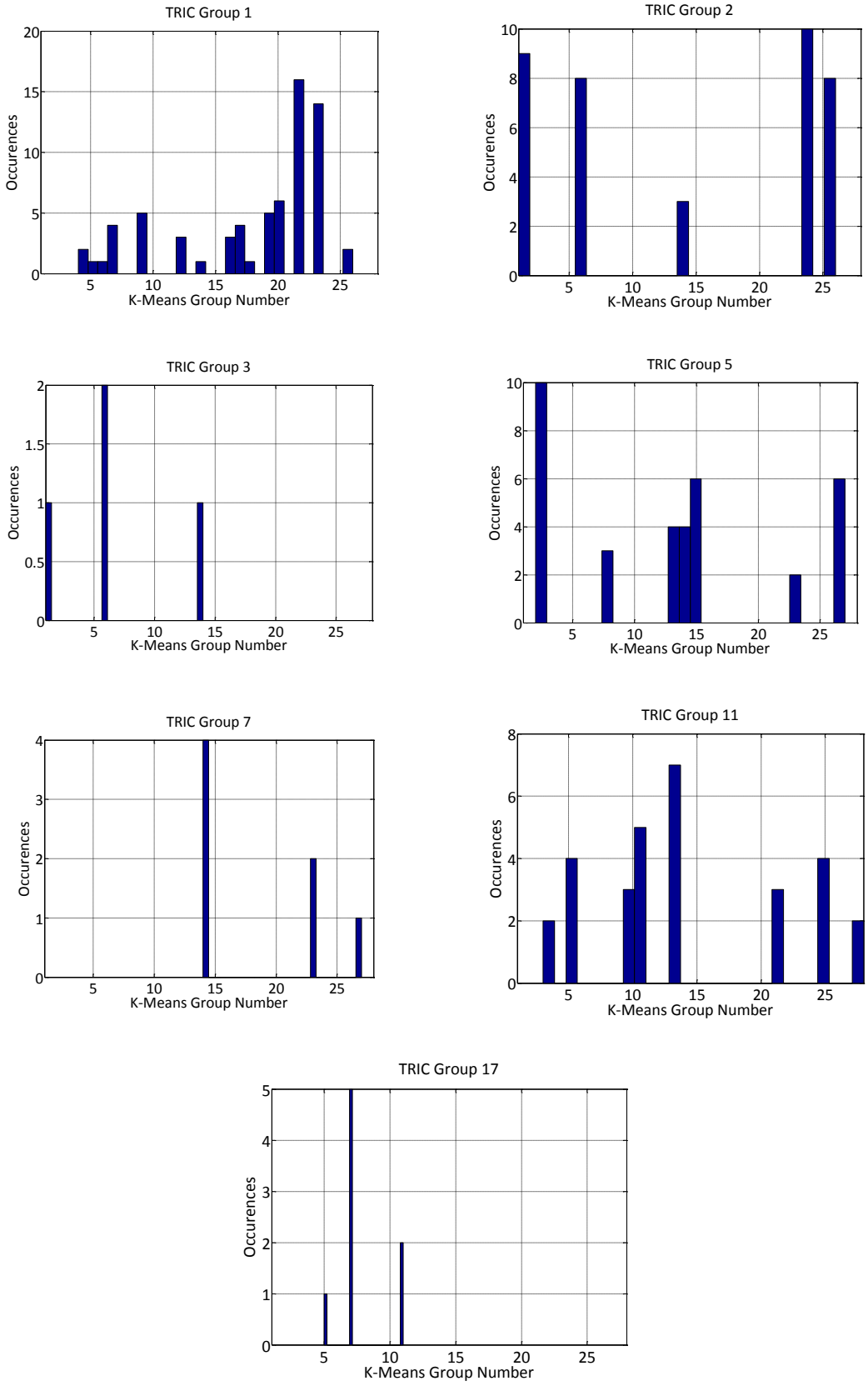

Figure 61: K-means group histograms for each of the TRIC classifications in the MTVR test

The comparison of TRIC classifications and k-means classifications from the variables used in the TRIC algorithm was also done on the data from the HEMTT Churchville test. Figure 62 shows the two-dimensional contour histogram of the classifications.

