

# Development of Mobile Application for Measuring Vehicle Ride Comfort and Road Surface Quality

Mun Chun Mah

*Department of Civil and Environmental Engineering*

*Imperial College London*

---

## Abstract

With millions of car journeys taking place every single day, improving the comfort of the overall experience and ensuring road safety is an important area of research, both for drivers and car manufacturers. Given the popularity of smartphones, this thesis looks to develop a smartphone application that will detect the presence of potholes in a journey and approximate the overall ride comfort. A prototype of the phone application was developed to test the feasibility of such an application and the data recorded compared against an industry-standard vibration meter. From the data, it was concluded that the smartphone application is capable of measuring accelerations that are comparable to industry standard equipment with a  $R$  of 0.5-0.7. The application also detected the presence of a road defects and recorded its Global Positioning System (GPS) coordinates with a 70% accuracy whenever a large spike in accelerations was observed. An approximation of ride comfort was also proposed to conform with the guidelines outlined in the ISO-2631 by introducing a scaling factor to acceleration values of 1.9 times the mean acceleration. Different phone models were also concluded to cause little difference ( $R$  of 0.5-0.9) on the accelerations recorded by the app. Given the relatively novel nature of this work, several recommendations for future work were also proposed at the end of the thesis.

---

## 1 – Introduction

Ride comfort and road quality can be seen as two sides of the same coin. The roughness or presence of defects in a road will affect both the quality of the ride and vehicle operating costs like fuel consumption and tire wear and tear (Abulizi et al, 2016). If left unchecked, these defects can drastically increase the costs of maintaining the road surface. Potholes for example, cost agencies in England and Wales responsible for maintaining road surfaces approximately £100 million every year (Asphalt Industry Alliance, 2016).

With 1 billion vehicles now on the road and 1.2 million lost lives due to vehicle accidents every year (World Health Organization, 2013), identifying and fixing defects in the road surface to maintain a safe level of road surface quality has never been more important. Doing so, however, is a costly procedure. Vehicles specially fitted with sensors or “System with Two Accelerometers

for Measuring Profile, Enabling Real-time data collection” (STAMPER) units that cost approximately £30,000 are the standard methods of measuring road surface quality. As such, the development of a new, low cost method that can replace traditional methods is paramount to improving road safety.

With the advancement in technology, smartphones have become increasingly ubiquitous. In 2017, a smartphone app capable of measuring ride comfort aboard trains was developed by the University of Birmingham (Azzoug & Kaewunruen, 2017). Given the already acceptable capabilities of the sensors present in a smartphone today to measure ride comfort and the rate of development in mobile technology, future smartphones will have even higher quality sensors installed within and might completely replace the need for specially built devices to measure road comfort. Through the power of crowdsourcing, more detailed information about the state of a road can be collected and government funding reallocated from detection of potholes to fixing potholes.

## ***1.1 – Literature Review***

### **1.1.1 – Ride Quality**

Ride quality is defined as the perception of a road user’s overall driving experience. A low-quality ride is one where passengers feel discomfort exceeding a threshold and have a high risk of experiencing motion sickness. Eriksson & Svensson (2015) describes comfort as a general feeling of well-being while passengers afflicted with motion sickness will experience dizziness, fatigues and nausea.

The concept of ride comfort is an elusive one, the ISO 2631-1 (1997) even acknowledging that the many measures of measuring and classifying ride comfort is meant only as a guide due to the high subjectivity involved. Various parameters must be considered where ride comfort is concerned, including but not limited to temperature of the vehicle, vehicle parameters such as suspension spring stiffness, age and gender of driver, roughness of the road and ergonomics of the vehicle in question. However, Wong (2011) states that in general, irregularities in the road surface, for example potholes, account for the largest source of excitation of vehicles. It can therefore be inferred, that the overall quality or comfort of any road vehicle journey will be largely dependent on the smoothness of the road.

To the best of the author’s knowledge, published work regarding ride quality and comfort of passenger cars are few and far in between. Much of the research gone into ride comfort has been regarding trains and heavy vehicles. This thesis largely consults from the international standard, ISO-2631-1 and is supplemented by several other papers that are described in the following section.

### **Weighted RMS Acceleration and Jerk**

The international standard ISO-2631-1 outlines the standard procedure for measuring the vibration felt by a passenger. In addition, it also outlines the mathematical formulae (RMS acceleration) in which these vibrations can be quantified into a standardized measure of ride comfort by setting acceptable ranges or thresholds on what is defined as a “comfortable” ride. There exist updated or more specific versions of this standard, such as the ISO-2631-4, which focuses on measuring ride

comfort on railway vehicles, but as the focus of this thesis was passenger cars, the other versions were disregarded.

Frequency weighted root mean square (RMS) acceleration is the standardized measure that ISO-2631 prescribes to compare the ride quality of each journey. It has the units of metres per second squared ( $\text{m/s}^2$ ) and is given by Equation 1.

$$a_{\text{RMS}} = \left[ \frac{1}{T} \int_0^T a_w^2(t) dt \right]^{\frac{1}{2}} \quad (1)$$

where  $a_{\text{RMS}}$  is the frequency weighted RMS acceleration,  $a_w(t)$  is the weighted acceleration as a function of time ( $\text{m/s}^2$ ) and T is the duration of the measurement, in seconds (s).

For vibrations in more than one axis, the total value of weighted RMS acceleration can be calculated by Equation 2.

$$a_{\text{Total}} = (k_x^2 a_{\text{RMS},X}^2 + k_y^2 a_{\text{RMS},Y}^2 + k_z^2 a_{\text{RMS},Z}^2)^{\frac{1}{2}} \quad (2)$$

Where  $a_{\text{RMS},X}$ ,  $a_{\text{RMS},Y}$ ,  $a_{\text{RMS},Z}$  are the weighted RMS accelerations with respect to the orthogonal axis x, y, z and  $k_x$ ,  $k_y$  and  $k_z$  are suggested multiplying constants detailed in Table 1 below.

Table 1 – Suggested values for multiplying constant, k for each orthogonal axis.

Constant	Value
$k_x$	1.4
$k_y$	1.4
$k_z$	1.0

The frequency weighted RMS acceleration is then compared to a range of values to determine the overall quality of the journey. This range of values is given below in Table 2.

Table 2 – Defining ranges for quality of a ride using weighted RMS acceleration (ISO-2631-1, 1997)

Range of Weighted RMS Acceleration Values	Description
$< 0.315 \text{ m/s}^2$	Not uncomfortable
$0.315 \text{ m/s}^2 - 0.63 \text{ m/s}^2$	A little uncomfortable
$0.5 \text{ m/s}^2 - 1 \text{ m/s}^2$	Fairly uncomfortable
$0.8 \text{ m/s}^2 - 1.6 \text{ m/s}^2$	Uncomfortable
$1.25 \text{ m/s}^2 - 2.5 \text{ m/s}^2$	Very uncomfortable
$> 2.5 \text{ m/s}^2$	Extremely uncomfortable

In addition to calculating the frequency weighted RMS values, acceleration can also be used to calculate jerk, which is also an important measure of road safety and comfort. According to Schot (1978), jerk is defined as the third derivative of distance, or the change of lateral acceleration with respect to time. As with the case of weighted RMS acceleration, the value of lateral jerk should be

between a specific range to ensure a comfortable ride. These ranges vary depending on the type of transport being used, but for passenger cars, Kilinc & Baybura (2012) suggest a jerk range of 0.3 to 0.9 m/s<sup>3</sup> while maintaining a maximum lateral acceleration value of 1.4 m/s<sup>2</sup>.

It is worth noting however that most published works on maximum acceleration and jerk values are regarding rail and public transport. Seeing as how both public and rail transports can have both seated and standing passengers, this thesis assumes that the upper limits of the thresholds should be used as seated passengers resist changes in movement and are much more comfortable seated than while standing up.

As proof to justify this assumption, a study conducted by Eriksson & Svensson (2015) suggests that these maximum tolerable values of jerk (0.9 m/s<sup>3</sup>) and lateral acceleration (1.4 m/s<sup>2</sup>) are too low. They surmise that these values would be acceptable for a passenger car that was keeping in lane but is an unrealistic expectation as changing lanes occurs often in real-life scenarios. Based on their study, they found that passengers classify a journey that had a maximum acceleration of 2 m/s<sup>2</sup> and a jerk of 3.5 m/s<sup>3</sup> as “comfortable”.

### **Motion Sickness**

While discomfort is characterized by high frequent motions, motion sickness is due to actions such as braking often or constantly changing lanes, actions that cause low frequent motion. (Eriksson & Svensson, 2015). Golding et al (2001) found that the critical value at which motion sickness is most likely to occur, or the “maximum nauseogenic potential”, is 0.2 Hz. In addition, according to Ekchian et al (2016), motion sickness is most highly correlated to vertical acceleration, though vertical acceleration alone cannot cause motion sickness. Only in combination with roll velocity and accelerations in the other planes will motion sickness occur. (Ekchian et al, 2016).

The manifestation and severity of motion sickness differ largely from person to person. Ekchian et al (2016) explains that, while anyone with a working vestibular system is subject to motion sickness, the range of susceptibility can vary by a factor of 10,000 to 1 in the general public. This large range is due to the fact that certain conditions, such as surrounding temperature or sleep deprivation, or specific activities like reading or using a laptop in a moving vehicle, can induce and increase the severity of motion sickness experienced by a person.

#### **1.1.2 – GPS in Mobile Devices**

The concept of using GPS systems to obtain location data for research in the transport sector has been around since the 1990s. Due to the importance of having a reliable way of locating and monitoring vehicles in real time, Zito et al (1995) studied the accuracy of using GPS to obtain data regarding vehicle speed, location and direction. Murakami & Wagner (1999) worked to determine if the adoption of GPS would improve trip reporting.

However, a large portion of these experiments were conducted using specialized equipment that would not generally be available to the public. For example, Zito et al (1995) conducted the experiment with a GPS receiver mounted atop the vehicle.

Because of advances made in GPS technology and the importance of increasing sampling accuracy, the use of “everyday” GPS systems have been recognized as a powerful and less expensive method to not only target a wider population, but also reduce the requirements on survey respondents (Vij & Shankari, 2015). Smartphone applications are one such example, with various different apps that can track a user’s location. This comes as no surprise, as smartphone penetration has reached 81% of the UK adult population (Deloitte, 2016). Consumers utilize apps like Strava and Google Maps while various government agencies have launched apps that collect trip data, which can then be used to develop travel demand models. (Strauss et al, 2015) The San Francisco Municipal Transportation Agency (SFMTA) for example, have *CycleTrack* while Atlanta has *Cycle Atlanta*.

### 1.1.3 – Road Surface Quality

The International Roughness Index (IRI) is a globally accepted measurement of road roughness experienced by a vehicle, and can be determined through various methods (Sayers et al, 1986). These methods, which are briefly explained by Sayers (1986), can largely be classified as one of four broad “styles”, which are each ranked Class 1 to Class 4 depending on how directly it can be related to the IRI. The Class 1 precision profile method marks the best possible standard for determination of the IRI and is associated with very stringent requirements and high costs both in terms of effort and financially. On the other end of the spectrum, Class 4 methods are defined as any that use subjective rating or uncalibrated measures. These could be in the form of visually determining the roughness of a road or using a person’s opinion of a particular journey over the road. As the description suggests, using any uncalibrated equipment also falls within the Class 4 jurisdiction.

However, by far the most popular and widely used method of determining the IRI is by estimating it from correlation equations, classified as a Class 3 method. Response-type road surface measurement systems (RTRRMS) can be used to determine the initial “uncorrected roughness” which is then scaled accordingly via a calibration equation in order to be compared to the IRI. Its popularity is derived from the fact that it qualifies as a Class 3 method so long as the results are calibrated and correlated to the IRI, regardless of the instruments used.

According to Du et al (2014), a large portion of highway agencies obtain IRI data via the use of specialized and expensive equipment, mounted on road vehicles in such a way that complicates using the equipment for short periods of time. With the advancement in technology, Abulizi et al (2016) proved that a system of 2 accelerometers and laser sensors known as STAMPER can be used as a cheaper replacement, with a price tag of only £30,000. Another low cost replacement suitable for use in developing countries was introduced in the form of MERLIN (Machine for Evaluating Roughness using Low-cost INstrumentation) which could be produced for \$250 (Cundill, 1991).

### 1.1.4 – Potholes

A pothole is defined as any bowl-shaped hole in a road surface that is at least 150mm wide (Miller & Bellinger, 2003), and is well known to be the cause of road accidents and loss of life. According

to Jo & Ryu (2015), there were an estimated of 90,000 to 180,000 potholes in Korea alone in the years 2008 to 2013 which caused 4223 car accidents.

Aside from the health and safety concerns, potholes also have severe financial repercussions. Based on the Asphalt Industry Alliance (2016), £118 million was spent by authorities in England and Wales to fill potholes. Given that planned filling of potholes are 15%-20% cheaper (depending on location) compared to reactive filling of potholes, the ability to detect and inform the authorities of potholes could save authorities up to £20 million per year.

Currently, there are 3 different methods of detecting potholes, via vibration, laser scanning and vision based methods. Laser scanning is a method that uses Automated Image Collection Systems (AICS) to scan and collect pavement images which are then analyzed to find the location of the pothole (Yu, 2011). While expensive and computationally expensive, the data obtained is extremely detailed (Jo & Ryu, 2015). Vibration based methods on the other hand, use accelerometers fitted onto a vehicle to detect the road surface conditions. An example would be Jaguar Land Rovers' MagneRide sensing technology (Jaguar Land Rover, 2017), which would detect and calculate the severity of potholes. However, this method has the disadvantage of only being able to detect potholes that have been driven over.

## **1.2 – Rationale and Aim**

The functionality of most, if not all road based applications that exist today serve to calculate and display the most efficient route between an origin destination pair. The efficiency is usually measured in terms of time taken to get from one point to another. However, precious few applications calculate efficiency in terms of ride comfort, and those that do exist are largely geared towards cyclists and runners. With 35.7 million cars in UK alone (Department for Transport, 2016) and given the financial savings and health benefits that could be obtained from early detection of potholes, an application that can calculate the most efficient route between an origin destination pair in terms of how comfortable the overall journey is while simultaneously locating potholes would be an excellent investment for both drivers and government agencies alike.

To achieve this, data on how comfortable each road is must be collected, analyzed and quantified. In addition to the methods described in the previous section, other methods include a specially built seat pad that is attached to the seat in a car, which comes with a built-in accelerometer and gyroscope to measure the motions that a regular passenger would normally experience.

However, specially built devices are costly to manufacture, cannot be mass produced and in the case of the seat pad, uncomfortable to use. The United States of America's road network alone measures 6.58 million kilometers and so, a cheaper method of measuring the quality of a road had to be created. This thesis aims to explore the feasibility of solving this problem by taking advantage of the sensors present in a smartphone, a device that is largely ubiquitous in most developed countries, by developing an app that will measure road quality and therefore in turn, ride comfort. This not only confers the benefit of reduced cost as it leverages the accelerometers already present in smartphones, but also opens up the possibility of utilizing the public to collect data over a much wider scope compared to a single fleet of specially built cars.

## **1.3 – Project Objectives**

In order to measure ride comfort, the accelerations experienced by the passengers must first be measured. This thesis looks to come up with proof of concept by building a working prototype to test the feasibility of measuring accelerations and finding potholes with any degree of accuracy. To achieve this, several key objectives were set and are outlined below.

- Creation of a working app that collects and stores the data from the phone accelerometer and internal clock.
- Inclusion of the phone's GPS sensor to supplement the data collection with spatial data.
- Determine if different phone models will have a significant impact on the data collected by the app.
- Determine the app's capability to detect sharp spikes in acceleration, which may signify the presence of potholes, instead of just an average acceleration value to describe ride comfort.
- Side by side testing of the app in a real-world scenario with the HVM200, a specially built seat pad (described in section 2.2) to measure the accuracy of the phone app in detecting accelerations.
- Testing the spatial accuracy of the GPS sensor in real world scenarios to determine viability of a commercialized version of the app that can send live updates to authorities upon detection of a pothole.
- Determine the effects of phone placement in a car during the experiment on the data and propose an optimum location for the phone. (Determine if the app can reliably measure road comfort with a certain level of accuracy despite being placed in areas of a car that aren't outlined in the international standard, ISO-2631-1.)

## **2 – Methodology and Materials**

### *2.1 – Application Development*

#### **2.1.1 – React Native**

With the goal of ultimately releasing the app to the public, it was clear that commerciality would play a large role in the design of the app. It was for this reason that React-Native was chosen as the language in which the app would be written in. React-Native is an upgraded version of React, an open source programming language released by Facebook and Instagram in 2013.

It is without a doubt that Android and iOS are the two main mobile operating systems being used today. However, apps that run on Android systems are largely built using Java as its main language while Objective-C or Swift is used for iOS systems. An app built using Java could not be run on iOS systems and would have to be translated into one of the two iOS accepted languages if the developer intended to target both markets. Instead of building two separate applications, React-Native was chosen, as among its many benefits, React-Native offers the ability to build large sections of the app for both Android and iOS systems without using Java or Objective-C. The flexibility to re-use code would reduce the time required to launch an app in both operating systems to take advantage of the wide customer base.

Despite the large similarities in the iOS and Android operating systems, certain functions and capabilities that are present in one system will be absent in the other. It is worth noting here that React-Native also allows developers the ability to code specific sections of the app in Java or Objective C to offer system specific functions.

### 2.1.2 – App Functionalities

Prior to this thesis, an initial version of the app had already been built by Dr Simon Hu. This prototype used the accelerometer present within the phone to measure the experienced accelerations at a frequency of 10Hz. These results were then displayed on screen and uploaded to a server where it could be downloaded at a later time for analysis. In this project, various upgrades in the form of either new functionality or improvements in the existing code were added to the app and are described below. The latest version of the code for the app can be found in Appendix A.

#### **Location Tracking**

Besides measuring the quality of a journey, accurately determining the location of potholes was also a major objective of this thesis. Therefore, the GPS functionality was included in the app to obtain acceleration data in both space and time. The GPS was programmed to work alongside the existing accelerometer, with a similar frequency of measurement (10Hz) and uploaded its data to the same database as the accelerometer.

#### **Code Efficiency**

Battery life was an important consideration when designing the different functionalities. With the various components (GPS, Accelerometer etc.) being active at the same time, it was important to ensure the power consumption of the app was kept to a minimum to allow continuous running of the app throughout the entire length of a drive.

To achieve this, large portions of the code had to be refactored and certain functionalities replaced. For example, the accelerometer data was previously rendered every 5 seconds and displayed to the phone screen. This was deemed irrelevant and removed in later versions of the app as the data is uploaded to a server and can be analyzed at a later stage. Various portions of the code were also refactored, both to improve readability and also to reduce the number of unnecessary objects being created and destroyed at every cycle.

### 2.1.3- App Prototype

During the development phase of the app, most functionalities could be tested without an actual phone via an emulator, which could simulate various inputs such as GPS coordinates and movement of the phone. Once the app was complete, the next stage of testing then required verifying the output of the app in real world scenarios. In order to do this, a version of the app that could run without the React Native development server was installed onto several phones without going through the Google Play Store.

In addition, due to the app's limited capabilities in uploading data to the server, 3 separate servers had to be created to properly test the app so as to not interfere with the data upload process. Each phone was then installed with a slightly different version of the app that would only upload data to a specific server.

## 2.2 – Instrumentation

### Mobile Smartphones

4 smartphones were used in conducting the experiments throughout this project with 2 being of the same make and model.

- 1) OnePlus2 (Phone 1 and Phone 2)
  - a. RAM: 4GB
  - b. Internal Storage: 64GB
  - c. Android Version: 6.0.1
  - d. Operating System: OxygenOS version 3.5.8
  - e. Network Operator: 4G LTE with Three / 4G LTE with Giffgaff
- 2) Nexus 5X (Phone 3)
  - a. RAM: 2GB
  - b. Internal Storage: 32GB
  - c. Android Version: 7.1.1
  - d. Operating System: Android
  - e. Network Operator: None. (Used a WiFi signal)
- 3) Alcatel Pop 4 (Phone 4)
  - a. RAM: 1GB
  - b. Internal Storage: 8GB
  - c. Android Version: 6.0.1
  - d. Operating System: Android
  - e. Network Operator: 4G LTE with Three

### HVM200

The Human Vibration Meter 200 (HVM200) is a vibration meter that includes frequency weightings used to measure whole body human vibration in line with the ISO-2631 standard (Larson Davis, n.d). It is paired up with the SEN027 seat pad, which the car driver simply sits on

over the course of the experiment. During the experiment, an iPad was used as the mobile interface for configuration of the HVM200 and the resulting data downloaded into a separate computer at the end of the experiment.

For the remainder of this report, the term “seat pad” will be used interchangeably with HVM200 to describe this piece of equipment.

### *2.3 – Experiments*

Three key experiments were carried out over the course of this project, with each experiment repeated several times to obtain an average result. In each experiment except the in-vehicle test, the phones were placed with the screens facing upwards and the “Auto-Rotation” function turned off. Testing of the app revealed the fact that phones with the “Auto-Rotation” function turned on would automatically flip the phones’ X and Y axis whenever the screen rotated, effectively mixing up the accelerations collected in the X and Y axis. In the In-Vehicle test described below, the auto rotation function was not turned off and thus, only accelerations in the Z axis or a total sum of the accelerations could be analyzed.

To ensure consistency in the orientation of all 3 axis, it is recommended that the auto rotation function be turned off in all future experiments using the app.

#### **Experiment 1 – Preliminary Bicycle Testing**

A cycling road with 2 speed bumps was located in Hyde Park, London and using Google Maps, the GPS coordinates of the 2 bumps could also be determined. This experiment involved securing Phone 1 to the handlebars of a Santander bicycle with the use of cellophane tape and cycling over the 2 known bumps. The beginning and end of each experiment was recorded on a notepad to facilitate data analysis at a later stage.