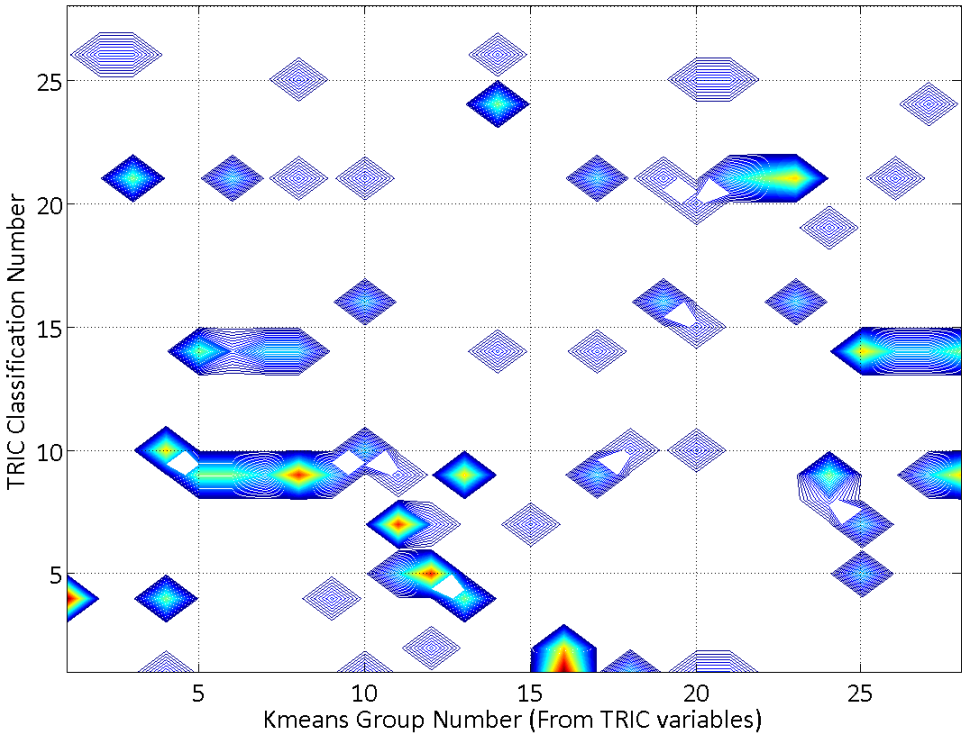

Figure 62: Contour histogram for the Churchville HEMTT test

Like the last test, several of the TRIC classification have matches with more than one of the k-means groups. This is because there are more k-means groups than TRIC classifications used, even though there are more TRIC classifications used here than in the previous test. There is some overlap of the k-means groups between the different TRIC classifications, but the strongest correlations show little to no overlap. The k-means group histograms for specific TRIC classification groups for the strongest correlations are shown

in Figure 63. The histograms show that TRIC classification groups 9 and 14 have the most overlap between matches with k-means groups. TRIC groups 9 and 14 have very similar terrain. Group 9 has low amplitude short wavelength, mid amplitude medium wavelength and mid amplitude long wavelength. Group 14 is the same except it has mid amplitude for the short wavelength as well. Like the previous test, this test shows that k-means algorithm can also be used to classify terrain, and can even break the TRIC classifications into smaller sub-groups, depending on the value of *k*. However, similar terrains that are classified in a different TRIC group may be classified in the same k-means group.

B.2 Comparison Conclusions

The conclusion is that the k-means grouping algorithm is a good way to classify the harshness of terrain, and can classify the terrain in tighter, smaller groups than the TRIC algorithm if that is what is desired. It can be thought of as an approximation to the TRIC algorithm, but should not be considered as a replacement, especially if classifications with definite thresholds are desired. It will group according the user inputted number of groups, and does not place the data into pre-determined classifications, as the TRIC algorithm does (see Chapter 5 for a discussion on future work to resolve this issue).

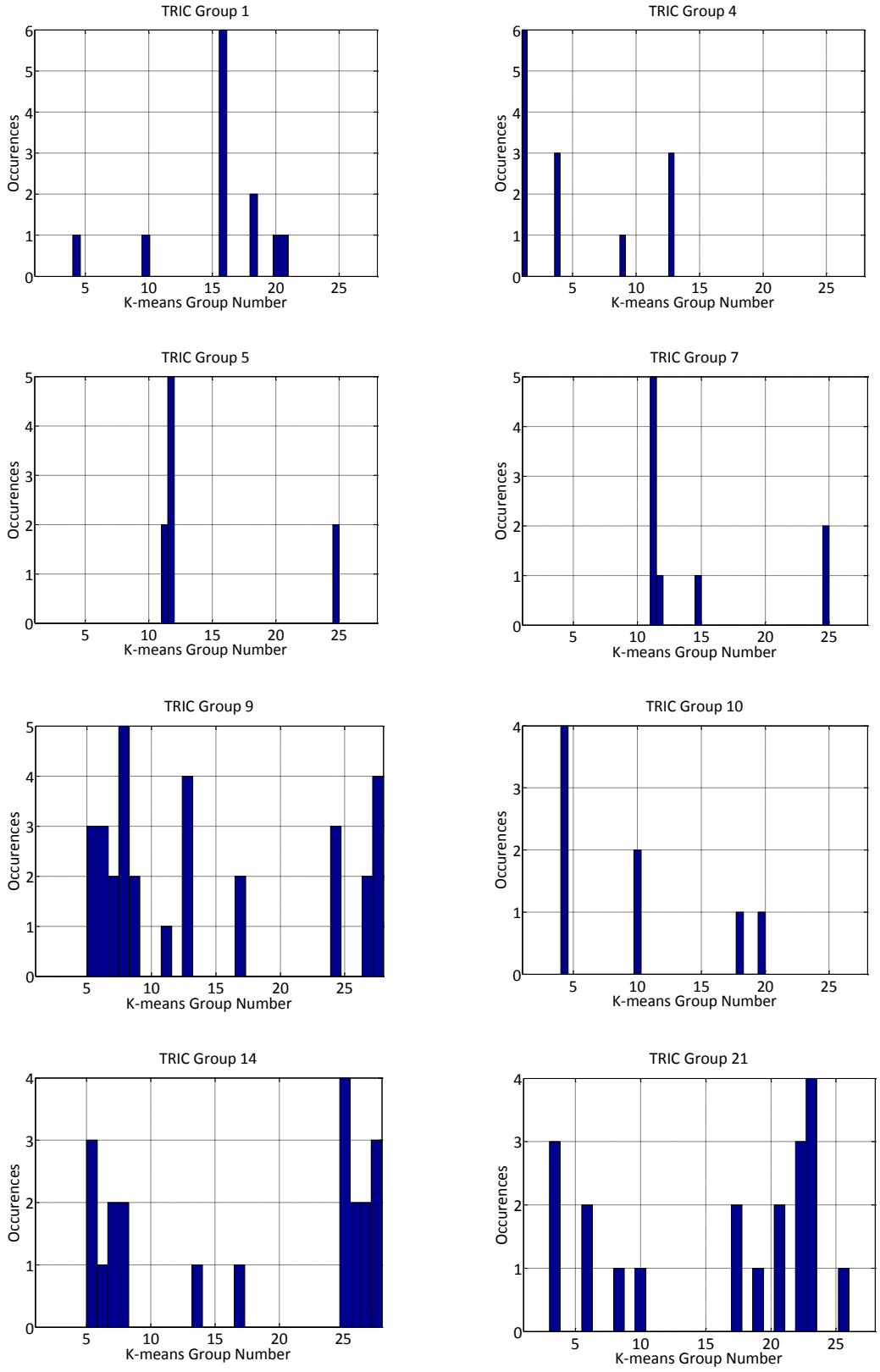

Figure 63: Histograms of the k-means groups for selected TRIC groups of the Churchville test