In addition, a separate app, Strava (Strava, 2017) was activated before the experiment began. It runs in the background of the phone and does not interfere with the testing of the ride comfort app. Strava is a popular phone application that, among its many features, offers the ability to track a phone’s location and plot on a map the routes taken.

#### **Experiment 2 – Controlled Laboratory Testing**

2 phones, Phone 1 and 2 were used in this experiment. Both phones had to be installed with a slightly modified version of the app to ensure the data from each phone would be uploaded to a separate database. Both phones were securely attached to each other side by side using cellophane tape with the screen facing upwards.

Both apps were started simultaneously and the phones placed on a flat surface (a thick hardcover book or similar hard flat surface). The surface was then moved up and down at varying speeds to simulate travelling over a road whilst occasionally encountering speed bumps or potholes.

A separate but similar version of this experiment was also conducted, one that used 3 phones instead of 2. For those experiments, the additional phone used was Phone 3.

### **Experiment 3 –In Vehicle Test**

This experiment was conducted in conjunction with Emissions Analytics, who operated both the seat pad and the BMW 5 Series 520D in which both the seat pad and phones were placed. As mentioned in Section 2.2, the seat pad is placed on the driver seat, underneath the driver throughout the drive while 2 phones were used in this experiment, Phones 1 and 4.

This experiment used 2 phone placements. Phone 4 acted as the “constrained” phone and was set up by securely attaching it to the bottom of the car floor with duct tape to prevent any sliding about or movement during the experiment. Phone 1, on the other hand was simply placed onto a passenger seat with no attempts at securing it, which acted as the “unconstrained” phone. Both phones were placed with the screens facing upwards. 4 repetitions of this experiment were carried out, each approximately a 2-hour journey around an area which covers a motorway, rural areas and urban areas. The exact journey is shown below in Figure 1. For confidentiality purposes, the coordinates representing the GPS location of the journey have been removed from the figure and the remainder of this report.

In addition, a Tom Tom camera was also mounted to the top of the car windscreen. This provided a video recording of the journey, which was then analyzed at a later stage to determine specific sections of the journey where the road surface changed significantly or to identify specific speed bumps. The camera also had the ability to determine the GPS coordinates of the entire journey which were used to compare against the GPS coordinates obtained by the phone.

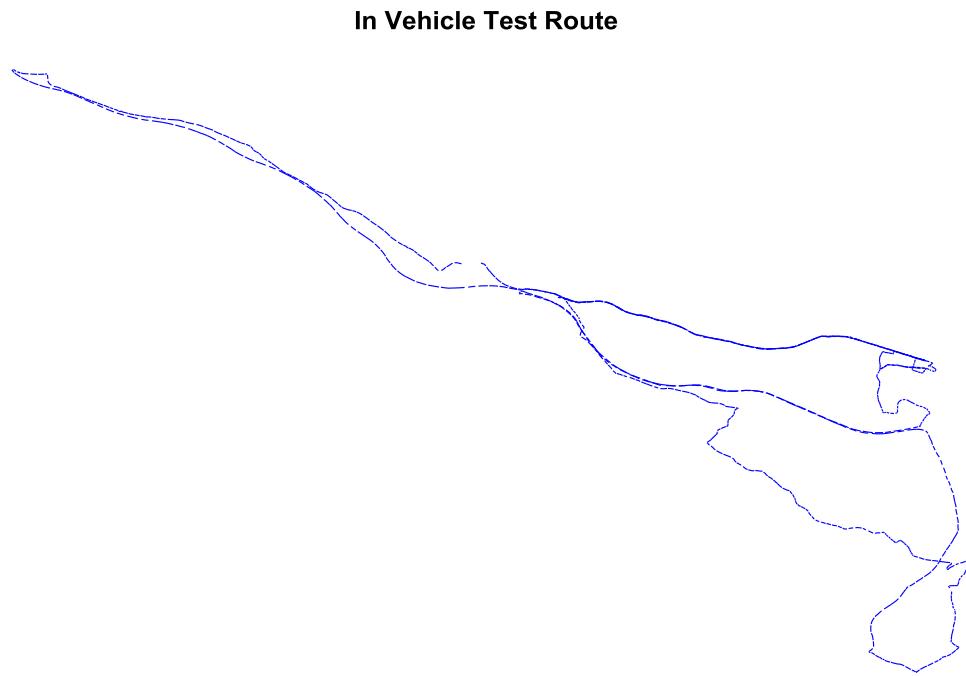


Figure 1 – Route taken for 2-hour journey in the In-Vehicle Test

#### 2.4 – Data Analysis

The app collects 6 sets of data at a frequency of 10Hz:

- Acceleration experienced by the phone along the X, Y and Z axis.
- Timestamp for each acceleration measured.
- Longitude and Latitude GPS coordinates
- Timestamp for each GPS coordinate.

The data is uploaded and stored in a database in JSON format, which then needs to be converted into a CSV file before data analysis can begin. The data is then read into MATLAB as a table and each data set stored as its own array (eg Acceleration\_X, Acceleration\_Y etc). The timestamps are also converted from their initial string values (eg 2017-18-5 19:21:07:455) into a corresponding Unix time, which is a measure of the number of seconds that have elapsed since 1<sup>st</sup> January 1970, 00:00:00, Coordinated Universal Time (UTC).

As the accelerometers in the phone measure proper acceleration, accelerations due to gravity ( $9.81\text{ms}^{-2}$ ) had to be subtracted from the value measured by the app when analyzing accelerations in the Z axis.

Ideally, the app would produce 10 data points per second, each corresponding to 0.1 of a second as dictated in the code. However, due to a combination of factors involving, but not limited to 4G signal strength, phone processing speed and app start/stop times, this is hardly the case in real life. In truth, the number of data points per second for each acceleration data set vary from 6 to 9 points, which are separated into “bins” of one second each.

The RMS value of each one second bin is then calculated throughout the entire journey and the results plotted without any further data manipulation. In some cases, the correlation coefficient, which describes how linearly dependent a set of data is compared to another is also computed via Equation 3.

$$R = \frac{1}{N-1} \sum_{i=1}^N \left( \frac{A_i - \mu_A}{\sigma_A} \right) \left( \frac{B_i - \mu_B}{\sigma_B} \right) \quad (3)$$

where  $\mu_A$  is the mean of A and  $\sigma_A$  is the standard deviation of A. Similarly,  $\mu_B$  and  $\sigma_B$  represents the mean and standard deviation of B.

In the case of the In-Vehicle Test, while every effort was made to calibrate the seat pad, an error of mismatched timing is unavoidable. Therefore, an extra iterative step, henceforth referred to as time alignment, is introduced at this stage to eliminate this error. By plotting the RMS values against time for the journey with both the data obtained from the phone app and the seat pad, certain peaks or patterns present in both data sets were identified. The data is then shifted forward or backward by a number of seconds using these peaks and patterns as a reference point and the correlation coefficient recalculated. The optimum time shift chosen for each time alignment step was then determined via the highest  $R$ .

In section 3.4, the coefficient of determination ( $R^2$ ) was also used to measure the accuracy of the model in predicting observed values by the phone app. This was computed via first computing the correlation coefficient using Equation 3 and squaring the result.

### 3 – Results and Discussion

This section is split into 4 parts, Location Accuracy, Accelerometer Proof of Concept, Accelerometer Accuracy and Ride Comfort Approximation, each with references to the experiments described in the previous section and how the data was used to come to the final conclusions.

It is also important to note that for the purposes of this thesis, most results focus on the analysis of accelerometer data in the Z axis. As the phones were all placed screen facing upwards, any large spikes in accelerations due to potholes or speed tables would be measured in the Z axis. While being able to compute ride comfort remains the primary aim of the project, doing so according to the guidelines set out in the ISO-2631 is currently impossible. This report further explains and

proposes a method of approximating a solution to this problem that is further described in section 3.4.

### 3.1 – Accelerometer Proof of Concept

#### Experiments: Preliminary Bicycle Test

The raw acceleration data was collected and plotted without any data manipulation. The result is shown below in Figure 2.

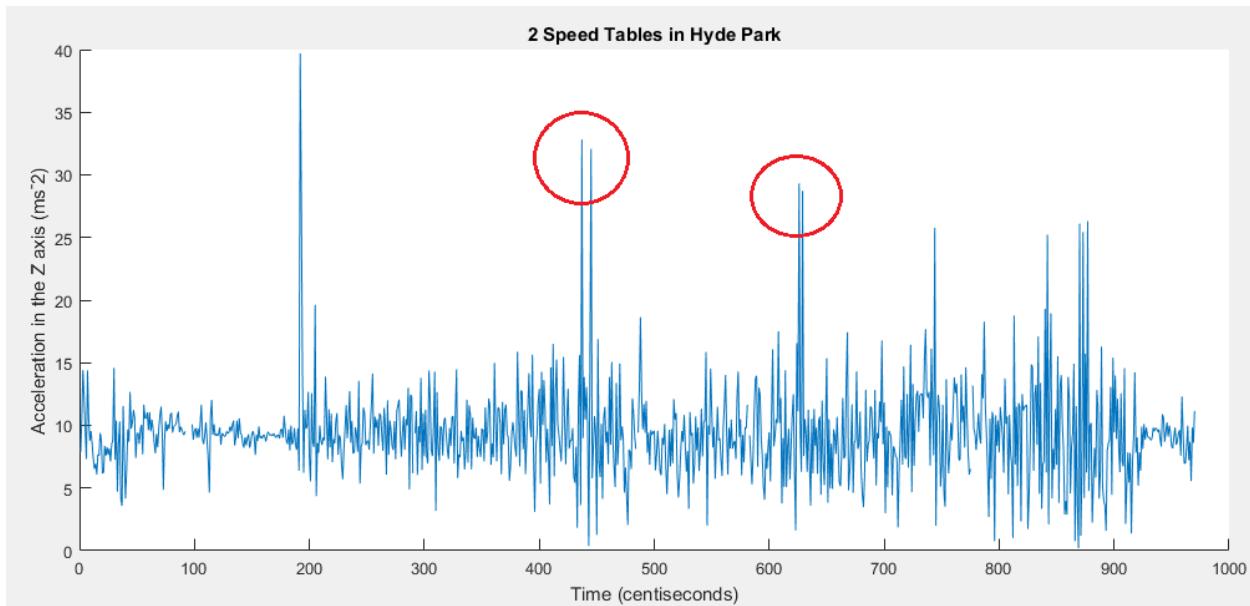


Figure 2 – Detection of 2 speed tables.

As can be seen from figure 2, there are several distinct peaks in the accelerometer readings, but only two (circled in red) that have a short time lag in between them. From Figure 3, it's clear that the presence of the time lag in between peaks fit the profile for data that one would expect from crossing 2 speed tables, each showing a peak in acceleration when the bicycle jerks upwards at the start of the speed table, returning to normal while the bicycle is travelling on the speed table and another peak upon exiting the speed table.



Figure 3 – Speed Table in Hyde Park, London (Google Maps, 2017)

### 3.2 – Location Accuracy

#### Experiments: Preliminary Bicycle Test & In-Vehicle Test

In the preliminary bicycle test, by using the GPS coordinates obtained from the phone, a plot of the overall journey taken was compared against the data obtained from the Strava app. This is shown below in Figure 4 and Figure 5. From the figures, it is clear that the GPS data for the overall journey obtained from both apps matched perfectly. In the case of the in-vehicle test, GPS data obtained from the Tom Tom camera was plotted against the GPS coordinates from the phone app. As can be seen from Figures 6 and 7, a similar result was achieved.

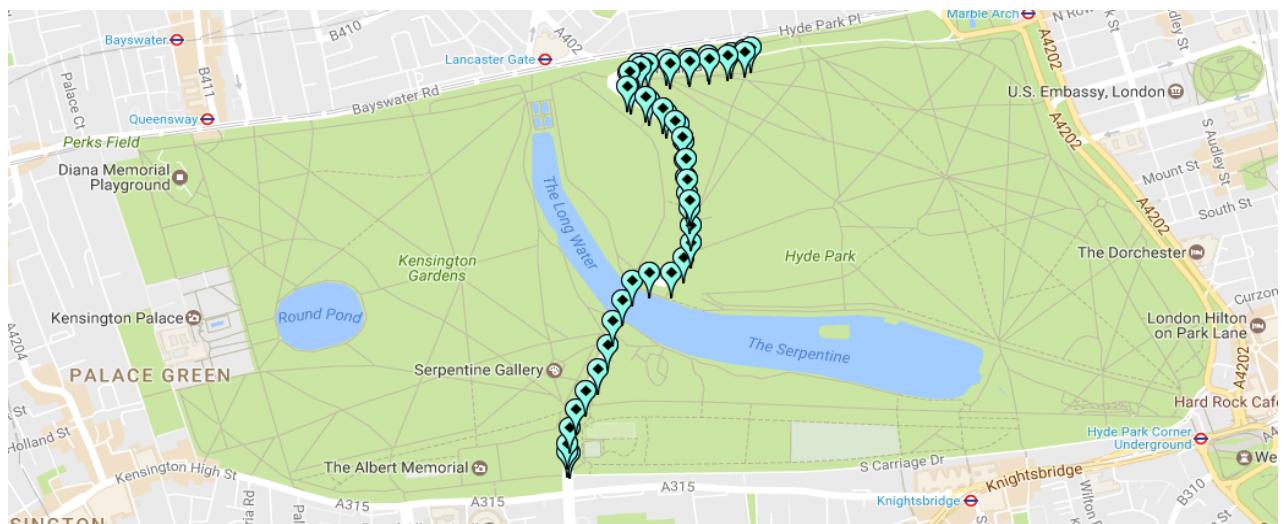


Figure 4 – Location Data obtained from the Ride Comfort App

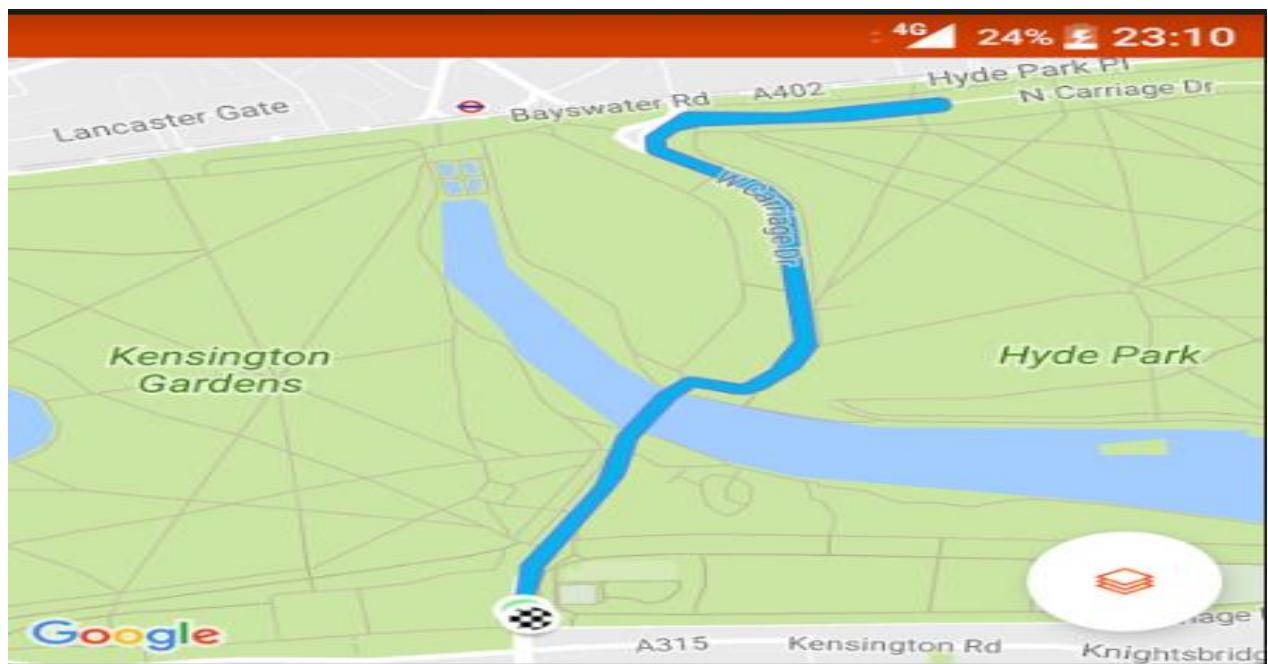


Figure 5 – Location Data obtained from the Strava app. (Strava, 2017)

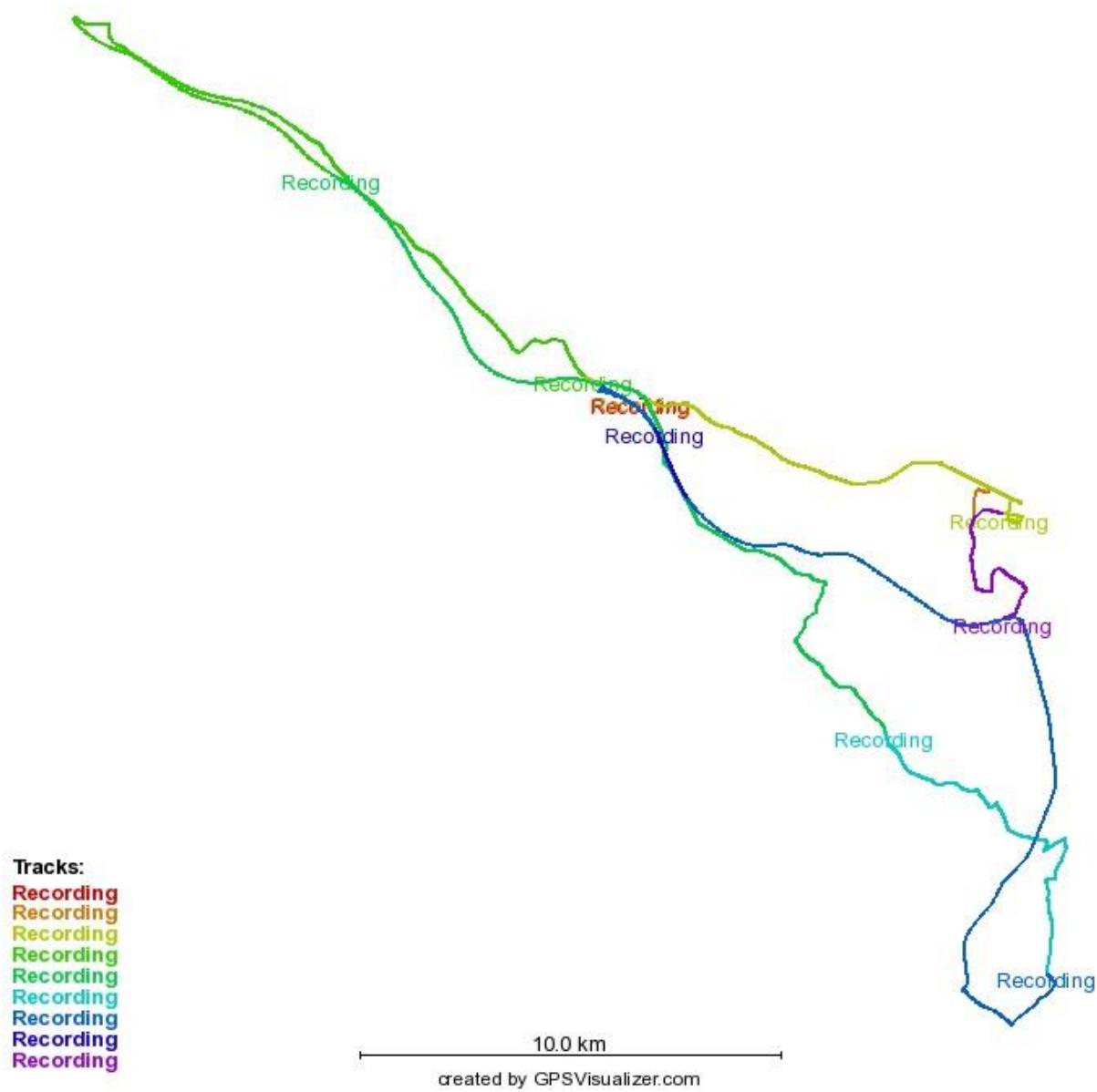


Figure 6 – Plot of GPS coordinates from Tom Tom Camera (GPSVisualizer, 2017)

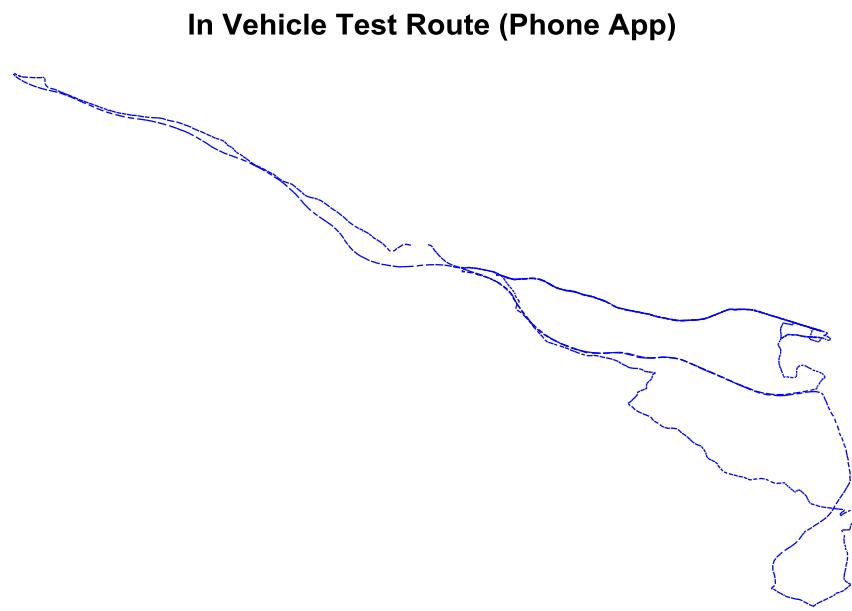


Figure 7 – Plot of GPS coordinates obtained from the ride comfort app.