Appendix C - MATLAB Code

C.1 TRIC Algorithm Function

```
function
[AVG_Speed,PctGrade,AVG_PR_Mag,SD_unsprung,SD_unld_unsprung,classTRIC,c
lassTRICd,mwl_hiamp_nonzero,swl,mwl,lwl,T] =
TRIC(acc_filename,RS232_filename,unloaded_acc_file,ParamDataStructure)

%
**********************************************************************
**
%
**********************************************************************
**
%  Before running this function the ParseDataBusFile.m function must be
ran
%  to create a .mat file of the dearborn data file.
%
**********************************************************************
**
%
**********************************************************************
**
% This part extracts altitude and speed from the .mat file.
% Altitude is converted from meters to feet
% A 20 second change in altitude is computed.
% Speed is converted from km/h to mph.
% Speed is integrated to get a distance array.
% 20 second block averages of speed are computed
% The 20 second difference of each point of the distance array is taken
so
% that it is really distance traveled every 20 seconds
% The 20 second change in altitude is divided by the 20 second change
in
% distance to give the slope of the terrain
%
**********************************************************************
**

%Threshold values


xh = .031; %Upper threshold slope for acc vs. speed
bh = -.1; %Upper threshold intercept for acc vs. speed
xl = .0055; %Lower threshold slope for acc vs. speed
bl = .03; %Lower threshold intercept for acc vs. speed
```

```
mst = 5.825; %min_speed_threshhold
APRMthrH = .8077; %AVG_PR_Mag_threshold high
APRMthrL = .3570; %AVG_PR_Mag_threshold low
% APRMthrH = .048; %New AVG_PR_Mag_threshold high
% APRMthrL = .02; %New AVG_PR_Mag_threshold low
PGthrH = .08; %PctGrade_threshold high
PGthrL = .02; %PctGrade_threshold low

%
**********************************************************************
*
% Altitude and Speed from GPS. 20 second Averages taken. Speed
integrated
% to get distance.
%
**********************************************************************
*

Alt_m =
ParamDataStructure.J1939.SRC_217.PGN_65256.SPNList.SPN_580.Data;
Alt = Alt_m*3.2808399;
Spd_kmh =
ParamDataStructure.J1939.SRC_217.PGN_65256.SPNList.SPN_517.Data;
Spd = Spd_kmh*.621371192;

delta_Alt = zeros([floor(length(Alt)/200)+1,1]);
for bb = 1:floor(length(Alt)/200)
    delta_Alt(bb) = Alt(200*bb)-Alt((bb-1)*200+1);
end
delta_Alt(bb+1) = Alt(length(Alt))-Alt(floor(length(Alt)/200)*200);

AVG_Speed = zeros([floor(length(Spd)/200)+1,1]);
for aa = 1:floor(length(Spd)/200)
    AVG_Speed(aa) = mean(Spd((aa-1)*200+1:200*aa));
end
AVG_Speed(aa+1) = mean(Spd(floor(length(Spd)/200)*200:length(Spd)));

Dist = cumtrapz(Spd);

delta_Dist = zeros([floor(length(Dist)/200)+1,1]);
for cc = 1:floor(length(Dist)/200)
    delta_Dist(cc) = Dist(200*cc)-Dist((cc-1)*200+1);
end
delta_Dist(cc+1) = Dist(length(Dist))-
Dist(floor(length(Dist)/200)*200);

PctGrade = delta_Alt./delta_Dist;
PctGrade(1) = 0;



%
**********************************************************************
**
```

```
% This part decimates unsprung acceleromter data sampled at 10000 Hz
down
% to 100 Hz according to the AMSAA TRIC algorithm. It then takes 20
second
% RMS values with the last point being an RMS of whatever the remainder
is.
%
**********************************************************************
**

fid = textread(acc_filename);
full_acc = fid(:,2);
fs_acc = 10000;
N_acc = length(full_acc);
dt_acc = 1/fs_acc;
T = N_acc*dt_acc;
dec1_acc = decimate(full_acc,10);
dec2_acc = decimate(dec1_acc,10);

% Low Pass Filter (30 Hz)
b = fir1(8,.6);
acc = filter(b,1,dec2_acc);

n = 2000;
SD_unsprung = zeros([ceil(length(acc)/n),1]);
for xx = 1:floor(length(acc)/n)
    SD_unsprung(xx) = rms(acc((xx-1)*2000+1:2000*xx));
end
SD_unsprung(ceil(length(acc)/n)) =
rms(acc(floor(length(acc)/n)+1:length(acc)));

%
**********************************************************************
**
% This part decimates sprung acceleromter data sampled at 10000 Hz down
% to 100 Hz according to the AMSAA TRIC algorithm. It then takes 20
second
% RMS values with the last point being an RMS of whatever the remainder
is.
%
**********************************************************************
**

fid3 = textread(unloaded_acc_file);
full_ul_acc = fid3(:,2);
dec1_ul_acc = decimate(full_ul_acc,10);
dec2_ul_acc = decimate(dec1_ul_acc,10);

% Low Pass Filter (30 Hz)
b = fir1(8,.6);
acc_ul = filter(b,1,dec2_ul_acc);

n = 2000;
SD_unld_unsprung = zeros([ceil(length(acc_ul)/n),1]);
for tt = 1:floor(length(acc_ul)/n)
```

```
    SD_unld_unsprung(tt) = rms(acc_ul((tt-1)*2000+1:2000*tt));
end
SD_unld_unsprung(ceil(length(acc_ul)/n)) =
rms(acc_ul(floor(length(acc_ul)/n)+1:length(acc_ul)));

%*********************************************************************
****
% This part parses the pitch and roll magnitudes
% It then takes the difference between each point and takes the pitch
and
% roll magnitude (rate of angular change) and then takes the 20 second
% block average
%*********************************************************************
****

fid2 = fopen(RS232_filename);
PR = textscan(fid2,'%.5f64 %*s %f,%f,%f','delimiter','=');

Pitch = PR{3};
Roll = PR{4};

delta_pitch = zeros([length(Pitch)-1,1]);
for  pp = 2:length(Pitch)
    delta_pitch(pp-1) = Pitch(pp)-Pitch(pp-1);
end

delta_roll = zeros([length(Roll)-1,1]);
for rr = 2:length(Roll)
    delta_roll(rr-1) = Roll(rr)-Roll(rr-1);
end

PR_Mag = sqrt(delta_pitch.^2+delta_roll.^2);

N = 985;

AVG_PR_Mag = zeros([ceil(length(PR_Mag)/N),1]);
AVG_PR_Mag(1) = mean(PR_Mag(1:N));

for nn = 2:floor(length(PR_Mag)/N)
    AVG_PR_Mag(nn) = mean(PR_Mag((nn-1)*N+1:nn*N));
end
AVG_PR_Mag(ceil(length(PR_Mag)/N)) =
mean(PR_Mag(nn*N+1:length(PR_Mag)));

fclose(fid2);

% Short wavelength classifications

swl_loamp = zeros(length(SD_unsprung),1);
for rr = 1:length(SD_unsprung)
    if (SD_unsprung(rr)<=(xl*AVG_Speed(rr)+bl))
        swl_loamp(rr) = SD_unsprung(rr);
        else swl_loamp(rr) = 0;
    end
```

```
end
swl_loamp_nonzero = swl_loamp~=0;

swl_midamp = zeros([length(SD_unsprung),1]);
for ss = 1:length(SD_unsprung)
    if (SD_unsprung(ss)<=(xh*AVG_Speed(ss)+bh) &&
SD_unsprung(ss)>(xl*AVG_Speed(ss)+bl))
    swl_midamp(ss) = SD_unsprung(ss);
    else swl_midamp(ss) = 0;
    end
end
swl_midamp_nonzero = swl_midamp~=0;

swl_hiamp = zeros([length(SD_unsprung),1]);
for tt = 1:length(SD_unsprung)
    if (SD_unsprung(tt)>(xh*AVG_Speed(tt)+bh))
    swl_hiamp(tt) = SD_unsprung(tt);
    else swl_hiamp(tt) = 0;
    end
end
swl_hiamp_nonzero = swl_hiamp~=0;

swl = 1:length(AVG_Speed);

% Plot
figure('Units', 'pixels', 'Position', [1380 550 500 375]);
slow =
scatter(AVG_Speed(swl_loamp_nonzero),swl_loamp(swl_loamp_nonzero),80);
set(slow,'MarkerFaceColor',[0 .75 0],'MarkerEdgeColor',[0 .5
0],'Marker','o')
hold on
smid =
scatter(AVG_Speed(swl_midamp_nonzero),swl_midamp(swl_midamp_nonzero),80
);
set(smid,'MarkerFaceColor',[1 .5 0],'MarkerEdgeColor',[.75 .5
0],'Marker','o')
shi =
scatter(AVG_Speed(swl_hiamp_nonzero),swl_hiamp(swl_hiamp_nonzero),80);
set(shi,'MarkerFaceColor',[1 0 0],'MarkerEdgeColor',[.75 0
0],'Marker','o')
line1 = xl*AVG_Speed+bl; %Lower threshold line
line2 = xh*AVG_Speed+bh; %Upper threshold line
sthreshold_l = line(AVG_Speed, line1);
sthreshold_h = line(AVG_Speed, line2);
set(sthreshold_l,'Color',[.5 .5 .5],'LineWidth',.25)
set(sthreshold_h,'Color',[.5 .5 .5],'LineWidth',.25)
x=[mst,mst]; %For the minimum speed threshold line
y=[-.05,.5];
spd = plot(x,y,':k');
%axis([0 50 .08 .3])
sleg = legend([slow, smid, shi],'Low Amplitude','Mid Amplitude','High
Amplitude', 'location','SouthEast');
set(gca, 'FontName', 'Georgia', 'FontSize', 26)
% sti = title('Short Wave Length Data');
xlabel('Average Speed (mph)');
ylabel('Acceleration (g''s)');
```

```
% set(sti, 'FontSize', 16, 'FontWeight', 'bold')
set(sleg,'FontName', 'Georgia', 'FontSize',20)
str1 = {'y = (.0325)x-.105'};
text(10,.2,str1,'FontName', 'Georgia')
str2 = {'y = (.01)x+.026'};
text(20,.1,str2,'FontName', 'Georgia')
hold off

%Classifications
for kk = 1:length(AVG_Speed)
        if (AVG_Speed(kk)<mst)
            swl(kk) = 0;
        else
            if (SD_unsprung(kk)<=line1(kk))
                swl(kk) = 1;
            elseif (SD_unsprung(kk)>line1(kk)) &&
(SD_unsprung(kk)<=line2(kk))
                swl(kk) = 2;
            else
                swl(kk) = 3;
            end
        end
end

% Medium Wavelength Classifications

mwl_loamp = zeros([length(AVG_PR_Mag),1]);
for uu = 1:length(AVG_PR_Mag)
    if (AVG_PR_Mag(uu)<=APRMthrL)
    mwl_loamp(uu) = AVG_PR_Mag(uu);
    else mwl_loamp(uu) = 0;
    end
end
mwl_loamp_nonzero = mwl_loamp~=0;

mwl_midamp = zeros([length(AVG_PR_Mag),1]);
for vv = 1:length(AVG_PR_Mag)
    if (AVG_PR_Mag(vv)<=APRMthrH && AVG_PR_Mag(vv)>APRMthrL)
    mwl_midamp(vv) = AVG_PR_Mag(vv);
    else mwl_midamp(vv) = 0;
    end
end
mwl_midamp_nonzero = mwl_midamp~=0;

mwl_hiamp = zeros([length(AVG_PR_Mag),1]);
for ww = 1:length(AVG_PR_Mag)
    if (AVG_PR_Mag(ww)>APRMthrH)
    mwl_hiamp(ww) = AVG_PR_Mag(ww);
    else mwl_hiamp(ww) = 0;
    end
end
mwl_hiamp_nonzero = mwl_hiamp~=0;

mwl = 1:length(AVG_Speed);
```

```
% Plot
figure('Units', 'pixels', 'Position', [1980 550 500 375]);
mlow =
scatter(AVG_Speed(mwl_loamp_nonzero),mwl_loamp(mwl_loamp_nonzero),80);
set(mlow,'MarkerFaceColor',[0 .75 0],'MarkerEdgeColor',[0 .5
0],'Marker','^')
hold on
mmid =
scatter(AVG_Speed(mwl_midamp_nonzero),mwl_midamp(mwl_midamp_nonzero),80
);
set(mmid,'MarkerFaceColor',[1 .5 0],'MarkerEdgeColor',[.75 .5
0],'Marker','^')
mhi =
scatter(AVG_Speed(mwl_hiamp_nonzero),mwl_hiamp(mwl_hiamp_nonzero),80);
set(mhi,'MarkerFaceColor',[1 0 0],'MarkerEdgeColor',[.75 0
0],'Marker','^')
set(gca, 'FontName', 'Georgia', 'FontSize', 26)
xx1 = [0, 50];
yy1 = [APRMthrL, APRMthrL]; %Lower threshold line
yy2 = [APRMthrH, APRMthrH]; %Upper threshold line
mthreshold_l = plot(xx1, yy1);
mthreshold_h = plot(xx1, yy2);
set(mthreshold_l,'Color',[.5 .5 .5],'LineWidth',.25)
set(mthreshold_h,'Color',[.5 .5 .5],'LineWidth',.25)
y2 = [0,.11];
plot(x,y2,':k')
%axis([0 50 .02 .055])
mleg = legend([mlow, mmid, mhi],'Low Amplitude','Mid Amplitude','High
Amplitude', 'location','SouthEast');
% mti = title('Medium Wave Length Data');
xlabel('Average Speed (mph)');
ylabel('Pitch/Roll Vector Magnitude');
% set(mti, 'FontSize', 16, 'FontWeight', 'bold');
set(mleg,'FontName', 'Georgia', 'FontSize',20);
str3 = {'y = .048'};
text(10,.1,str3,'FontName', 'Georgia')
str4 = {'y = .02'};
text(20,.1,str4,'FontName', 'Georgia')
hold off

% Classifications
for kk = 1:length(AVG_Speed)
        if (AVG_Speed(kk)<mst)
            mwl(kk) = 0;
        else
            if (AVG_PR_Mag(kk)<=APRMthrL)
                mwl(kk) = 10;