Satisfied with the accuracy of the overall journey, it was also important to determine the accuracy of the app's ability to detect potholes. Using the speed tables encountered during the journey to simulate travelling over a "pothole", the next test for location accuracy for the preliminary bicycle test came in the form of trying to determine the exact location where the app detected the 2 large peaks previously mentioned in section 3.1.

By typing the GPS coordinates given by the phone app for these 2 peaks directly into Google Maps, both speed bumps were located with an error of 2.59m and 14.96m respectively. With the speed table itself being 3m wide, the error is insignificant as the speed table would be easily spotted from the location given by the phone app. The following figures, Figure 8 and Figure 9 are the images shown in Google Maps for each coordinate.

A similar test was performed on the data from the in-vehicle test. By locating significantly sharper peaks relative to the average RMS acceleration value, the GPS coordinates were determined and the speed table located on Google Maps. While the error in this experiment was 16.12m, with a speed table measuring approximately 5.8m wide as can be seen from Figure 10, it is still well within eyeshot of a person standing in the location specified by the phone app.



Figure 8 – Speed bump located by coordinates obtained from the ride comfort app.



Figure 9 – Second speed bump located by coordinates from ride comfort app



Figure 10 – Speed table located by coordinates from ride comfort app in the In-Vehicle Test

In conclusion, the phone app has an acceptable capability of locating potholes in the real world via the GPS coordinates collected via the app. However, comparison of the ride comfort app with other professional apps reveal that improvements can still be done to increase the accuracy of the measurement. One such example app called Physics ToolBox (Vieyra Software, 2017) detected the same speed table with an accuracy of 3m.

### *3.3 – Accelerometer Accuracy*

Experiments: Controlled Laboratory Test & In-Vehicle Test

#### **Controlled Laboratory Test**

From the data, it was concluded that only if the auto-rotation function on the phones were turned off, the axis of orientation of the phones would remain as shown in Figure 11. As both phones 1 and 3 were used, it can also be concluded that the orientation is consistent among all Android phones as long as the condition is met.



Figure 11 – Axis orientation for Phones 1, 2 and 3.

When phones 1 and 2 were secured together, the RMS acceleration in the Z axis was plotted and shown below in Figure 12. As can be seen from the figure, despite there being no attempts to time align the data, there is a strong linear relationship between the two sets of data with a correlation coefficient ( $R$ ) of 0.757. However, subsequent repetitions of this experiment showed varying degrees of correlation coefficients, ranging from 0.5 – 0.7. The reduced  $R$  was attributed to the time lag in between the 2 data sets and intentionally left in the results. Because both phones were connected to the same WiFi signal, there should not have been a time lag as both clocks were synchronized to internet time and therefore time alignment should not be necessary.

In the three phone version of this experiment, a similar result was achieved. Figure 13 below shows the RMS acceleration of Phone 1 plotted against Phone 3. The experiment was more successful with a  $R$  of 0.9248, proving that the difference in phone models does not affect the accelerations measured by the app. Despite this, repetitions of this experiment cause the correlation coefficient to fluctuate in the same manner as the two phone version of the experiment.

Based on the results, it can be concluded that the presence of an error will always exist in measurements, regardless if the phone used was the same model or not. Other factors like strength of internet connectivity and phone processing speed may also play a part in the existence and severity of the error and should be taken into account for any future conclusions drawn from data obtained using the app. However, with a range of  $R$  between 0.5 – 0.9, this is still definitive proof of the app's ability to consistently measure similar accelerations.

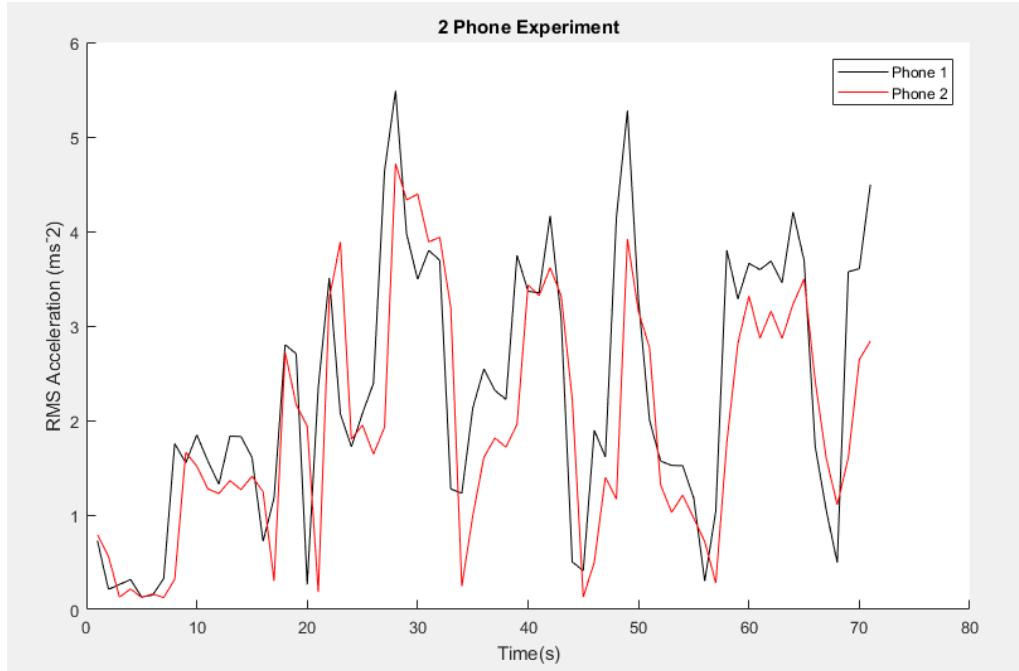


Figure 12 – RMS Acceleration in the Z axis for Phones 1 and 2.  $R: 0.757$

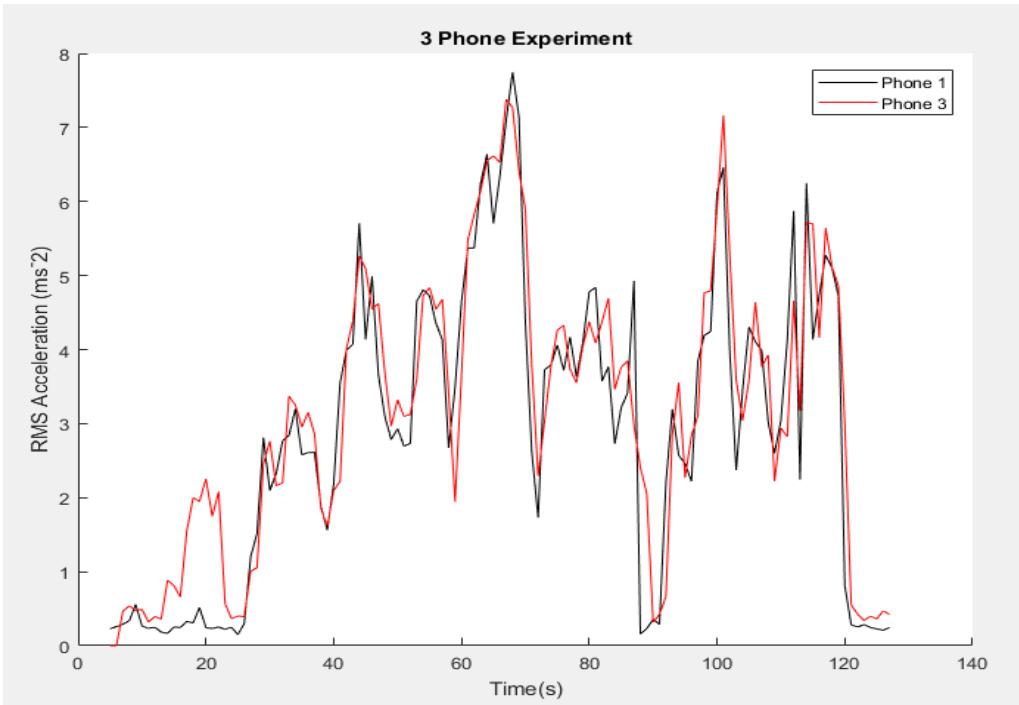


Figure 13 – RMS Acceleration in the Z axis for Phones 1 & 3.  $R: 0.9248$

### In-Vehicle Test

Several interesting conclusions were made after analyzing the data. This first part of the section looks at the unconstrained phone and its results. As mentioned before, four repetitions of the experiment were carried out. However, due to the constraints of this report, only two sets of

experimental results will be included. For clarity and ease of reading, each repetition will be labelled Journey 1 and Journey 2. First, this report looks at the unconstrained phone for both journeys.

### Journey 1 – Unconstrained Phone

In the following Figure 14, RMS acceleration in the Z axis recorded by the phone was plotted against time and compared with a similar plot with data obtained by the seat pad. There is barely, if any similarity between the two plots with a corresponding  $R$  of 0.1172. From the phone app, the first half of the journey seems to have an average RMS acceleration of approximately  $2.5\text{ms}^{-2}$  that doubles to  $5\text{ms}^{-2}$  in the second half of the journey. However, the seat pad shows no such change in average value and seems to have a fairly consistent average value throughout the entire journey. In addition, roughly 35 minutes (2100 seconds) into the journey, the phone app registered several large spikes that lasted 250 seconds. This is nowhere to be found in the seat pad data.

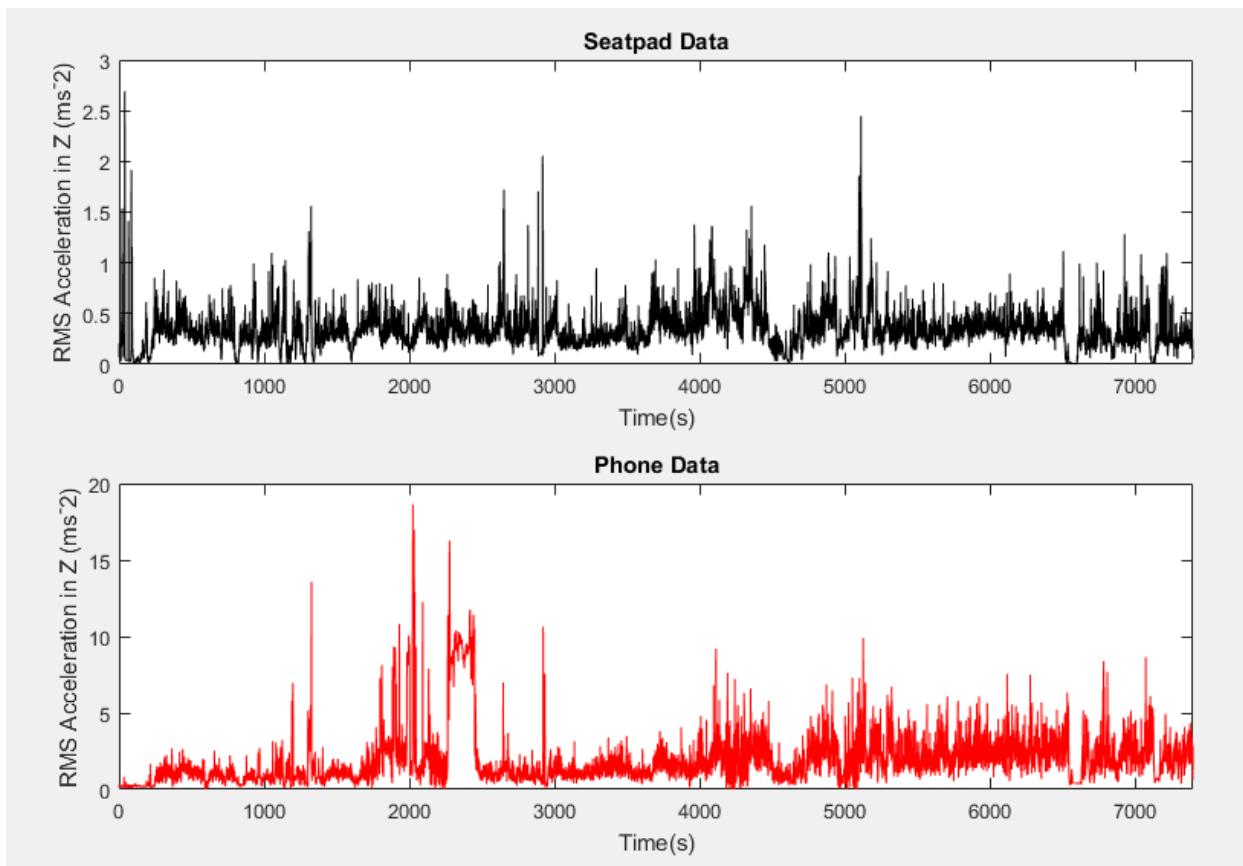


Figure 14 – RMS Acceleration in the Z axis of the entire journey 1.  $R: 0.1172$

However, with data obtained from the seat pad and the confined phone, it can be safely concluded that the several sharp spikes during the 35 minute mark were due to the phone being picked up and placed back after dropping off the car seat. In order to account for this anomaly, only the second half of the journey was analyzed once more. The result is much better with a  $R$  of 0.4516 and clearly shows similar patterns that are shared between the two data sets. As can be seen from Figure 15, similarities include the sharp trough at the 4500 second mark, the sharp trough followed by the

highest peak in the entire second half of the journey at the 5000 second mark and the U shaped chasm slightly past 6500 seconds into the journey.

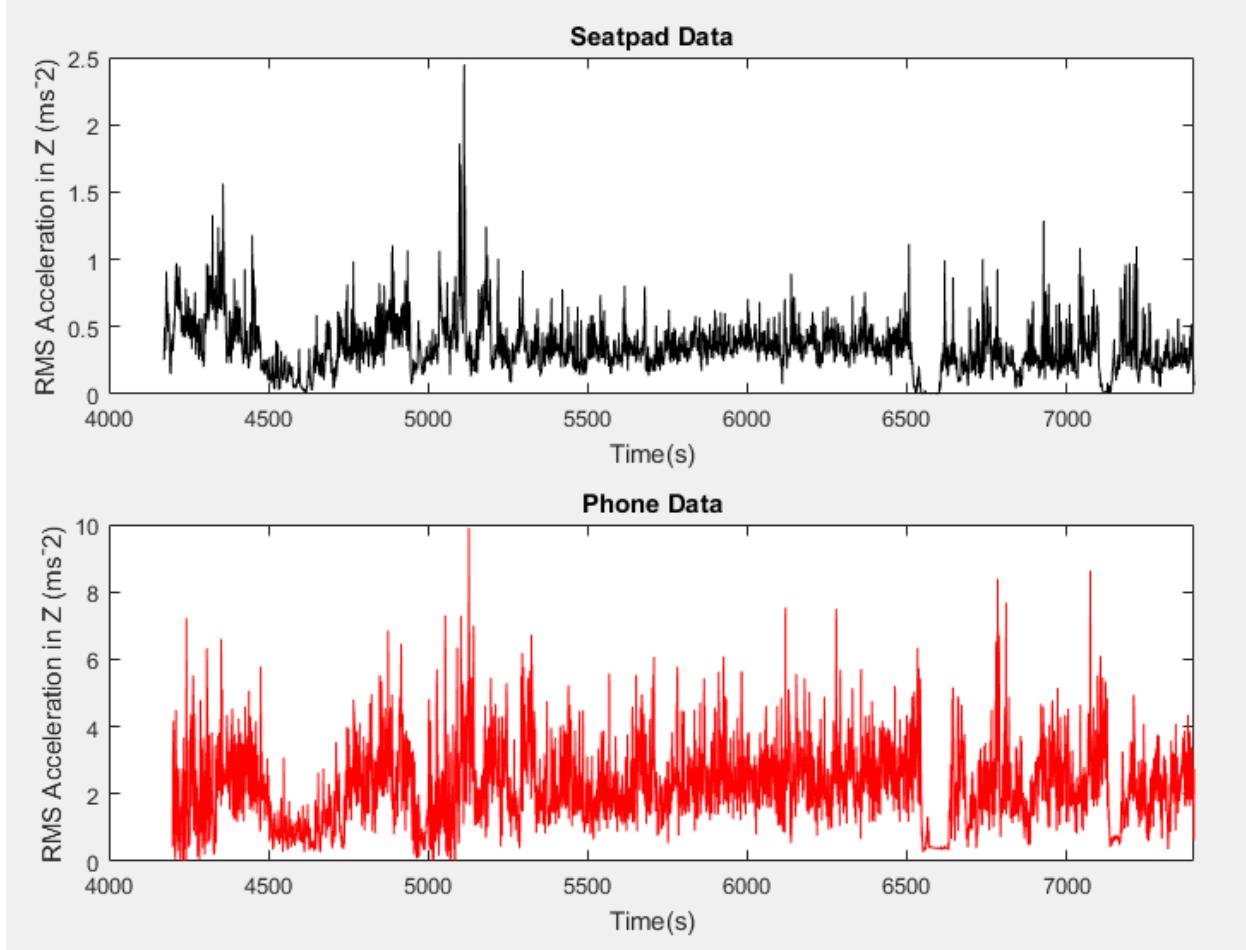


Figure 15 – RMS Acceleration in the Z axis of the second half of journey 1.  $R: 0.4516$

### Journey 2 – Unconstrained Phone

Results similar to Journey 1 with the unconstrained phone were observed in this journey. As can be seen from Figure 16, data obtained from the first half of the journey contains an anomaly, in this case, a period of 300 seconds at the 3000 second mark where nothing was detected by the phone app. Despite the low  $R$  of 0.2605, certain patterns can be identified and a second analysis of the data was done only on the second half of the journey. Figure 17 shows the updated result with a  $R$  of 0.7021 between the two sets of data, a value that was far higher than expected from an unconstrained phone, and can be seen as tentative proof that simply leaving the phone on a car seat can be a fairly accurate method of accurately measuring accelerations.

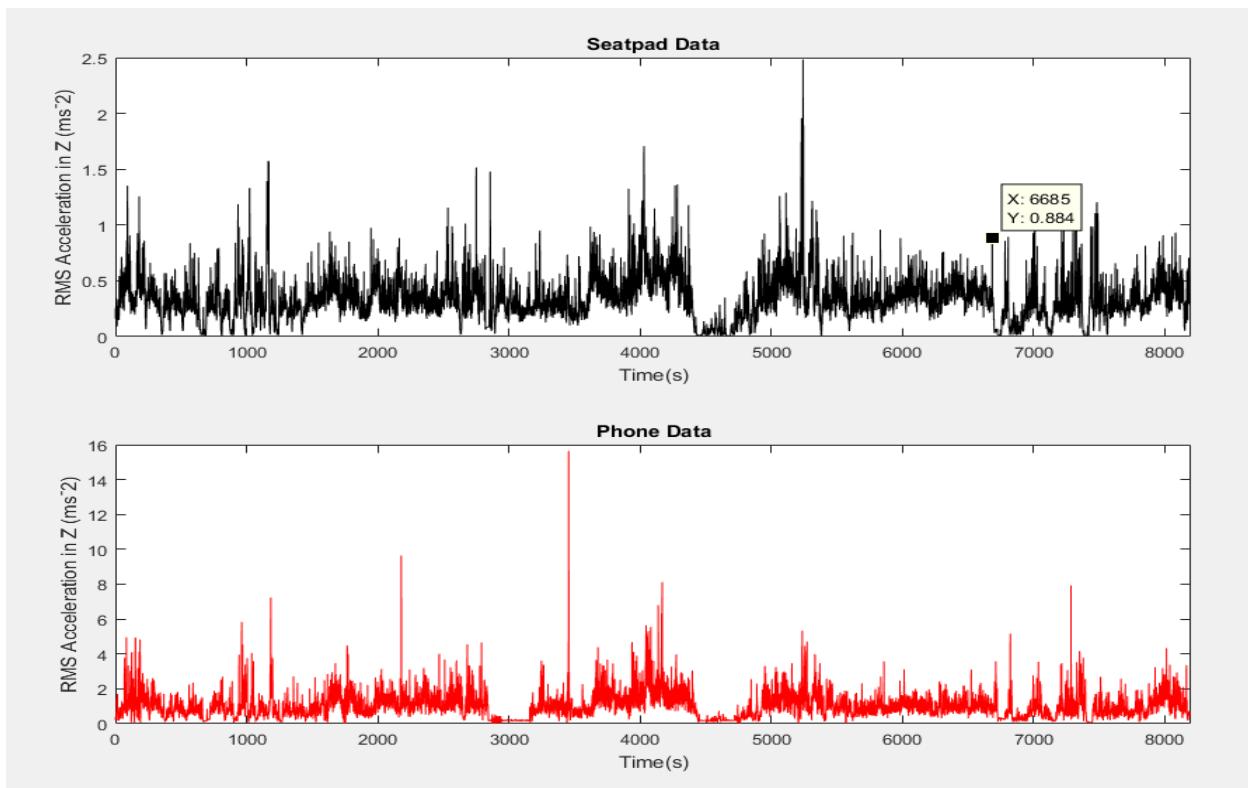


Figure 16 – RMS Acceleration in the Z axis of the entire journey 2.  $R: 0.2605$

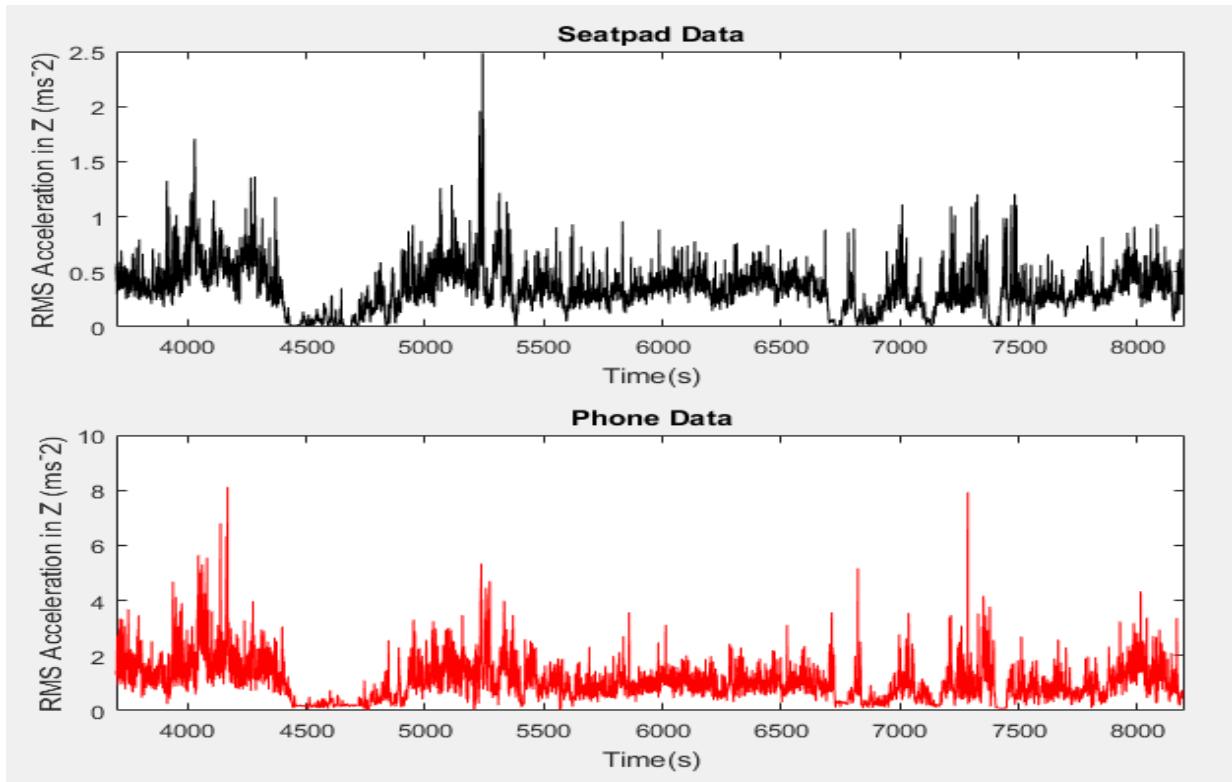


Figure 17 – RMS Acceleration in the Z axis of the second half of journey 2.  $R: 0.7021$

Similar to the unconstrained case, this next section will describe the results obtained from the constrained phone. It is worth re-mentioning here that with the in-vehicle tests, the goal was to prove that the phone app can measure accelerations similar to those detected by the seat pad. Due to the seat pad automatically frequency weighting its results and the phone app's inability to do so, the magnitudes of the RMS values of acceleration obtained by the phone app and seat pad cannot be directly compared. However, what is important is that the patterns of accelerations measured by both phone app and seat pad remain similar.