            elseif (AVG_PR_Mag(kk)>APRMthrL) &&
(AVG_PR_Mag(kk)<=APRMthrH)
                mwl(kk) = 20;
            else
                mwl(kk) = 30;
            end
        end
end
```

```
%Long Wavelength Classifications

lwl_loamp = zeros([length(PctGrade),1]);
for xx = 1:length(PctGrade)
    if (PctGrade(xx)<=PGthrL)
    lwl_loamp(xx) = PctGrade(xx);
    else lwl_loamp(xx) = 0;
    end
end
lwl_loamp_nonzero = lwl_loamp~=0;

lwl_midamp = zeros([length(PctGrade),1]);
for yy = 1:length(PctGrade)
    if (PctGrade(yy)<=PGthrH && PctGrade(yy)>PGthrL)
    lwl_midamp(yy) = PctGrade(yy);
    else lwl_midamp(yy) = 0;
    end
end
lwl_midamp_nonzero = lwl_midamp~=0;

lwl_hiamp = zeros([length(PctGrade),1]);
for zz = 1:length(PctGrade)
    if (PctGrade(zz)>PGthrH)
    lwl_hiamp(zz) = PctGrade(zz);
    else lwl_hiamp(zz) = 0;
    end
end
lwl_hiamp_nonzero = lwl_hiamp~=0;

% Plot
figure('Units', 'pixels', 'Position', [1980 50 500 375]);
llow =
scatter(AVG_Speed(lwl_loamp_nonzero),lwl_loamp(lwl_loamp_nonzero),80);
set(llow,'MarkerFaceColor',[0 .75 0],'MarkerEdgeColor',[0 .5
0],'Marker','s')
hold on
lmid =
scatter(AVG_Speed(lwl_midamp_nonzero),lwl_midamp(lwl_midamp_nonzero),80
);
set(lmid,'MarkerFaceColor',[1 .5 0],'MarkerEdgeColor',[.75 .5
0],'Marker','s')
lhi =
scatter(AVG_Speed(lwl_hiamp_nonzero),lwl_hiamp(lwl_hiamp_nonzero),80);
set(lhi,'MarkerFaceColor',[1 0 0],'MarkerEdgeColor',[.75 0
0],'Marker','s')
xx1 = [0, 50];
yyy1 = [PGthrL, PGthrL]; %Lower threshold line
yyy2 = [PGthrH, PGthrH]; %Upper threshold line
lthreshold_l = plot(xx1, yyy1,'b');
lthreshold_h = plot(xx1, yyy2,'g');
set(lthreshold_l,'Color',[.5 .5 .5],'LineWidth',.25)
set(lthreshold_h,'Color',[.5 .5 .5],'LineWidth',.25)
y2 = [-.3,.5];
plot(x,y2,':k')
%axis([0 50 -.15 .1])
set(gca, 'FontName', 'Georgia', 'FontSize', 26)
```

```
lleg = legend([llow, lmid, lhi],'Low Amplitude','Mid Amplitude','High
Amplitude', 'location','NorthEast');
% lti = title('Signed Long Wave Length Data');
xlabel('Average Speed (mph)');
ylabel('Slope');
% set(lti,'FontName','Georgia','FontSize', 16, 'FontWeight', 'bold')
set(lleg,'FontName', 'Georgia', 'FontSize',20)
str7 = {'y = .08'};
text(10,.1,str7,'FontName', 'Georgia')
str8 = {'y = .02'};
text(20,.1,str8,'FontName', 'Georgia')
hold off

lwl = 1:length(AVG_Speed);

% Classifications
for kk = 1:length(AVG_Speed)
        if (AVG_Speed(kk)<mst)
            lwl(kk) = 0;
        else
            if (PctGrade(kk)<=PGthrL)
                lwl(kk) = 100;
            elseif (PctGrade(kk)>PGthrL) && (PctGrade(kk)<=PGthrH)
                lwl(kk) = 200;
            else
                lwl(kk) = 300;
            end
        end
end

% Overall TRIC Classifications for 27 harshness levels or 'degenerate'
% classifications for just 7 levels of harshness

classTRIC = lwl+mwl+swl;
classTRICd = lwl+mwl+swl;

clear i;
for i = 1:length(AVG_Speed)
    if classTRICd(i) == 111
        classTRICd(i) = 1;
    elseif classTRICd(i) == 112 || classTRICd(i) == 211 ||
classTRICd(i) == 121
        classTRICd(i) = 2;
    elseif classTRICd(i) == 113 || classTRICd(i) == 122 ||
classTRICd(i) == 212 || classTRICd(i) == 221 || classTRICd(i) == 311 ||
classTRICd(i) == 131
        classTRICd(i) = 3;
    elseif classTRICd(i) == 123 || classTRICd(i) == 213 ||
classTRICd(i) == 222 || classTRICd(i) == 312 || classTRICd(i) == 321 ||
classTRICd(i) == 132 || classTRICd(i) == 231
        classTRICd(i) = 4;
    elseif classTRICd(i) == 223|| classTRICd(i) == 313 || classTRICd(i)
== 322 || classTRICd(i) == 331 || classTRICd(i) == 133 || classTRICd(i)
== 232
        classTRICd(i) = 5;
```

```
    elseif classTRICd(i) == 323 || classTRICd(i) == 332 ||
classTRICd(i) == 233
        classTRICd(i) = 6;
    elseif classTRICd(i) == 333
        classTRICd(i) = 7;
    end
end

clear i;
for i = 1:length(AVG_Speed)
    if classTRIC(i) == 111
        classTRIC(i) = 1;
    elseif classTRIC(i) == 112
        classTRIC(i) = 2;
    elseif classTRIC(i) == 211
        classTRIC(i) = 3;
    elseif classTRIC(i) == 121
        classTRIC(i) = 4;
    elseif classTRIC(i) == 113
        classTRIC(i) = 5;
    elseif classTRIC(i) == 122
        classTRIC(i) = 6;
    elseif classTRIC(i) == 212
        classTRIC(i) = 7;
    elseif classTRIC(i) == 221
        classTRIC(i) = 8;
    elseif classTRIC(i) == 311
        classTRIC(i) = 9;
    elseif classTRIC(i) == 131
        classTRIC(i) = 10;
    elseif classTRIC(i) == 123
        classTRIC(i) = 11;
    elseif classTRIC(i) == 213
        classTRIC(i) = 12;
    elseif classTRIC(i) == 222
        classTRIC(i) = 13;
    elseif classTRIC(i) == 312
        classTRIC(i) = 14;
    elseif classTRIC(i) == 321
        classTRIC(i) = 15;
    elseif classTRIC(i) == 132
        classTRIC(i) = 16;
    elseif classTRIC(i) == 231
        classTRIC(i) = 17;
    elseif classTRIC(i) == 223
        classTRIC(i) = 18;
    elseif classTRIC(i) == 313
        classTRIC(i) = 19;
    elseif classTRIC(i) == 322
        classTRIC(i) = 20;
    elseif classTRIC(i) == 331
        classTRIC(i) = 21;
    elseif classTRIC(i) == 133
        classTRIC(i) = 22;
    elseif classTRIC(i) == 232
        classTRIC(i) = 23;
    elseif classTRIC(i) == 323
```

```
        classTRIC(i) = 24;
    elseif classTRIC(i) == 332
        classTRIC(i) = 25;
    elseif classTRIC(i) == 233
        classTRIC(i) = 26;
    elseif classTRIC(i) == 333
        classTRIC(i) = 27;

    end
end
end
```

C.2 K-Means Processing and Map Generation

```
% Uses kmeans to classify vehicle operating modes. 20 second averages
of
% speed, engine speed, and accelerator position are used.

%% Load the Data Structure
clearvars -except T classTRIC classTRICd; close all;
load('Sort loaded paved_20121106_211003CAN-0_dearborn.mat')
%%

close all; clearvars -except ParamDataStructure T classTRIC classTRICd;

% Define the variables from the data structure

engineSpeed =
ParamDataStructure.J1939.SRC_000.PGN_61444.SPNList.SPN_190.Data;
accPos =
ParamDataStructure.J1939.SRC_000.PGN_61443.SPNList.SPN_91.Data;
Spd_kmh =
ParamDataStructure.J1939.SRC_217.PGN_65256.SPNList.SPN_517.Data;
lat = ParamDataStructure.J1939.SRC_217.PGN_65267.SPNList.SPN_584.Data;
long = ParamDataStructure.J1939.SRC_217.PGN_65267.SPNList.SPN_585.Data;
SpdMPH = Spd_kmh*.621371192;

t = 1:20:T;

% Take the 20 second averages of the engine speed
fs1 = length(engineSpeed)/T;
P20 = fs1*20;
engineSpdAvg = zeros([floor(length(engineSpeed)/P20)+1,1]);
for i = 1:floor(length(engineSpeed)/P20)
    engineSpdAvg(i) = mean(engineSpeed((i-1)*P20+1:P20*i));
end
engineSpdAvg(i+1) =
mean(engineSpeed(floor(length(engineSpeed)/P20)*P20:length(engineSpeed)
));

clear i; clear P20;
```

```
% Take the 20 second averages of the accelerator position
fs2 = length(accPos)/T;
P20 = fs2*20;
accPosAvg = zeros([floor(length(accPos)/P20)+1,1]);
for i = 1:floor(length(accPos)/P20)
    accPosAvg(i) = mean(accPos((i-1)*P20+1:P20*i));
end
accPosAvg(i+1) =
mean(accPos(floor(length(accPos)/P20)*P20:length(accPos)));

clear i; clear P20;

% Take the 20 second averages of the vehicle speed
fs3 = length(SpdMPH)/T;
P20 = fs3*20;
speedAvg = zeros([floor(length(SpdMPH)/P20)+1,1]);
for i = 1:floor(length(SpdMPH)/P20)
    speedAvg(i) = mean(SpdMPH((i-1)*P20+1:P20*i));
end
speedAvg(i+1) =
mean(SpdMPH(floor(length(SpdMPH)/P20)*P20:length(SpdMPH)));

clear i; clear P20;

% Take the 20 second averages of the latitude
fs4 = length(lat)/T;
P20 = fs4*20;
latAvg = zeros([floor(length(lat)/P20)+1,1]);
for i = 1:floor(length(lat)/P20)
    latAvg(i) = mean(lat((i-1)*P20+1:P20*i));
end
latAvg(i+1) = mean(lat(floor(length(lat)/P20)*P20:length(lat)));

% Take the 20 second averages of the longitude
longAvg = zeros([floor(length(long)/P20)+1,1]);
for i = 1:floor(length(long)/P20)
    longAvg(i) = mean(long((i-1)*P20+1:P20*i));
end
longAvg(i+1) = mean(lat(floor(length(long)/P20)*P20:length(long)));

clc;

% Construct the operational mode variables matrix and run the built in
% kmeans algorithm
varMatrix = [engineSpdAvg, accPosAvg, speedAvg];
[IDX, C] = kmeans(varMatrix, 28);
%%
% Define a random color map
map = rand(28,3);
colormap(map);
cmap = colormap;
colorCode = [0,0,0; 0 .6 0;.6 .8 .2;.8 1 .4;1 1 0;1 .7 .2;1 0 0;.7 0
0];

% Scatter plot of the operational modes variables
```

```
clear i
figure(1)
hold on
for i = 1:28

scatter3(varMatrix(IDX==i,1),varMatrix(IDX==i,2),varMatrix(IDX==i,3),20
,cmap(i,:),'fill');
end
grid on
title('Operational Modes of Vehicle')
xlabel('Average Engine Speed (RPM)')
ylabel('Average Accelerator Position (%)')
zlabel('Average Vehicle Speed (mph)')
hold off

% Scatter plot of the kmeans classification against speed
figure(2)
clear i
hold on
for i = 1:28
    scatter(speedAvg(IDX==i),IDX(IDX==i),25,cmap(i,:),'fill')
    grid on
end
title('Average Speed vs. Operational Mode')
xlabel('Average Speed (mph)')
ylabel('Operational Mode Number')
hold off

% Plot of the vehicle path
figure(3)
scatter(long,lat,12,'k','fill')
title('Test Track')
xlabel('Longitude (degrees)')
ylabel('Latitude (degrees)')
grid on
%%
% Plot of the color coded operational modes on the map
clear i;
figure(4)
hold on
for i = 1:28
    scatter(longAvg(IDX==i),latAvg(IDX==i),50,cmap(i,:),'fill')
end
set(gca, 'FontName', 'Calibri', 'FontSize', 22)
title('Test Track')
xlabel('20 Second Average Longitude (degrees)')
ylabel('20 Second Average Latitude (degrees)')
axis([-119.374 -119.354 39.296 39.306])
plot_google_map('maptype','satellite')


for j = 0:7

scatter(longAvg(classTRICd==j),latAvg(classTRICd==j)+.0003,50,colorCode
(j+1,:),'fill','square')
end
```

```
%%
% Scatter plot of the operational modes against time
figure(5)
hold on
for i = 1:28
    scatter(t(IDX==i),IDX(IDX==i),30,cmap(i,:),'fill')
end
title('Operational Modes Over Time')
xlabel('time (s)')
ylabel('Operation Severity Level')
%axis([0 3800 0 8])

grid on
```