### Journey 1 – Constrained Phone

In the following Figure 18, the RMS acceleration in the Z axis recorded by the phone was plotted against time and compared with a similar plot with data obtained by the seat pad. Without any manipulation, the 2 data sets were correlated, with a  $R$  of 0.5068. Similar patterns in the data were easily identifiable and the average RMS value remained consistent throughout the entire journey. Purely for comparison purposes against the unconstrained case, Figure 19 shows the results when only the second half of the journey was analyzed and the resulting  $R$  slightly improved at 0.5611.

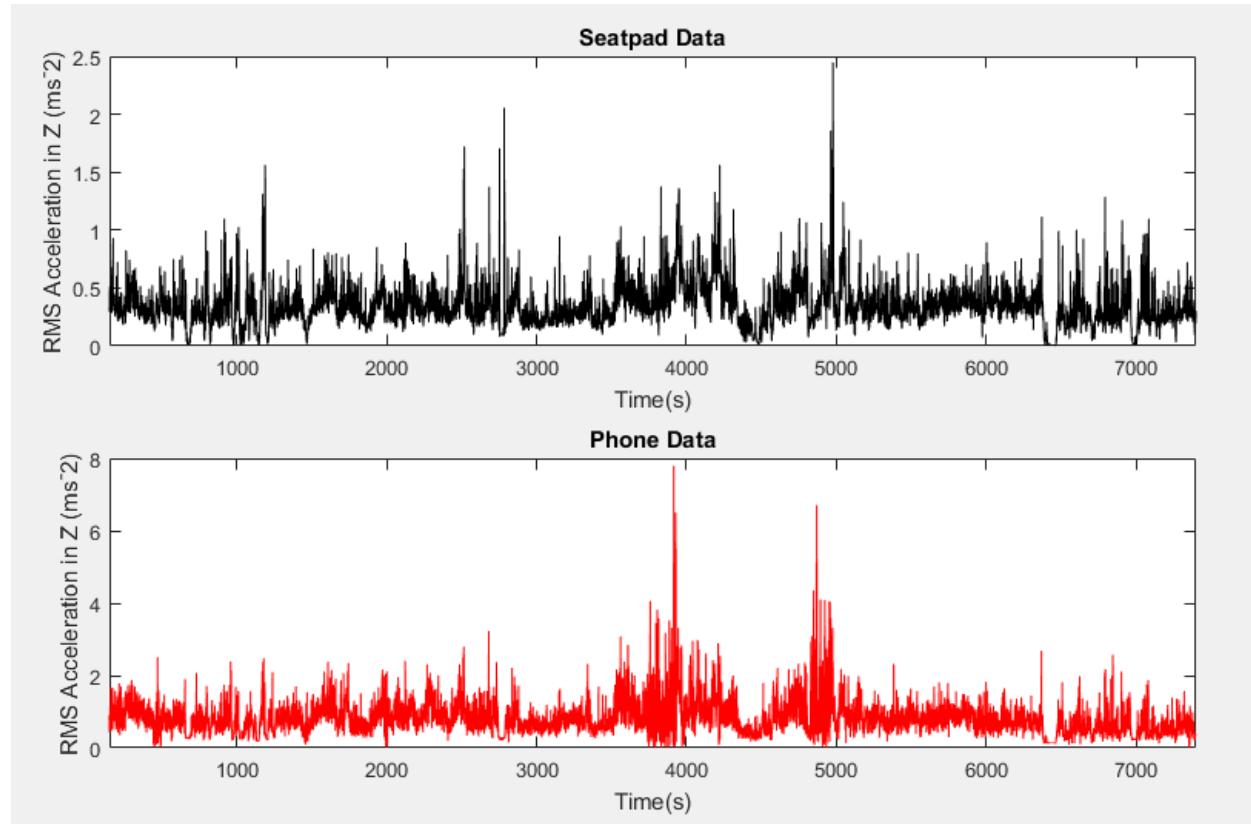


Figure 18 – RMS Acceleration in the Z axis of the entire journey 1.  $R$ : 0.5068

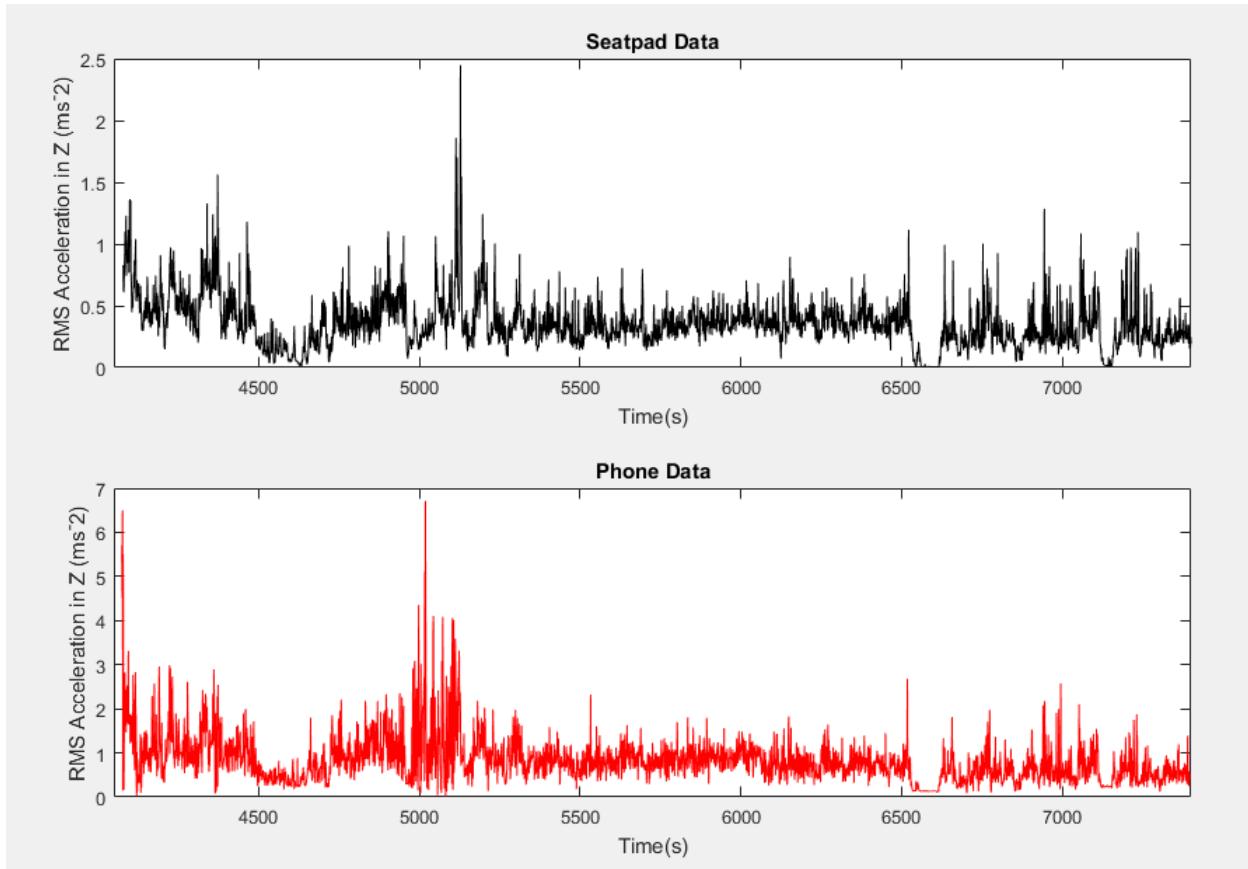


Figure 19 – RMS Acceleration in the Z axis of the second half of journey 1.  $R: 0.5611$

### Journey 2 – Constrained Phone

As with the results obtained from Journey 1 with the constrained phone, the accelerations measured by the phone in journey 2 also exhibits a moderate linear relationship with the seat pad with a  $R$  of 0.4660. Obvious patterns shared between the two data sets can be seen from Figure 20 without any manipulation of the data in post processing. There is a slight time lag between the two sets of the data however, and time aligning the data resulted in an improved  $R$  of 0.5496 and 0.5485 for the first and second half of the journey respectively. The time aligned data is shown in Figures 21 and 22.

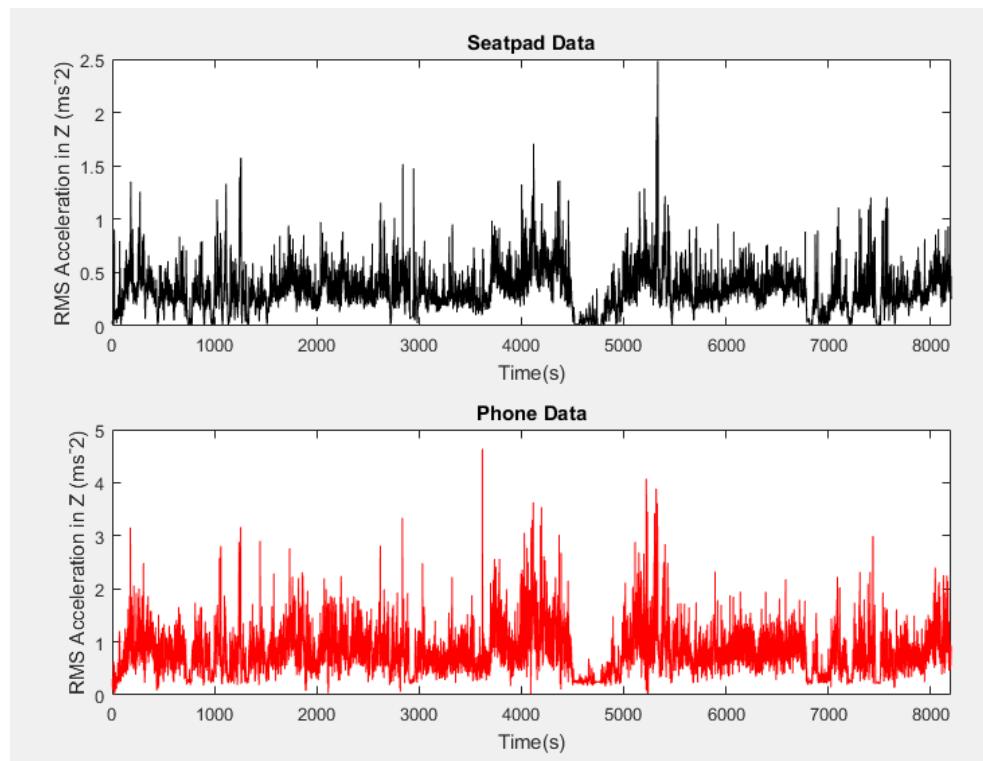


Figure 20 – RMS Acceleration in the Z axis of the entire journey 2.  $R: 0.4660$

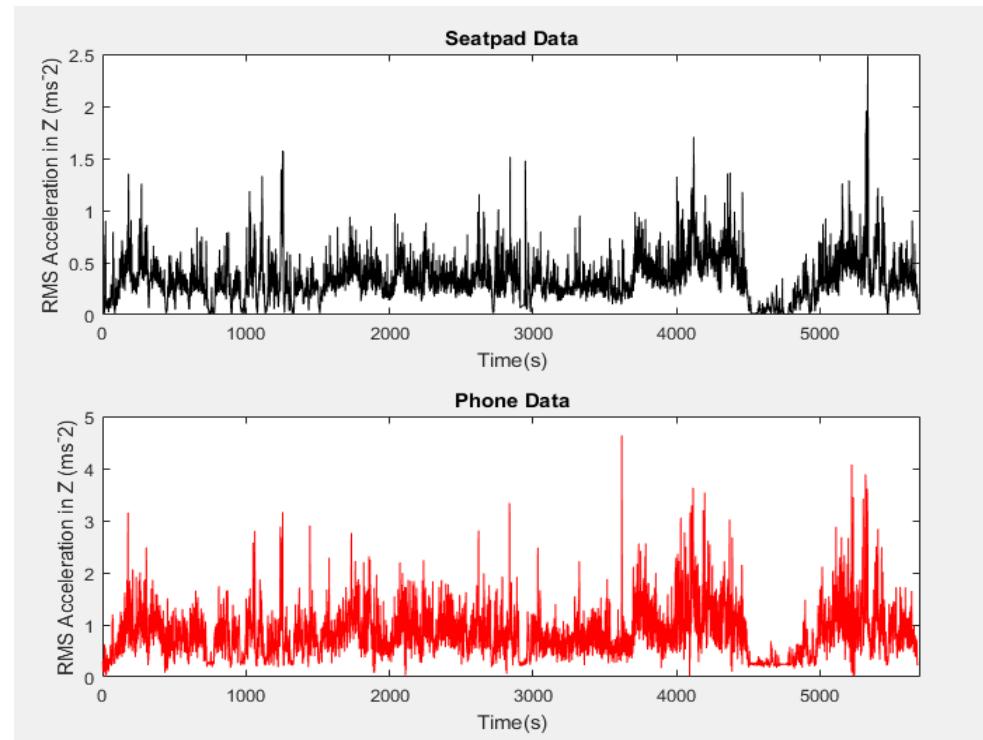


Figure 21 – RMS Acceleration in the Z axis of the first half of journey 2.  $R: 0.5499$

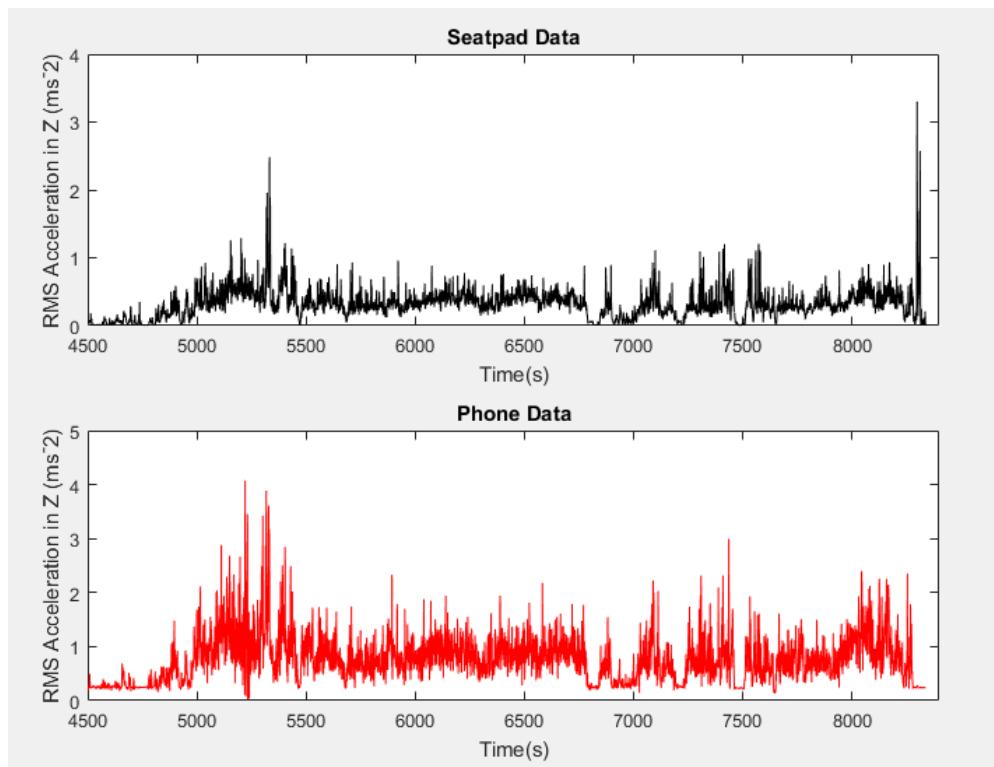


Figure 22 – RMS Acceleration in the Z axis of the second half of journey 2.  $R: 0.5485$

In summary, both phone placements each have their advantages and disadvantages. With the unconstrained phone, it is much simpler to set up but the  $R$  ranges between 0.11 to 0.7. While in some cases it can be more accurate than the constrained phone in measuring accelerations, this is only possible when there is some form of human interaction at the data processing stage (ie: Determining what sections of the journey to disregard as an obvious anomaly). In contrast, data from the constrained phone is consistently consistent with the seat pad data with a  $R$  of 0.5 but would require more effort to set up. As the app relies on crowdsourcing via the mass public, the app should be as easy to set up and use as possible in order to prevent alienating potential users. Despite this, the constrained phone placement would still be recommended due to the reduced variability in measuring accelerations and in order to reduce the already computationally expensive task of data analysis and the extra set up effort required can easily be remedied with a phone mount installed in the car.

### 3.4 – Ride Comfort Approximation

#### Experiment: In-Vehicle Test

In order to use the guidelines for road comfort outlined in the ISO-2631, the frequency weighted RMS acceleration must be calculated. This requires accelerations in all three axis to be frequency weighted and a suggested weightage applied to each acceleration, depending on the axis, during

summation of the weighted accelerations. Refer to section 1.1.1 for the exact equation suggested by the ISO-2631 standard. However, due to the ride comfort app's limited capabilities to only measure accelerations at a single frequency (10Hz), determining the actual ride comfort level via the suggested values is currently impossible. However, an approximation of the comfort level can be determined by comparing the seat pad data, which conforms to the ISO-2631 standard, to the accelerations obtained from the app and obtaining an acceptable scaling factor.

For this section, data was taken from the In-Vehicle test experiment using a constrained phone. Figure 23 below shows the journey split into 3 different sections based on the area in which the car travelled through, namely Rural, Urban and Motorway sections. The data from each section was then analyzed via 2 different methods.

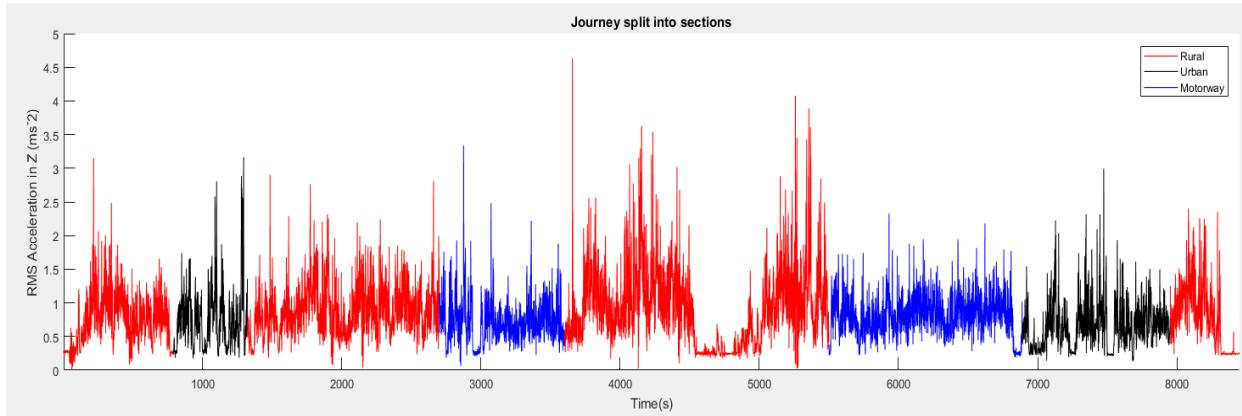


Figure 23 – Ride comfort app RMS Accelerations in Z axis in the Rural, Urban and Motorway sections.

### 3.4.1 – Section by Section Analysis using RMS Accelerations in the Z axis only.

Time alignment was carried out at every section and each section analyzed independently. Firstly, a scatter plot of RMS accelerations in the Z axis obtained via the phone app was plotted against the corresponding seat pad values. A line of best fit was then fitted to the resulting plot using the least square method and the goodness of fit measured via the coefficient of determination,  $R^2$ . The residual plots for each section was also plotted to allow analysis of the error. Table 3 summarizes the findings and links each individual analysis with the corresponding figure and full analysis in the appendix.

Table 3 – Section by Section Analysis (RMS Acceleration in Z axis only)

Section	Time (seconds)	Correlation Coefficient, $R$	Gradient of Line of Best Fit, m	Coefficient of Determination, $R^2$	Appendix
Rural 1	1-750	0.6675	1.4125	0.446	B.1
Urban 1	751-1283	0.7785	1.4133	0.308	B.2
Rural 2	1284-2662	0.5380	1.4091	0.409	B.3
Motorway 1	2663-3562	0.5228	1.1094	0.378	B.4

Rural 3	3563-5462	0.7567	1.6533	0.573	B.5
Motorway 2	5463-6852	0.5864	1.2793	0.344	B.6
Urban 2	6853-7914	0.7131	1.2711	0.509	B.7
Rural 4	7915-8250	0.5577	1.5798	0.311	B.8

From the results, the first obvious takeaway is the improved  $R$ . In section 3.3, time alignment was carried out either for the journey as a whole or for much larger “sections” (first half or second half of the journey) and an average  $R$  of 0.5 was obtained. By breaking the journey into smaller sections based on road type, a  $R$  of up to 0.7785 was achieved.

As can be seen from the corresponding residual plots in the appendix, every journey section had errors that are largely centered around the zero value and are not systematically high or low, which is characteristic of stochastic errors and therefore allow the use of  $R^2$  as a measure of how well the model will predict observed values.

As a general rule of thumb,  $R^2$  values of 0.5 and above are considered acceptable when attempting to test how well a model predicts observed values. Based on this, the app is found lacking as only 2 sections (Rural 3 and Urban 2) have  $R^2$  values above 0.5. If the overall journey was considered, this would mean that the app could only acceptably predict observed values 25% of the time.

#### 3.4.2 – Section by Section Analysis using total RMS Accelerations in all 3 axis.

The exact same analysis was then repeated, this time using the total RMS Accelerations in all 3 axis from both the phone app and the seat pad instead of only the RMS accelerations in the Z axis as was done previously. Table 4 below summarizes the findings and links each individual analysis with the corresponding figure and full analysis in the appendix.

Table 4 – Section by Section Analysis (Total RMS Acceleration in all 3 axis)

Section	Time (seconds)	Correlation Coefficient, $R$	Gradient of Line of Best Fit, m	Coefficient of Determination, $R^2$	Appendix
Rural 1	1-750	0.5208	2.5052	0.271	C.1
Urban 1	751-1283	0.6262	2.5011	0.392	C.2
Rural 2	1284-2662	0.5158	3.1290	0.266	C.3
Motorway 1	2663-3562	0.4645	2.2519	0.216	C.4
Rural 3	3563-5462	0.6392	2.8745	0.409	C.5
Motorway 2	5463-6852	0.3981	2.1258	0.158	C.6
Urban 2	6853-7914	0.5990	2.4953	0.359	C.7
Rural 4	7915-8250	0.4710	2.7368	0.222	C.8

From the results, it can be concluded that the model using total RMS acceleration values is worse compared to the model that uses RMS accelerations in the Z axis only. Not a single section has an  $R^2$  value that is above 0.5 while the  $R$  can also be observed to be smaller in every section. While

this may be due, either in part or in full, to the fact that the “Auto-Rotation” function was not turned off during the experiment, further testing will have to be done to properly determine why the total RMS acceleration model pales in comparison to the model that uses only RMS accelerations in the Z axis.

### *3.5 – Locating Potholes*

#### Experiments: In-Vehicle Test

As mentioned before, pothole detection represents one of the major objectives in the development of this app. By using data from a constrained phone in the in-vehicle test, the accuracy of the app in detecting a large change in road quality or potholes when registering a large spike in acceleration was determined.

Like in Section 3.4, the entire journey was split into several sections depending on the road type and each section analyzed independently. First, the mean RMS acceleration was determined and divided into intervals as defined below.

Table 5 – Definition of Intervals

Interval	Range of mean RMS Acceleration
A	1 -1.5
B	1.5 -1.6
C	1.6 -1.7
D	1.7 -1.8
E	1.8 -1.9
F	> 1.9

Figure 24 shows a sample of the resultant graph for an individual section.

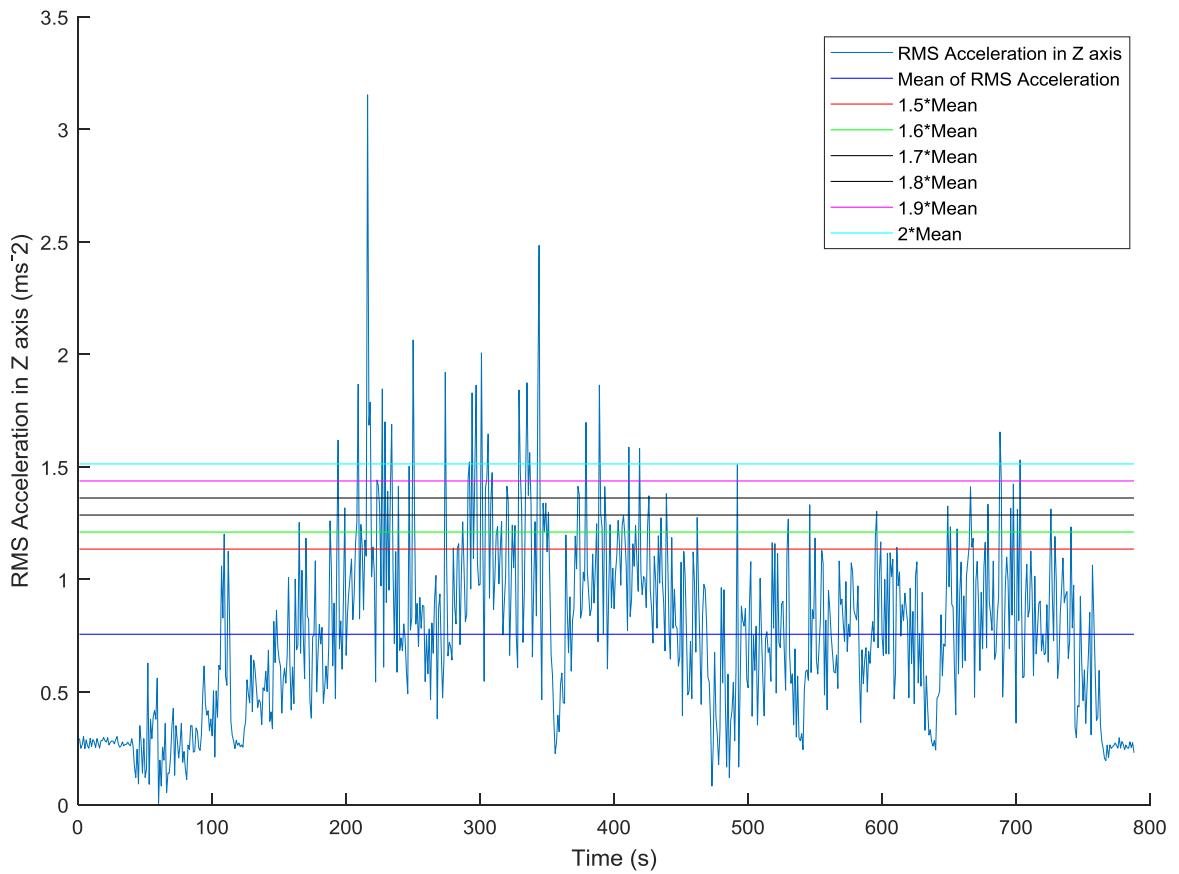


Figure 24 – Sample Section of RMS Accelerations in the Z axis split into several interval ranges shown by the multicolored horizontal lines.

4 points that peaked within each interval were then randomly chosen, depending on availability. For motorway and urban sections, more points were identified as the journey consisted of 4 rural sections but only 2 motorway and 2 urban sections. For each point chosen, the corresponding GPS location was identified and analyzed via Google Maps. By visual inspection, if potholes, speed bumps or cracks in the road that could potentially cause the car to jerk upwards were present, that point was given a score of 1, and if not, a score of 0. Table 5 below summarizes the results obtained.

Table 6 – Experimental Results from individual peaks. 1 represents a road defect and 0 represents a road without defects.

Interval	Rural 1	Urban 1	Rural 2	Motorway 1	Rural 3	Motorway 2	Urban 2	Rural 4
A	0 0 0	0 0 0 0 0	0 0 0 1	0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 0 1	0 1 1
B	1 0 0 0	0 0 0 1 1	0 1 1 0	0 0 0 0	0 0 1	0 0 0 0	0 0 1 0 1	0 0
C	0 0 0 0	0 0 1 1	1 0 0	0 1 0 1	0 0 0	0 1 1 0	1 0 0 0	0 0

D	1 0 0 1	1 0 0	1 0 1	1 0 1 0 0	1 0	0 0 1	1 1 0 0	1 1 0
E	1 0 1 1	1 0 0 1	0 0 1 1	0 1 0 1	0 1 0 1	0 0 1 1	1 0 0 1	0
F	1 0 1 1	1 0 1 1	1 1 0	1 0 1 1 1	1 0 1 1	0 1 1 0	1 0 1 1	0 1 1 0

Based on the results summarized in Table 6 below, the app detects potholes with varying accuracies depending on the road type. However, it can be concluded that a general rule of thumb to follow for any road section is to take note of any peak that is greater than 1.9 times the mean value as it has shown to detect road defects with an accuracy of approximately 70%. Other intervals have a higher percentage of false positives (i.e phone app registering a peak in accelerations in the absence of a road defect) and are deemed unreliable.

Table 7 – Summarized Results for pothole locating accuracy

Interval	Rural	Urban	Motorway
A	28.0%	20.0%	12.5%
B	30.0%	40.0%	0.0 %
C	8.3%	37.5%	50.0 %
D	58.3%	42.8%	44.4%
E	53.8%	50.0%	50.0%
F	71.4%	66.7%	70.0%

While this method might be able to approximate the accuracy of the phone app in successfully locating potholes, there is a degree of subjectivity involved. Limitations of the method include:

- Subjectivity in what the author believes to be significant enough to give a particular road section a score of 1 as opposed to 0.
- How updated Google Maps is compared to the actual road conditions.
- Certain interval ranges do not have 4 or more points that peak within the range and thus not enough information could be collected regarding that interval range.
- Certain interval ranges have more than 4 points that peak within the range and not every peak could be analyzed.

For further reading, appendix D contains examples of road sections in which the author gave a score of 1 and examples of road sections that were assigned a score of 0. In addition, the complete set of graphs depicting the intervals for each road section are also included in appendix D.

#### 4 – Conclusions and Future Work

With the number of vehicles on the road increasing every year and the expensive nature of potholes, both financially and in terms of human safety, a low cost, widely accessible method of measuring road quality and safety is paramount. This project aimed to solve this problem by building an application for mobile phones that could be easily installed to millions of smartphones

around the world. A prototype ride comfort app was built that uses the accelerometer and GPS sensor to record and upload acceleration and location data to a server.

While this project is still very much in its infancy, preliminary testing conducted during the course of this project show that an app is capable of replacing expensive specially made devices for measuring ride comfort. Spikes in acceleration were detected by the phone app when travelling across potholes and speed bumps. The acceleration data measured by the app was proven to have a moderate to strong linear relationship with a correlation coefficient of 0.5 – 0.7 compared to data measured by the HVM200. The capability to detect potholes was also proven to be possible. Upon detection of a large spike, it was found that the app would return GPS coordinates that show the presence of potholes with approximately 70% accuracy when using a factor of 1.9 times the mean acceleration to filter through the acceleration data.

Comparison between different phone models show that the accelerations measured have a moderate to strong linear correlation ( $R$  of 0.5 – 0.9). Difference in phone models was therefore concluded to have little to no consequence on the ability of the app to measure ride comfort and detect road defects. In addition, while both unconstrained and constrained phone placements have proven capable of measuring accelerations similar to the industry standard HVM200 with a  $R$  of 0.1 - 0.7 and 0.5 - 0.7 respectively, the use of a constrained phone placement was recommended due to its consistency, reduced variability in measured accelerations and reduced computational effort required in analyzing data.

Moving forward, there are still various facets of the project that have yet to be considered or can be improved upon. Firstly, the destination of the data upload process (the database in which the data is stored) is hard coded into the app with no way of differentiating between sets of data coming from different phones. In this project, multiple different databases were used by recoding the app specifically for this project, but with mass production to the public in mind, a method of separating uploaded data from different phones would have to be designed and implemented.

Battery life is another important consideration that should be taken into account before adoption by the public. A 2 hour experiment consumed approximately 40% of a fully charged phone's battery and this is an unacceptable level as it will greatly deter the public from installing and using the app. Further improvements to the code will have to be made in order to make the app more efficient.

In addition, further analysis of the data should be undertaken in order to fit a better model to the acceleration data. Currently, the model can only acceptably predict the observed values 25% of the time, which leaves plenty of room for improvement.

Furthermore, the app currently relies on continuous, uninterrupted internet connectivity in order to upload the measured data to the database every 10 seconds. Should the app be unable to connect to the database due to a lack of signal, the data is lost and the app simply continues with the experiment, leaving a 10 second gap in the results in which there is no data. In order to combat this, an offline version of the app should be trialed, where data for every single journey can be stored into the phone's memory and uploaded only once the experiment is complete.

Another thing to consider is the presence of time lag in the data recorded using the app. From the controlled laboratory tests, it was concluded that despite using the same phone model and connecting it to the same WiFi, the data obtained did not match perfectly in time. While this can

be classified as a secondary issue, since importance was placed on ensuring both phones successfully measuring the same location, further testing should be conducted to determine if this particular error in time can be eliminated.

Finally, post processing of the data is a lengthy process depending on the length of the experiment. One repetition of the in-vehicle test (2 hour journey) resulted in a 238 megabyte large CSV file that took a computer over 2 hours to process in Matlab. Unfortunately, that was merely to analyze the acceleration data obtained in the Z axis.

## Acknowledgements

The author would like to express gratitude to Dr Marc Stettler and Dr Simon Hu for their time, patient guidance and amazing support over the course of this project. Special thanks should also be given to the staff at Emission Analytics for going out of their way to support this project.

## References

- Abulizi, N., Kawamura, A., Tomiyama, K. & Fujita, S. (2016) Measuring and evaluating of road roughness conditions with a compact road profiler and ArcGIS. *Journal of Traffic and Transportation Engineering (English Edition)*. 3 (5), 398-411.
- Asakura, Y. & Iryo, T. (2007) Analysis of tourist behaviour based on the tracking data collected using a mobile communication instrument. *Transportation Research Part A: Policy and Practice*. 41 (7), 684-690.
- Asphalt Industry Alliance. (2016) *Annual Local Authority Road Maintenance (ALARM)* Bristol. Available from: [http://www.asphaltuk.org/wp-content/uploads/ALARM\\_survey\\_2016.pdf](http://www.asphaltuk.org/wp-content/uploads/ALARM_survey_2016.pdf) [Accessed 24th May 2017]
- Azzoug, A. & Kaewunruen, S. (2017) Ride Comfort: A Development of Crowdsourcing Smartphones in Measuring Train Ride Quality. *Frontiers in Build Environment*. Available from: doi: 10.3389/fbuil.2017.00003. [Accessed 6<sup>th</sup> June 2017]
- Cundill, M.A. (1991) MERLIN – A low cost machine for measuring road roughness in developing countries. In: *International Conference on Low-Volume Roads May 19-23 1991, Raleigh, North Carolina, USA; volumes 1 and 2*. Transportation Research Board. pp. 106-112.
- Deloitte. (2016) *Consumer usage patterns in the era of peak smartphone. Global mobile consumer survey 2016: UK Cut.* Deloitte LLP. Available from: <https://www.deloitte.co.uk/mobileuk/assets/pdf/Deloitte-Mobile-Consumer-2016-There-is-no-place-like-phone.pdf> [Accessed 15<sup>th</sup> May 2015]

Department for Transport (2016) *Vehicle Licensing Statistics Quarter 1 (Jan-Mar) 2016*. Department for Transport. Available from:  
[https://www.gov.uk/government/uploads/system/uploads/attachment\\_data/file/528143/vehicle-licensing-january-to-march-2016.pdf](https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/528143/vehicle-licensing-january-to-march-2016.pdf) [Accessed 5th June 2017]

Ekchian, J., Graves, W., Anderson, Z., Giovanardi, M., Godwin, O., Kaplan, J., Ventura, J., Lackner, J. R. & Dizio, P. (2016) A High-Bandwidth Active Suspension for Motion Sickness Mitigation in Autonomous Vehicles. SAE 2016 World Congress and Exhibition, April 12, 2016 - April 14. 2016, Detroit, MI, United states, SAE International.

Eriksson, J. & Svensson, L. (2015) *Tuning for Ride Quality in Autonomous Vehicle*. Master's Thesis. Uppsala University

Golding, J.F., Mueller, A.G., and Gresty, M.A. (2001). A motion sickness maximum around the 0.2 Hz frequency range of horizontal translational oscillation. *Aviation Space and Environmental Medicine* 72(3), 188-192.

Google Maps (2017) *Map of Speed Table in Hyde Park*, Google. Available from: <https://www.google.co.uk/maps/> [Accessed 5<sup>th</sup> June 2017]

Google Maps (2017) *Map of Speed Table*, Google. Available from: <https://www.google.co.uk/maps/> [Accessed 5<sup>th</sup> June 2017]

GPSVisualizer (2017). Draw an SVG, JPEG, or PNG map. Available from: [http://www.gpsvisualizer.com/map\\_input?form=png](http://www.gpsvisualizer.com/map_input?form=png) [Accessed 5<sup>th</sup> June 2017]

Jaguar Land Rover (2017) *Pothole Detection Technology Research Announced by Jaguar Land Rover*. Available from: <https://www.landrover.com/experiences/news/pothole-detection.html> [Accessed 24th May 2017]

Jo, Y., & Ryu, S. (2015) *Pothole Detection System Using a Black-box Camera*. Korea Institute of Civil Engineering and Building Technology, Korea.

Kilinc, A. & Baybura, A. (2012) *Determination of minimum horizontal curve radius used in the design of transportation structures, depending on the limit value of comfort criterion lateral jerk*. Available from: [https://www.fig.net/resources/proceedings/fig\\_proceedings/fig2012/papers/ts06g/TS06G\\_kilinc\\_baybura\\_5563.pdf](https://www.fig.net/resources/proceedings/fig_proceedings/fig2012/papers/ts06g/TS06G_kilinc_baybura_5563.pdf) [Accessed 14<sup>th</sup> March 2017]

Larson Davis (n.d) Human Vibration Meter for Worker Safety & Product Testing. United States of America, Larson Davis A PCB Piezotronics Div. Available from: [http://www.larsondavis.com/contentstore/mktgcontent/LD\\_Brochures/LD\\_HVM200\\_Lowres.pdf](http://www.larsondavis.com/contentstore/mktgcontent/LD_Brochures/LD_HVM200_Lowres.pdf) [Accessed 6th June 2017]

Miller, J.S., & Bellinger, W.Y. (2003) *Distress Identification Manual for the Long-Term Pavement Performance Program*. FHWA-RD-03-031, Federal Highway Administration, Washington, USA. Available from: <http://ccar.colorado.edu/dot/files/FHWA-RD-03-031.pdf> [Accessed 24th May 2017]

Murakami, E. & Wagner, D. P. (1999) Can using global positioning system (GPS) improve trip reporting? *Transportation Research Part C: Emerging Technologies*. 7 (2–3), 149-165.

Sayers, M.W., Gillespie, T.D., Paterson, W.D. (1986) *Guidelines for the Conduct and Calibration of Road Roughness Measurements*. Technical Paper 46. World Bank, Washington DC. Available from: <https://deepblue.lib.umich.edu/bitstream/handle/2027.42/3133/72764.pdf?sequence=2>. [Accessed 22th May 2017]

Schot, S.H. (1978) The Time Rate of Change of Acceleration, *American Journal of Physics*, 46 (11), 1090-1094.

Strauss, J., Miranda-Moreno, L. F. & Morency, P. (2015) Mapping cyclist activity and injury risk in a network combining smartphone GPS data and bicycle counts. *Accident Analysis & Prevention*. 83 132-142.

Strava Inc. (2017) *Strava*. [Mobile App] [Accessed 24th May 2017]

Vieyra Software (2017) *Physics Toolbox*. Available from <https://www.vieyrasoftware.net/> [Accessed 6<sup>th</sup> June 2017]

Vij, A. & Shankari, K. (2015) When is big data big enough? Implications of using GPS-based surveys for travel demand analysis. *Transportation Research Part C: Emerging Technologies*. 56 446-462.

Wong, J.Y. (2001) *Theory of Ground Vehicles*, John Wiley & Sons, Chap. 7, p.431.