Appendix D - Full K-means Group and Variable Tables

D.1 Sort Test with MTVR on Oval Paved Track

Table 10: : Twenty second averages of operational mode variables from a sort test on a paved, oval road driven by an MTVR and sorted into groups by the k-means algorithm

Engine Speed (RPM)	Acc Position (% Max)	Vehicle Speed (mph)	Vehicle Operational Mode
800	10	1	1
778	3	4	1
811	11	1	1
808	3	7	1
774	3	5	1
793	11	1	1
1269	31	14	2
1273	18	17	2
1281	19	18	2
1177	41	9	3
1159	37	7	3
1174	21	12	3
1181	37	19	3
1177	42	9	3
865	3	10	4
873	3	8	4
877	8	11	4
875	3	10	4
852	3	7	4
850	3	8	4
845	3	7	4
868	3	8	4
836	13	1	4
881	3	10	4
1222	21	17	5
1209	26	24	5
1207	19	23	5

700	3	0	6
700	3	0	6
700	3	0	6
700	3	0	6
700	3	0	6
700	3	0	6
700	3	0	6
700	3	0	6
699	3	0	6
700	3	0	6
700	3	0	6
700	3	0	6
700	3	0	6
699	3	0	6
700	3	0	6
700	3	0	6
700	3	0	6
700	3	0	6
701	3	0	6
699	3	3	6
704	4	0	6
700	3	0	6
699	3	0	6
700	3	0	6
700	3	0	6
698	3	0	6
700	3	0	6
700	3	0	6
700	3	0	6
701	3	2	6
700	3	0	6
700	3	0	6
700	3	0	6
927	3	11	7
956	3	16	7
945	21	2	7
942	20	2	7
950	9	13	7
1106	33	6	8
1094	34	6	8
1099	31	5	8