World Health Organization. (2013), *World report on road traffic injury prevention*. Available from: [http://www.who.int/violence\\_injury\\_prevention/publications/road\\_traffic/world\\_report/en/](http://www.who.int/violence_injury_prevention/publications/road_traffic/world_report/en/) [Accessed 13<sup>th</sup> March 2017]

Yu, X. (2011). *Pavement surface distress detection and evaluation using image processing technology*. Masters Thesis, University of Toledo.

Zito, R., D'Este, G. & Taylor, M. A. P. (1995) *Global positioning systems in the time domain: How useful a tool for intelligent vehicle-highway systems?* Available from: <http://www.sciencedirect.com/science/article/pii/0968090X95000065>.

## Appendix A

### Accelerometer Code

```
import React, { Component } from 'react';
import {
  AppRegistry,
  StyleSheet,
  Text,
  View,
  DeviceEventEmitter,
  Dimensions
} from 'react-native';
import Echarts from 'native-echarts';
import { SensorManager } from 'NativeModules';
import UploadManager from './UploadManager'
const {height, width} = Dimensions.get('window');

const uploadDatas=[];
let testing = 1;
let myDatas={
  time:[],
  AccelerometerX:[],
  AccelerometerY:[],
  AccelerometerZ:[],
};
//Base values for accelerometer.
let baseValue={
  x: 2.235170057929281e-7,
  y: 9.776309967041016,
  z: 0.8123480081558228,
};
let extremeValues={
  time:[],
  AccelerometerX:[],
  AccelerometerY:[],
  AccelerometerZ:[],
};
let myProcessedData={
  time:[],
  AccelerometerX:[],
  AccelerometerY:[],
  AccelerometerZ:[],
};
const myRenderData={
  time:[],
  AccelerometerX:[],
```

```

AccelerometerY:[],
AccelerometerZ:[],
};

// const myDatas=[];
export default class AccelerometerChart extends Component {
constructor(props){
super(props);
this.state={
option: {},
}
}

componentWillMount(){
//get value from sensor
SensorManager.startAccelerometer(100);
DeviceEventEmitter.addListener('Accelerometer', (data) => {
//style1
// let myNow = new Date();
// let uploadData={
// 'timeStamp': myNow,
// 'accelerometerX': data.x,
// 'accelerometerY': data.y,
// 'accelerometerZ': data.z,
// };
// uploadDatas.push(uploadData);

//style2
let now=new Date();

if(data.x != baseValue.x || data.y != baseValue.y || data.z != baseValue.z) {
myDatas.time.push(now);
myDatas.AccelerometerX.push(data.x);
myDatas.AccelerometerY.push(data.y);
myDatas.AccelerometerZ.push(data.z);
}

myRenderData.time.push(now);
myRenderData.AccelerometerX.push(data.x);
myRenderData.AccelerometerY.push(data.y);
myRenderData.AccelerometerZ.push(data.z);

if(myRenderData.time.length>12){
myRenderData.time.shift();
myRenderData.AccelerometerX.shift();
myRenderData.AccelerometerY.shift();
myRenderData.AccelerometerZ.shift();
}
}

```

```

this.setState({
  x:data.x,
  y:data.y,
  z:data.z,
})
});
//render on screen
this.timer=setInterval(
()=>{
  this.state.option=this.getOption(myRenderData);
  this.setState({
    option:this.state.option
  })
},5000
);

this.dataProcessTimer=setInterval(
()=>{

  var meanDataX = averageData(myDatas.AccelerometerX);
  var meanDataY = averageData(myDatas.AccelerometerY);
  var meanDataZ = averageData(myDatas.AccelerometerZ);

  var outlierX = findOutlier(myDatas.AccelerometerX,meanDataX);
  var outlierY = findOutlier(myDatas.AccelerometerY,meanDataY);
  var outlierZ = findOutlier(myDatas.AccelerometerZ,meanDataZ);

  if(outlierX != null || outlierY != null || outlierZ != null){
    extremeValues={
      time: myDatas.time[myDatas.time.length - 1],
      AccelerometerX: outlierX,
      AccelerometerY: outlierY,
      AccelerometerZ: outlierZ,
    };
  }
}

myProcessedData={
  AccelerometerX: meanDataX,
  AccelerometerY: meanDataY,
  AccelerometerZ: meanDataZ,
};

myDatas={
  time:[],
  AccelerometerX:[],
  AccelerometerY:[],
}

```

```

        AccelerometerZ:[],
      }
    },2000
  )

// this.uploadTimer=setInterval(
// ()=>{
//   this.uploadData(myDatas);
//   myDatas={
//     time:[],
//     AccelerometerX:[],
//     AccelerometerY:[],
//     AccelerometerZ:[],
//   };
//   },10000
// )
}

componentDidMount() {
}
componentWillUnmount() {
  SensorManager.stopAccelerometer();
  this.timer && clearInterval(this.timer);
  this.uploadTimer && clearInterval(this.uploadTimer);
}
// uploadData(jsonData){
//   console.log("start uploading");
//   console.log(jsonData);
//   fetch("https://ridecomfort-5e3bf.firebaseio.com/data/.json",{
//     method:'POST',
//     headers:{
//       'Content-Type':'application/json',
//     },
//     body: JSON.stringify(jsonData),
//   })
//   .then((response)=> response.json())
//   .then((json)=> {
//     console.log(json);
//   })
//   .catch((error)=> {
//     console.log(error);
//   });
// }

```

```
getOption(data){
  let time=data.time.map(
    (t)=>{
      return t.getSeconds().toString()+'.'+t.getMilliseconds().toString();
    });
  const option = {
    title: {
      text: 'Accelerometer'
    },
    // tooltip: {
    //   trigger: 'axis'
    // },
    legend: {
      right:'5%',
      data:['X','Y','Z']
    },
    grid: {
      top: '10%',
      left: '3%',
      right: '4%',
      bottom: '3%',
      containLabel: true
    },
    // toolbox: {
    //   feature: {
    //     saveAsImage: {}
    //   }
    // },
    xAxis: {
      type: 'category',
      boundaryGap: false,
      data: time
    },
    yAxis: {
      type: 'value'
    },
    series: [
      {
        name:'X',
        type:'line',
        data:data.AccelerometerX,
        animation: false,
      },
      {
        name:'Y',
        type:'line',
        data:data.AccelerometerY,
        animation: false,
      },
      {
        name:'Z',
        type:'line',
        data:data.AccelerometerZ,
        animation: false,
      }
    ]
  };
  return option;
}
```

```

        type:'line',
        data:data.AccelerometerY,
        animation: false,
    },
    {
        name:'Z',
        type:'line',
        data:data.AccelerometerZ,
        animation: false,
    }
]
};

return option;
}

render() {
// return (
//   <View>
//     <View style={{flexDirection:'row',justifyContent:'space-around'}}>
//       <Text>
//         x:{Math.round(this.state.x*1000)/1000}
//       </Text>
//       <Text>
//         y:{Math.round(this.state.y*1000)/1000}
//       </Text>
//       <Text>
//         z:{Math.round(this.state.z*1000)/1000}
//       </Text>
//     </View>
//     <Echarts refs='lineChart' option={this.state.option} width={width} height={420} />
//   </View>
// );
if(extremeValues.AccelerometerX.length > 0){
  return(
<UploadManager
  Accelerometer_Outlier_Data = {extremeValues}
  Accelerometer_Average_Data = {myProcessedData}
/>
);
} else if(testing == 4) {
  return(
<UploadManager
  Accelerometer_Average_Data = {myDatas}
/>
);
} else {
  return (null)
}

```

```

        }
        extremeValues = {
            time:[],
            AccelerometerX:[],
            AccelerometerY:[],
            AccelerometerZ:[],
        };
        myProcessedData = {
            time:[],
            AccelerometerX:[],
            AccelerometerY:[],
            AccelerometerZ:[],
        };
    }
}

function averageData(data){
    console.log("Averaging Data");
    let averagedData = 0;
    for(var i = 0; i<data.length; i++) {
        averagedData += data[i];
    }
    averagedData = averagedData/data.length;
    console.log(averagedData);
    return averagedData;
}

function findOutlier(data,average){
    console.log("Finding an outlier");
    var outlier = []
    var position_q3 = Math.ceil(data.length * 3 / 4);
    var position_q1 = Math.ceil(data.length / 4);
    var iqr = (data[position_q3] - data[position_q1]);
    var cutoff1 = average + (1.5 * iqr);
    var cutoff2 = average - (1.5 * iqr);

    for(var i = 0; i<data.length; i++) {
        if(data[i] > cutoff1 || data[i] < cutoff2){
            outlier.push(data[i])
        }
    }

    if(outlier.length > 0){
        return outlier;
    } else {
        outlier = null;
    }
}

```

```
    return outlier;
}
}
```

## GPS Service code

```
/**  
 * app for ridecomfort  
 */  
  
import React, {  
  Component,  
} from 'react';  
import {  
  AppRegistry,  
  StyleSheet,  
  Text,  
  View,  
  ScrollView  
} from 'react-native';  
  
import UploadManager from './UploadManager'  
  
let myDatas={  
  timestamp:[],  
  longitude:[],  
  latitude:[],  
};  
  
export default class GeoLocation extends Component {  
  
  constructor() {  
    super();  
  
    this.state = {  
      latitude: null,  
      longitude: null,  
      timestamp: null,  
      error: null,  
    };  
  }  
  
  componentDidMount() {  
    this.watchId = navigator.geolocation.watchPosition(  
      (position) => {  
        let timestamp = new Date().getTime();  
        let latitude = position.coords.latitude;  
        let longitude = position.coords.longitude;  
        let error = position.coords.accuracy;  
  
        myDatas.timestamp.push(timestamp);  
        myDatas.longitude.push(longitude);  
        myDatas.latitude.push(latitude);  
        myDatas.error.push(error);  
      },  
      {  
        enableHighAccuracy: true,  
        timeout: 20000,  
        maximumAge: 10000,  
      },  
      {  
        enableHighAccuracy: true,  
      }  
    );  
  }  
  
  componentWillUnmount() {  
    navigator.geolocation.clearWatch(this.watchId);  
  }  
}
```

```
var t = Date(position.timestamp)
myDatas.longitude.push(position.coords.longitude);
myDatas.latitude.push(position.coords.latitude);
myDatas.timestamp.push(t);
this.setState({
  latitude: position.coords.latitude,
  longitude: position.coords.longitude,
  timestamp: t,
  error: null,
});
},
(error) => this.setState({error:error.message}),
{ enableHighAccuracy: true, timeout: 20000, maximumAge: 1000, distanceFilter: 1 },
);
}

componentWillUnmount() {
  navigator.geolocation.clearWatch(this.watchId);
}

render() {
  return (
    <UploadManager
      LocationData = {myDatas}
    />
  );
}
}
```

## Appendix B

### B.1 – Rural 1

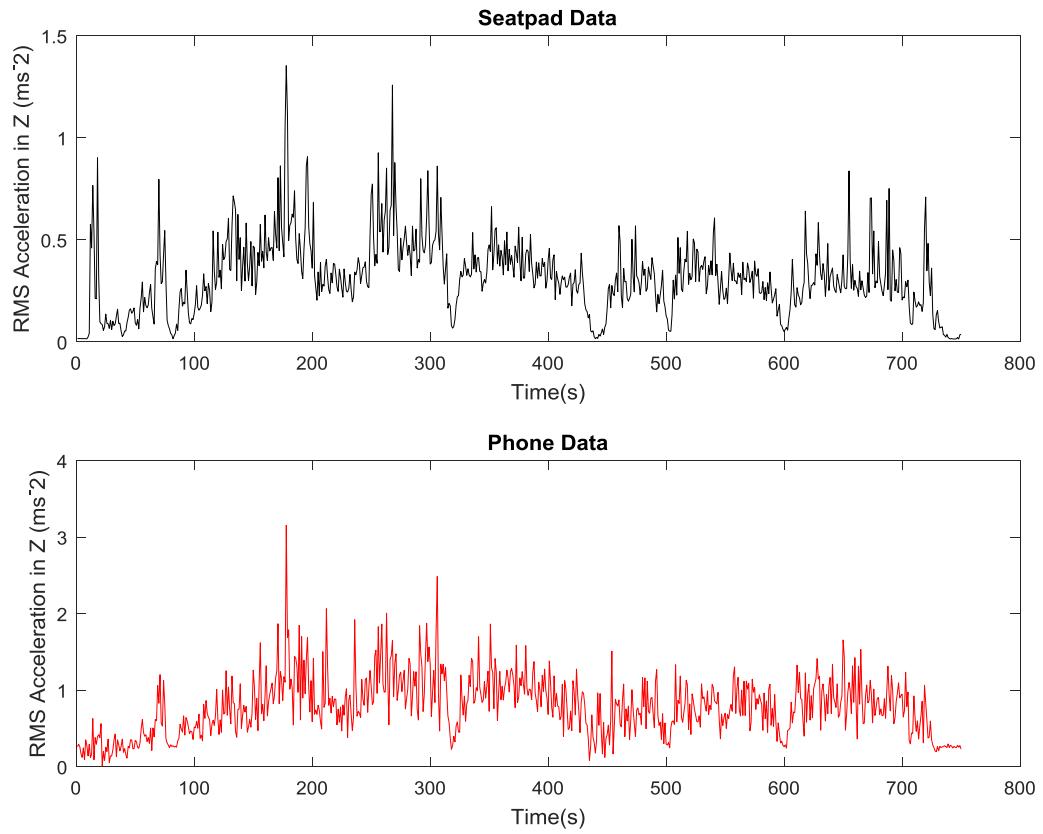


Figure B.1 – Time aligned RMS Accelerations in the Z axis obtained from the seatpad and the phone app

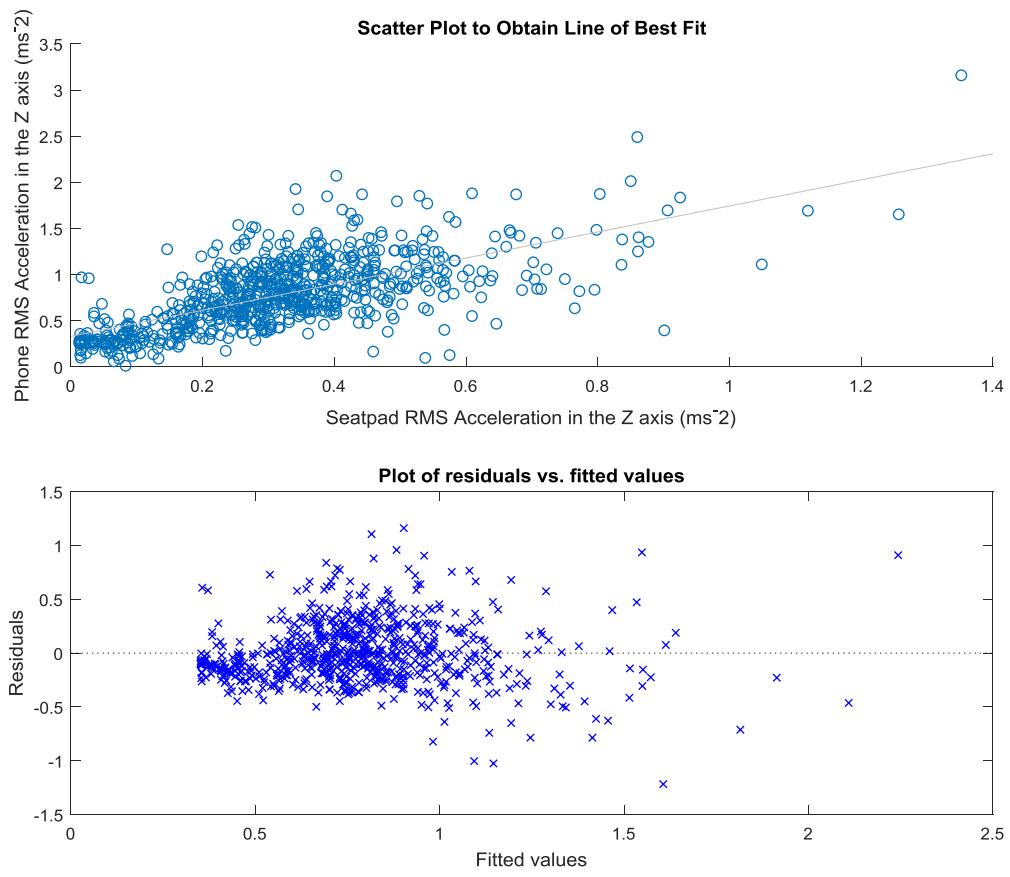


Figure B.2 – Line of best fit fitted to RMS Accelerations in the Z axis & Residual plot to show stochastic nature of error

```

Linear regression model:
y ~ 1 + x1

Estimated Coefficients:
              Estimate      SE      tStat    pValue
(Intercept) 0.33199  0.021139  15.705 3.0012e-48
x1          1.4125   0.057607  24.519 6.7821e-98

Number of observations: 750, Error degrees of freedom: 748
Root Mean Squared Error: 0.289
R-squared: 0.446, Adjusted R-Squared 0.445
F-statistic vs. constant model: 601, p-value = 6.78e-98

```

Figure B.3 – Linear regression model applied to RMS Accelerations in the Z axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

## B.2 – Urban 1

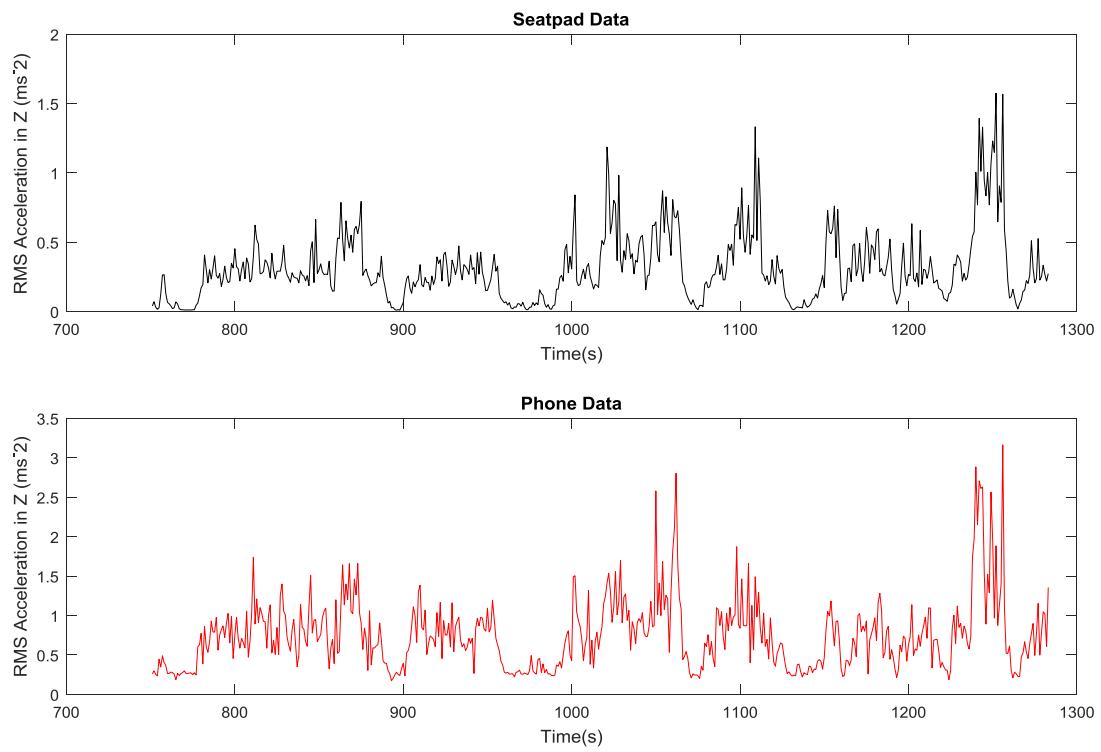


Figure B.4 – Time aligned RMS Accelerations in the Z axis obtained from the seatpad and the phone app

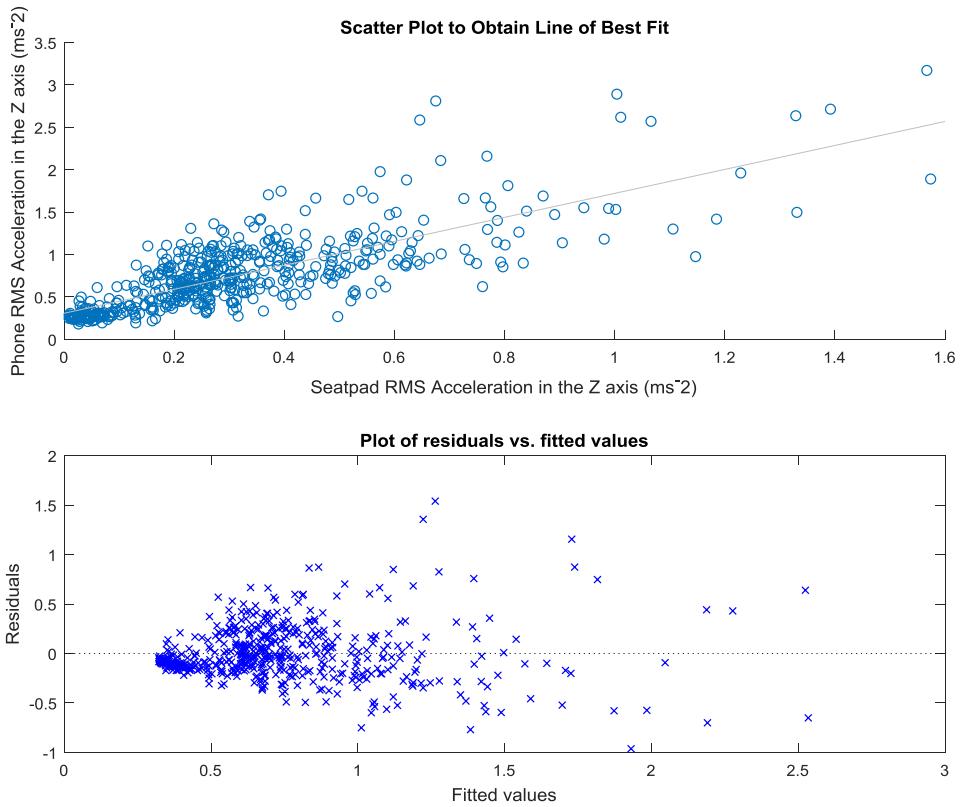


Figure B.5 – Line of best fit fitted to RMS Accelerations in the Z axis & Residual plot to show stochastic nature of error

```

Linear regression model:
y ~ 1 + x1

Estimated Coefficients:
              Estimate      SE      tStat    pValue
(Intercept) 0.30822  0.0197  15.646  1.183e-45
x1          1.4133   0.049449 28.581  1.7339e-109

```

```

Number of observations: 533, Error degrees of freedom: 531
Root Mean Squared Error: 0.282
R-squared: 0.606, Adjusted R-Squared 0.605
F-statistic vs. constant model: 817, p-value = 1.73e-109

```

Figure B.6 – Linear regression model applied to RMS Accelerations in the Z axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

### B.3 – Rural 2

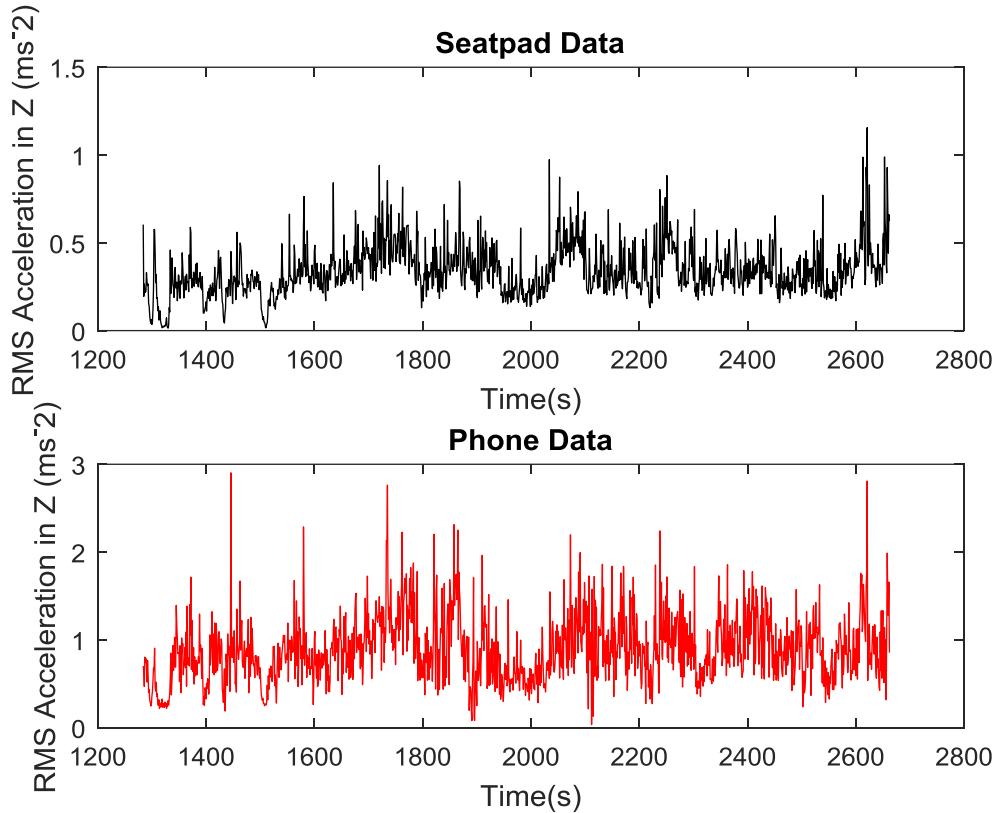


Figure B.7 – Time aligned RMS Accelerations in the Z axis obtained from the seatpad and the phone app

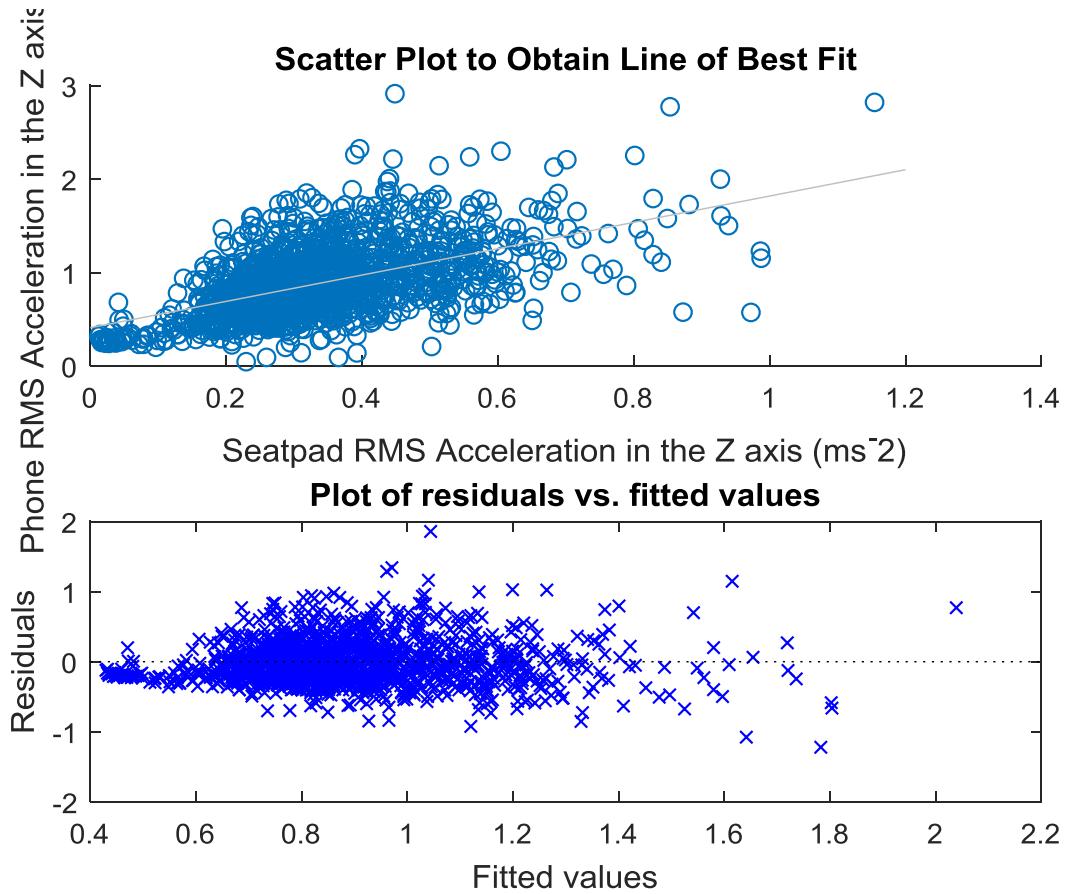


Figure B.8 – Line of best fit fitted to RMS Accelerations in the Z axis & Residual plot to show stochastic nature of error

```

Linear regression model:
y ~ 1 + x1

Estimated Coefficients:
              Estimate      SE   tStat    pValue
(Intercept) 0.40995 0.022067 18.578 6.3871e-69
x1          1.4091  0.059501 23.682 2.7439e-104

Number of observations: 1379, Error degrees of freedom: 1377
Root Mean Squared Error: 0.32
R-squared: 0.289, Adjusted R-Squared 0.289
F-statistic vs. constant model: 561, p-value = 2.74e-104

```

Figure B.9 – Linear regression model applied to RMS Accelerations in the Z axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

#### B.4 – Motorway 1

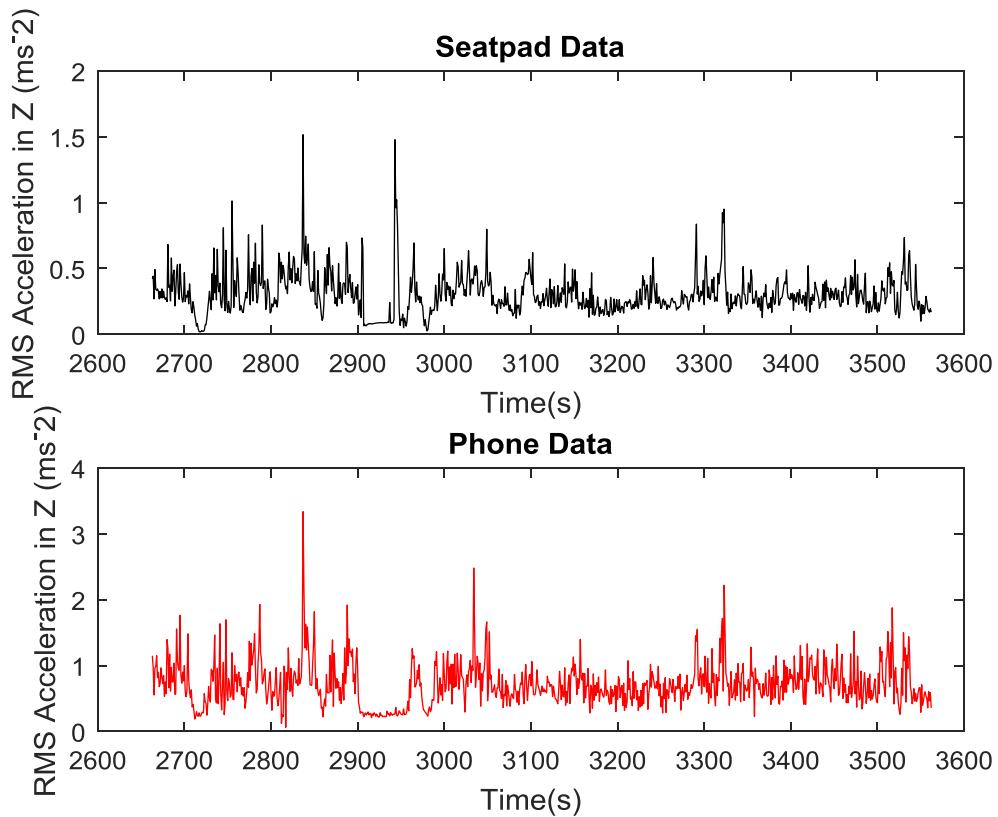


Figure B.10 – Time aligned RMS Accelerations in the Z axis obtained from the seatpad and the phone app

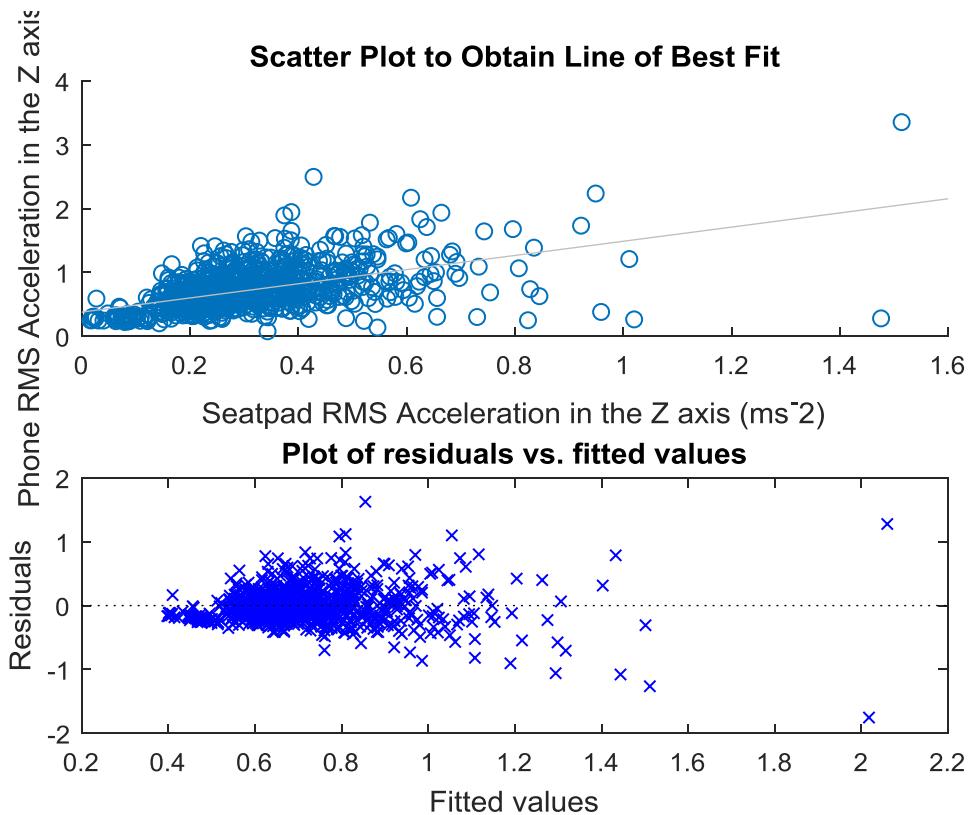


Figure B.11 – Line of best fit fitted to RMS Accelerations in the Z axis & Residual plot to show stochastic nature of error

Linear regression model:

$$y \sim 1 + x_1$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	0.37761	0.02069	18.251	1.5428e-63
x1	1.1094	0.060374	18.376	2.895e-64

Number of observations: 900, Error degrees of freedom: 898

Root Mean Squared Error: 0.282

R-squared: 0.273, Adjusted R-Squared 0.272

F-statistic vs. constant model: 338, p-value = 2.89e-64

Figure B.12 – Linear regression model applied to RMS Accelerations in the Z axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

### B.5 – Rural 3

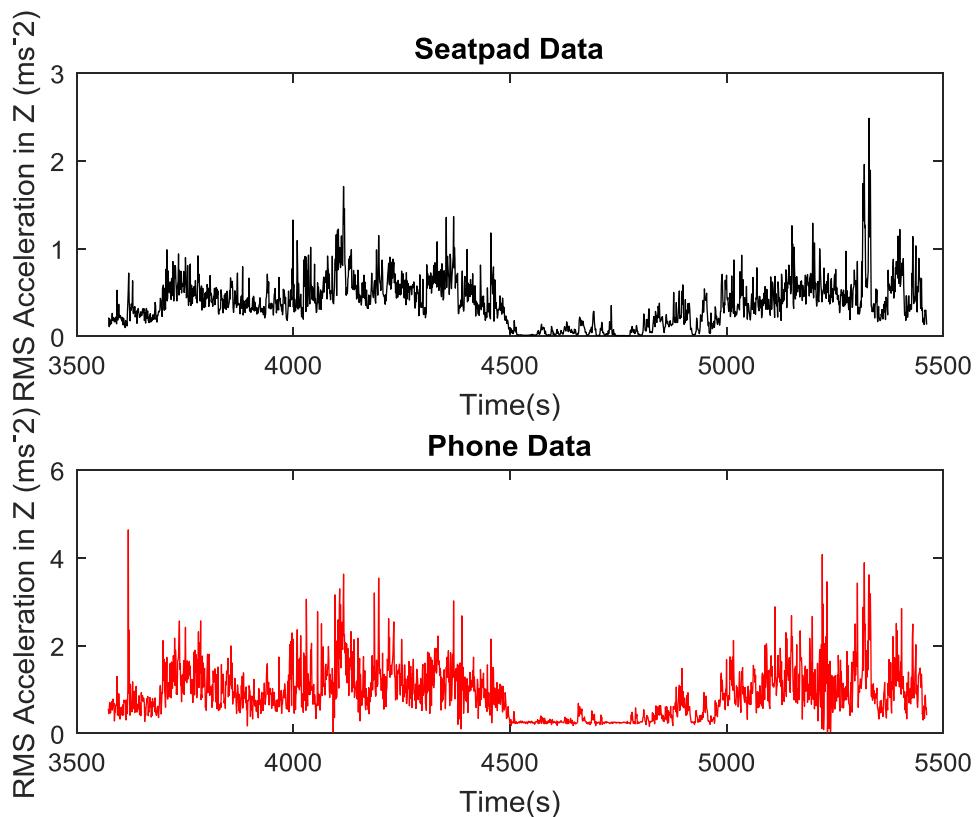


Figure B.13 – Time aligned RMS Accelerations in the Z axis obtained from the seatpad and the phone app

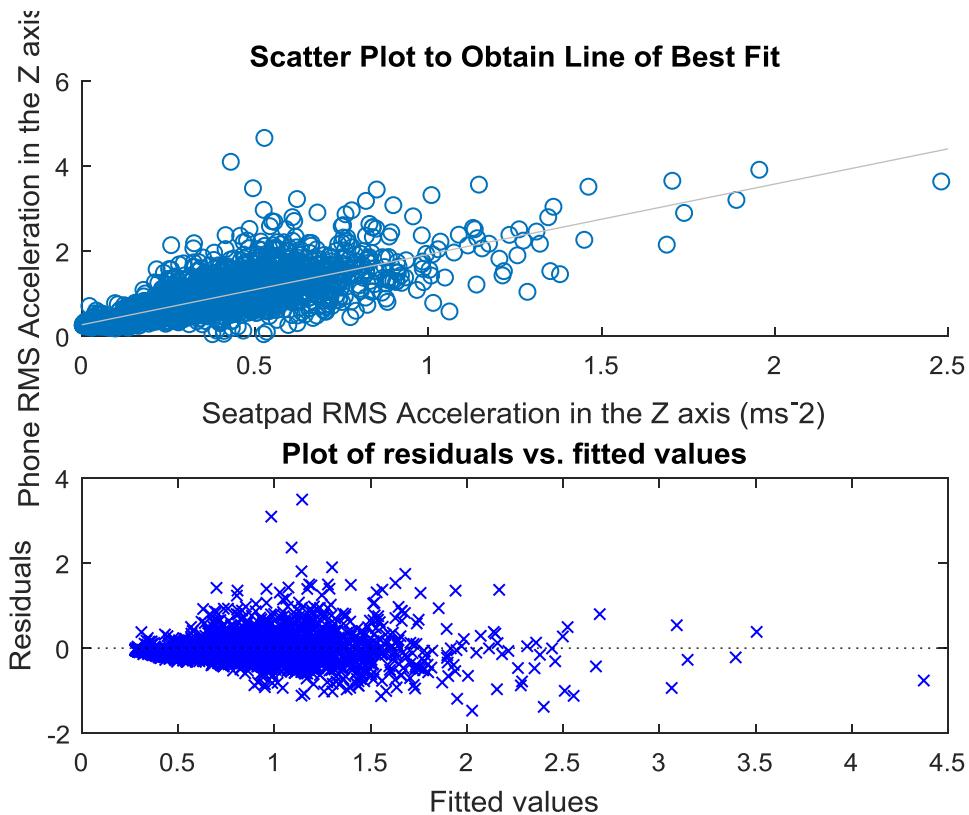


Figure B.14 – Line of best fit fitted to RMS Accelerations in the Z axis & Residual plot to show stochastic nature of error

Linear regression model:

$y \sim 1 + x1$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	0.26855	0.015465	17.366	8.7314e-63
x1	1.6533	0.032871	50.296	0

Number of observations: 1890, Error degrees of freedom: 1888  
Root Mean Squared Error: 0.383  
R-squared: 0.573, Adjusted R-Squared 0.572  
F-statistic vs. constant model: 2.53e+03, p-value = 0

Figure B.15 – Linear regression model applied to RMS Accelerations in the Z axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

### B.6 – Motorway 2

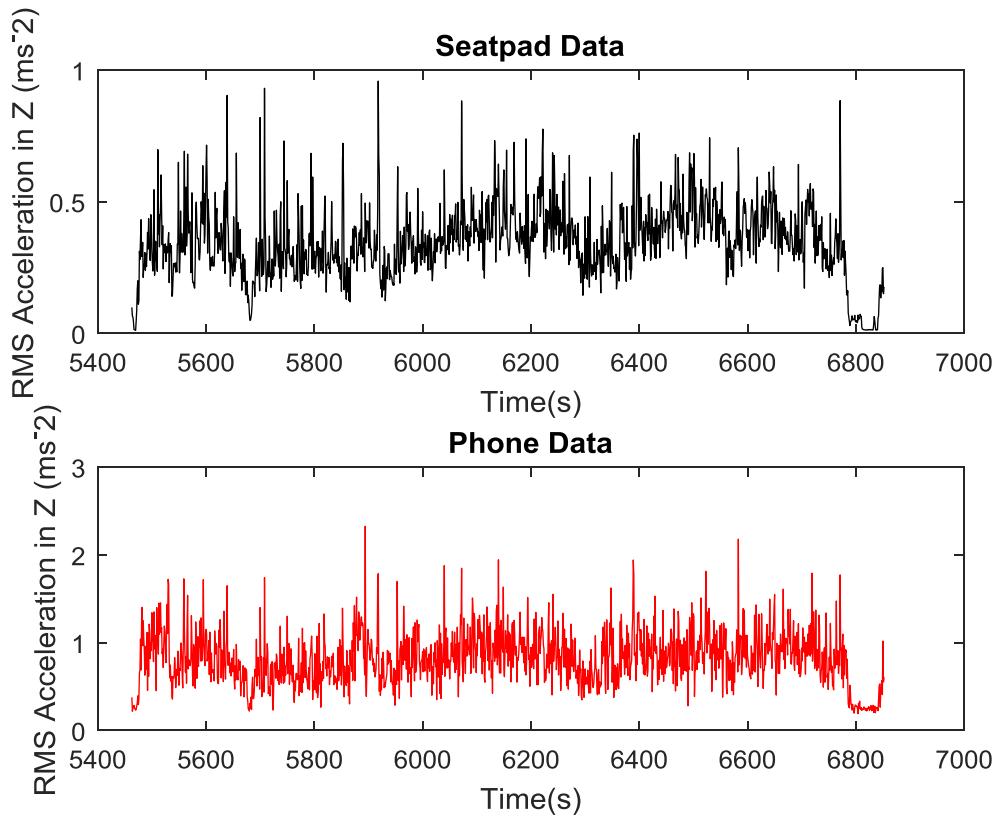


Figure B.16 – Time aligned RMS Accelerations in the Z axis obtained from the seatpad and the phone app