1090	29	22	8
1066	29	4	8
1091	29	5	8
1216	38	8	9
1223	38	8	9
1206	46	10	9
1219	43	10	9
1203	42	9	9
1229	45	11	9
731	3	3	10
738	8	0	10
741	7	0	10
1284	42	25	11
1298	43	11	11
1291	52	30	11
1297	50	26	11
1283	41	25	11
1279	43	10	11
1279	45	10	11
1284	48	25	11
1354	50	23	12
1371	52	20	12
1350	48	18	12
1348	60	34	12
1376	53	20	12
1354	57	35	12
1367	48	40	12
1370	52	20	12
1357	55	36	12
1132	28	31	13
1136	33	31	13
1158	36	30	13
1139	34	31	13
1147	34	31	13
1133	34	30	13
1159	36	29	13
1116	31	15	14
1113	24	29	14
1117	31	30	14
1127	9	24	14
1120	24	30	14

1129	15	28	14
1113	17	27	14
1113	10	23	14
1117	26	30	14
1121	3	25	14
693	3	1	15
695	3	0	15
697	3	0	15
692	3	1	15
694	3	0	15
696	3	0	15
694	3	1	15
692	3	1	15
695	3	0	15
1253	35	25	16
1251	42	30	16
1250	40	9	16
1256	25	13	16
1252	41	19	16
1343	47	12	17
1328	49	22	17
1323	51	23	17
715	3	3	18
709	3	3	18
714	3	2	18
984	23	3	19
975	22	4	19
990	9	15	19
984	24	3	19
913	12	10	20
898	3	12	20
921	20	2	20
887	17	3	20
898	3	9	20
1344	24	36	21
1374	17	37	21
1202	40	30	22
1201	39	31	22
1273	41	25	23
1273	44	24	23
1269	35	22	23

1264	47	11	23
1271	40	25	23
1270	40	25	23
1265	41	25	23
1277	41	25	23
1272	41	25	23
1026	9	12	24
1045	28	4	24
1033	23	20	24
1032	12	11	24
1048	28	4	24
1011	3	20	24
1031	25	20	24
1431	50	15	25
1448	51	20	25
1457	54	39	25
1459	51	39	25
1425	49	15	25
1428	49	15	25
1471	54	40	25
1438	50	14	25
1446	51	20	25
1432	49	15	25
1453	51	20	25
1432	49	15	25
1446	50	15	25
1480	55	40	25
1388	37	15	26
1410	51	13	26
1397	53	17	26
1418	49	20	26
1408	43	15	26
1391	53	17	26
1380	36	14	26
1402	49	19	26
1398	43	19	26
1304	23	38	27
1088	3	23	28
1095	9	19	28
1085	12	19	28

D.2 Churchville HEMTT Test

Table 11: Twenty second averages of operational mode variables from a test on the Churchville course at the Aberdeen Proving Ground in Maryland with a HEMTT

Engine Speed (RPM)	Accelerator Position (% Maximum)	Vehicle Speed (mph)	Vehicle Operational Mode Group
1000	21	11	1
952	24	12	1
950	21	13	1
1819	5	11	2
1842	8	11	2
1868	8	11	2
1874	26	11	2
1825	15	9	2
1880	23	11	2
1835	22	29	2
1496	59	9	3
1462	75	8	3
1497	74	25	3
1494	66	14	3
1466	72	17	3
1475	74	17	3
1485	76	18	3
1447	74	14	3
1514	76	10	3
1490	85	16	3
1465	64	18	3
1453	67	14	3
1248	51	10	4
1184	9	16	4
1261	2	8	4
1262	14	17	4
1577	28	10	5
1540	30	18	5
1553	19	18	5
1559	29	23	5
1580	20	22	5

1548	73	27	5
1528	69	19	5
1693	99	21	6
1708	100	16	6
1690	90	31	6
1699	88	22	6
2105	94	12	7
818	15	3	8
836	2	12	8
836	10	12	8
846	3	12	8
821	19	3	8
1733	71	10	9
1721	70	20	9
1714	21	11	10
1713	5	10	10
1728	12	10	10
1710	21	26	10
1740	20	30	10
702	0	3	11
734	0	8	11
704	2	13	11
1689	42	10	12
1691	63	10	12
1696	65	10	12
1683	40	16	12
1669	48	27	12
1637	62	10	13
1655	63	10	13
1664	93	30	13
1663	72	16	13
1657	49	20	13
1646	74	26	13
1673	75	21	13
1815	99	8	14
1810	55	10	14
1799	100	8	14
1826	69	16	14
1765	54	11	15
1760	81	10	15
1758	50	11	15

1772	63	11	15
1902	66	11	16
1877	67	11	16
1872	54	11	16
1916	60	11	16
1919	64	11	16
1912	71	11	16
1878	64	11	16
1921	72	11	16
1900	100	7	16
776	6	11	17
771	17	2	17
780	6	4	17
1593	52	13	18
1602	50	10	18
1626	78	18	18
1594	85	13	18
1625	92	17	18
1570	100	25	18
1622	63	26	18
1603	77	21	18
1604	96	16	18
1590	99	25	18
1771	18	11	19
1788	18	11	19
1785	37	11	19
1770	20	29	19
1063	50	2	20
1064	44	2	20
1041	7	16	20
1090	14	16	20
1041	20	13	20
1025	21	8	20
1302	46	12	21
1311	58	11	21
1333	22	7	21
1294	19	17	21
1287	40	15	21
1304	27	13	21
1740	100	5	22
1746	100	6	22

1725	100	12	22
1755	100	6	22
1449	12	9	23
1412	8	18	23
2001	38	12	24
1926	26	12	24
1957	24	12	24
1920	7	11	24
1919	20	11	24
1915	9	11	24
1617	22	10	25
1629	14	10	25
1606	8	9	25
1635	31	9	25
1610	36	16	25
1611	1	21	25
1402	65	14	26
1427	76	13	26
1363	65	3	26
1425	67	12	26
1359	58	7	26
1416	51	9	26
1431	52	7	26
1385	45	12	26
1370	39	10	26
1417	75	12	26
929	28	4	27
912	31	0	27
924	23	2	27
1654	10	10	28
1642	39	26	28
1653	13	25	28
1656	31	25	28
1670	16	25	28
1646	6	25	28