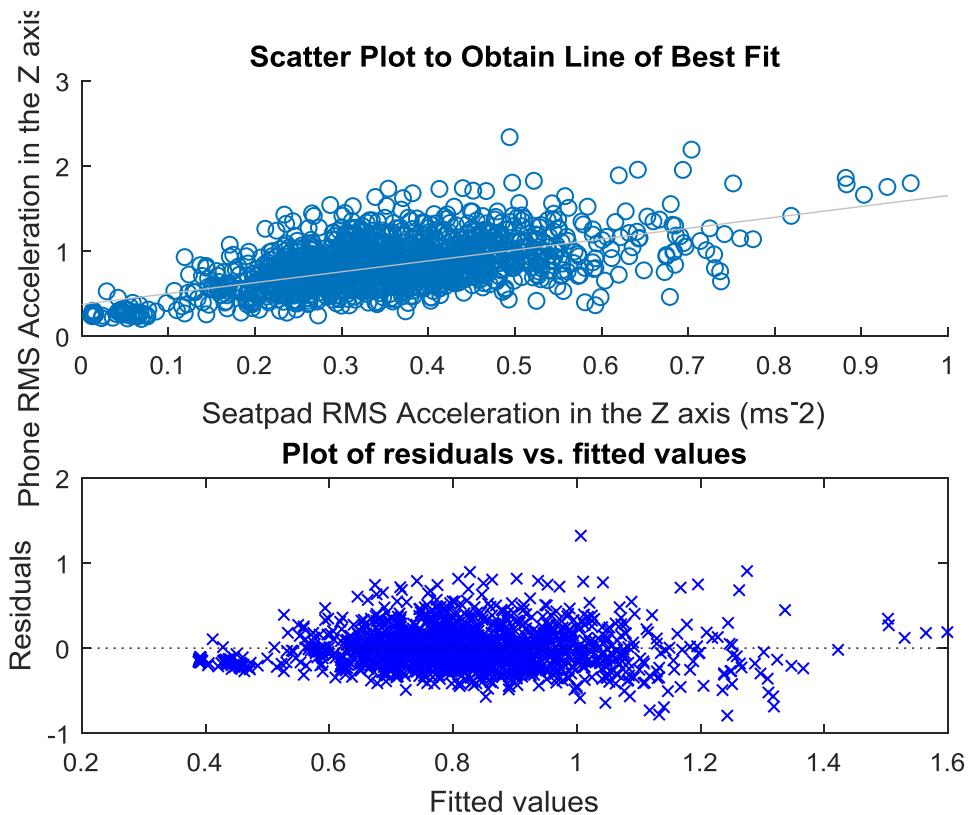


Figure B.17 – Line of best fit fitted to RMS Accelerations in the Z axis & Residual plot to show stochastic nature of error

Linear regression model:

$$y \sim 1 + x_1$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	0.37321	0.017755	21.02	2.1798e-85
x1	1.2793	0.047429	26.972	3.5063e-129

```
Number of observations: 1390, Error degrees of freedom: 1388
Root Mean Squared Error: 0.246
R-squared: 0.344, Adjusted R-Squared 0.343
F-statistic vs. constant model: 727, p-value = 3.51e-129
```

Figure B.18 – Linear regression model applied to RMS Accelerations in the Z axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

B.7 – Urban 2

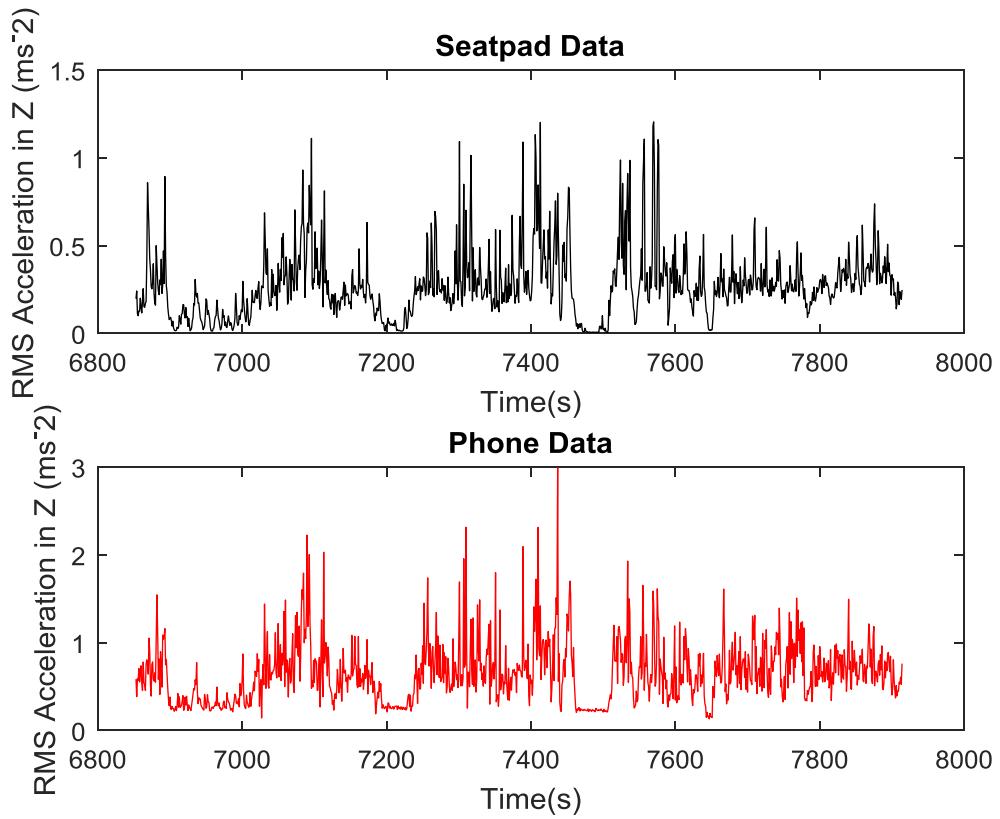


Figure B.19 – Time aligned RMS Accelerations in the Z axis obtained from the seatpad and the phone app

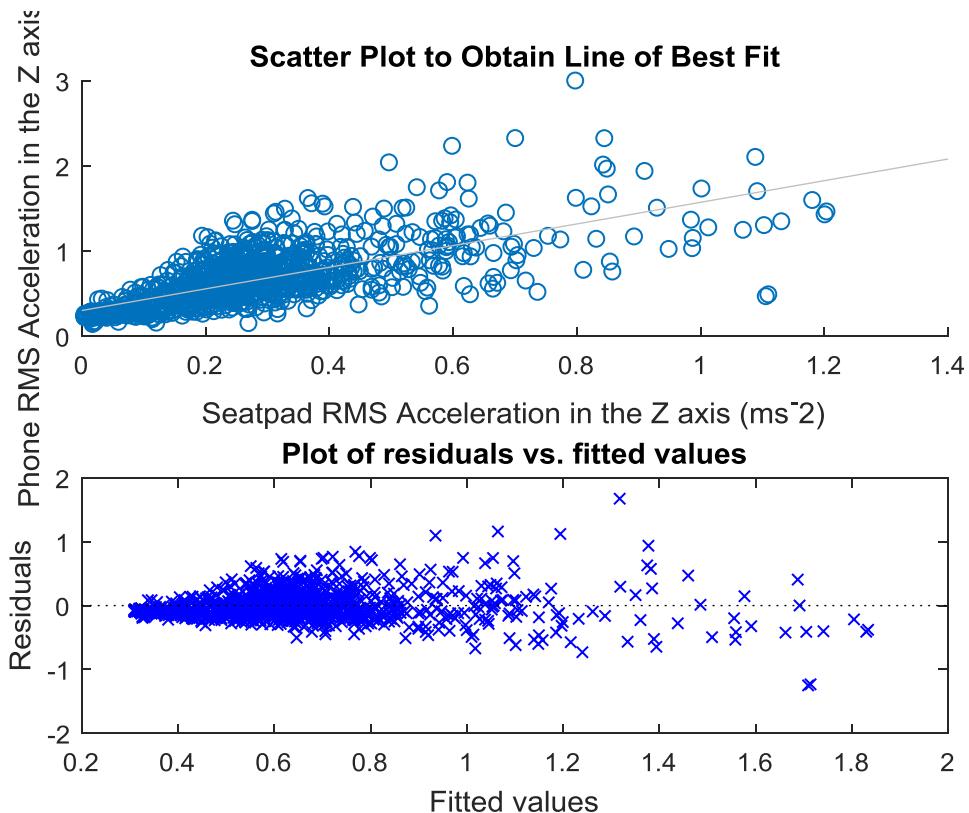


Figure B.20 – Line of best fit fitted to RMS Accelerations in the Z axis & Residual plot to show stochastic nature of error

Linear regression model:

$y \sim 1 + x1$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	0.30199	0.012848	23.504	1.1678e-98
x1	1.2711	0.038379	33.121	1.0202e-165

Number of observations: 1062, Error degrees of freedom: 1060

Root Mean Squared Error: 0.242

R-squared: 0.509, Adjusted R-Squared 0.508

F-statistic vs. constant model: 1.1e+03, p-value = 1.02e-165

Figure B.21 – Linear regression model applied to RMS Accelerations in the Z axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

B.8 – Rural 4

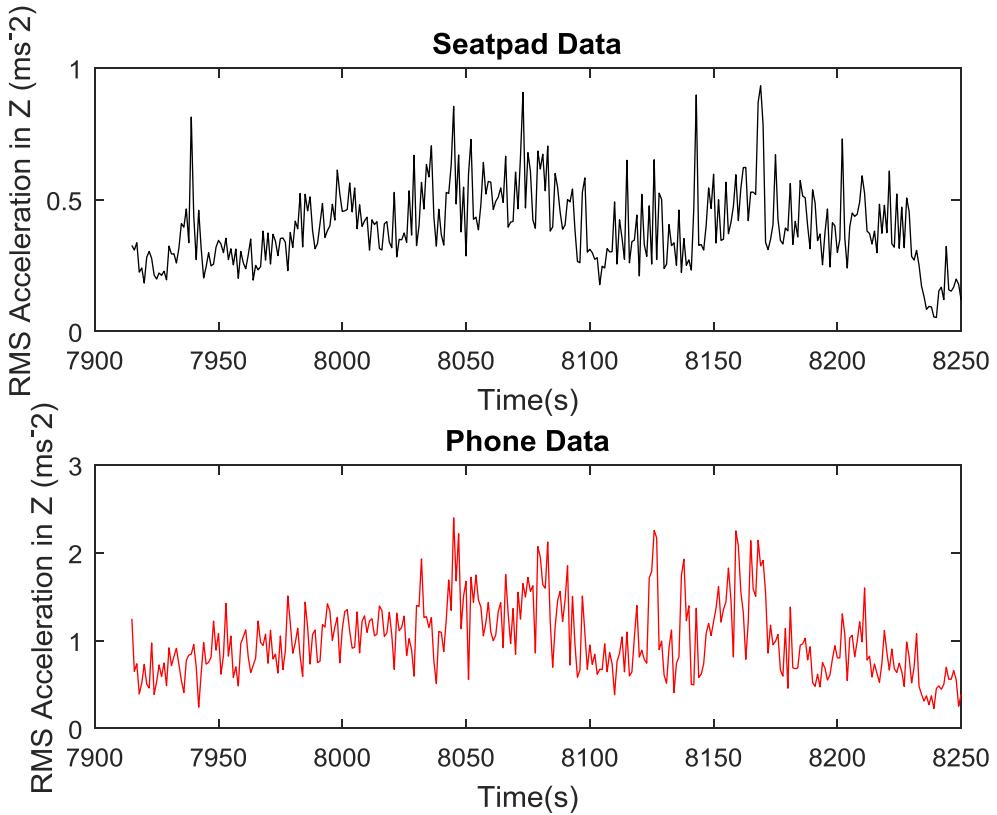


Figure B.22 – Time aligned RMS Accelerations in the Z axis obtained from the seatpad and the phone app

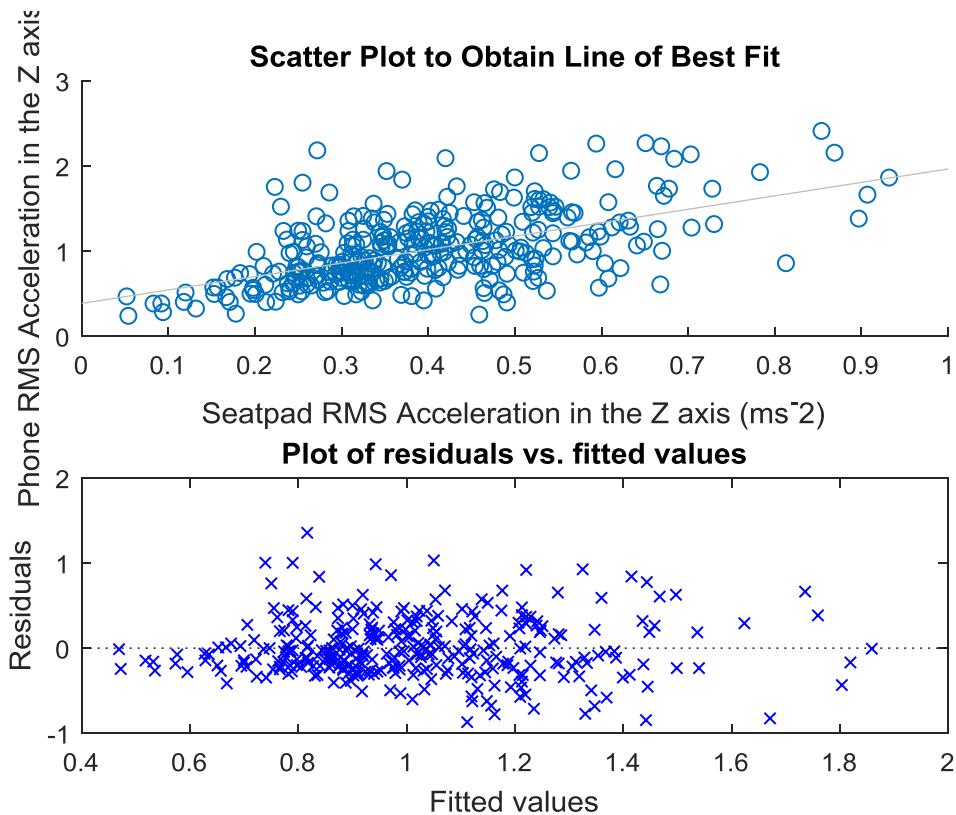


Figure B.23 – Line of best fit fitted to RMS Accelerations in the Z axis & Residual plot to show stochastic nature of error

Linear regression model:

$$y \sim 1 + x_1$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	0.38514	0.054696	7.0414	1.0897e-11
x1	1.5798	0.12864	12.281	7.3214e-29

```
Number of observations: 336, Error degrees of freedom: 334
Root Mean Squared Error: 0.35
R-squared: 0.311, Adjusted R-Squared 0.309
F-statistic vs. constant model: 151, p-value = 7.32e-29
```

Figure B.24 – Linear regression model applied to RMS Accelerations in the Z axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

## Appendix C

### C.1 – Rural 1

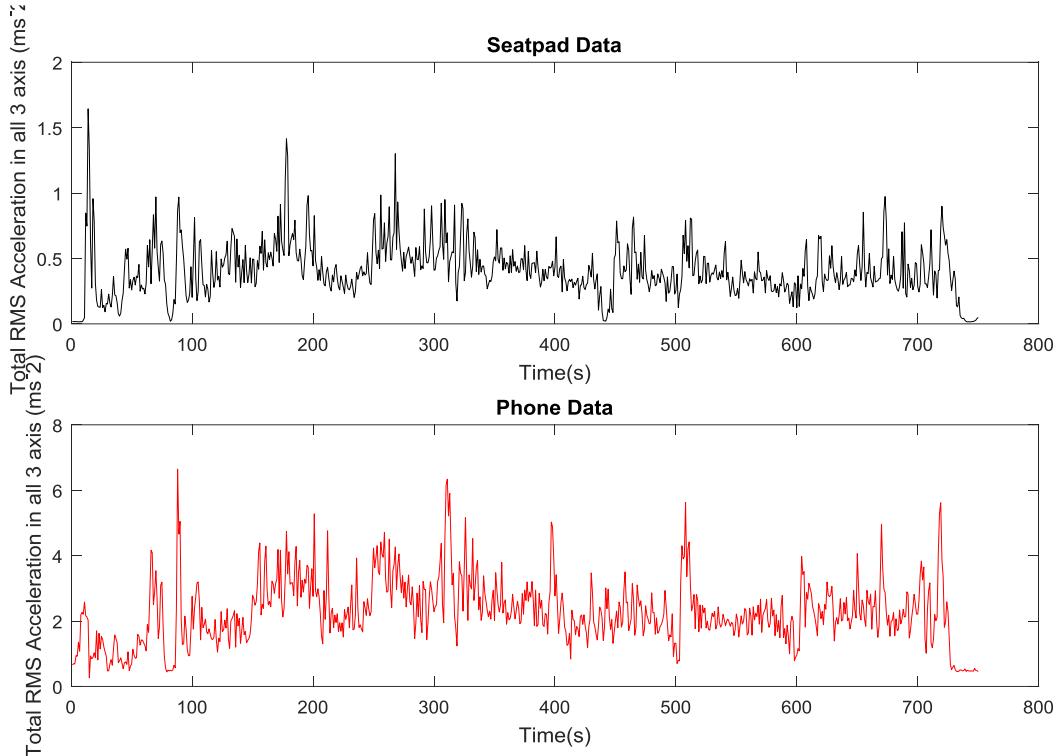


Figure C.1 – Time aligned Total RMS Accelerations in all 3 axis obtained from the seatpad and the phone app

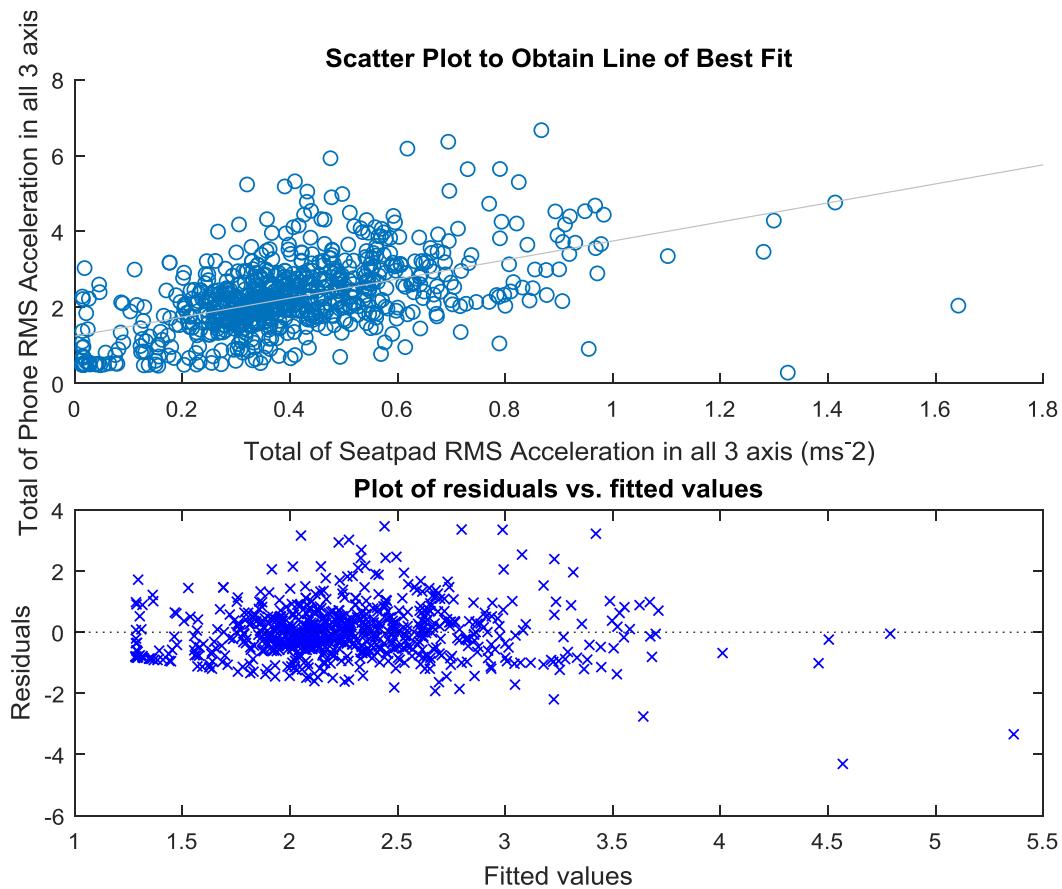


Figure C.2 – Line of best fit fitted to total RMS Accelerations in all 3 axis & Residual plot to show stochastic nature of error

```

Linear regression model:
y ~ 1 + x1

Estimated Coefficients:
              Estimate      SE   tStat    pValue
(Intercept) 1.2447 0.069047 18.026 1.3338e-60
x1          2.5052 0.15015 16.685 2.273e-53

```

```

Number of observations: 750, Error degrees of freedom: 748
Root Mean Squared Error: 0.85
R-squared: 0.271, Adjusted R-Squared 0.27
F-statistic vs. constant model: 278, p-value = 2.27e-53

```

Figure C.3 – Linear regression model applied to total RMS Accelerations in all 3 axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

C.2 – Urban 1

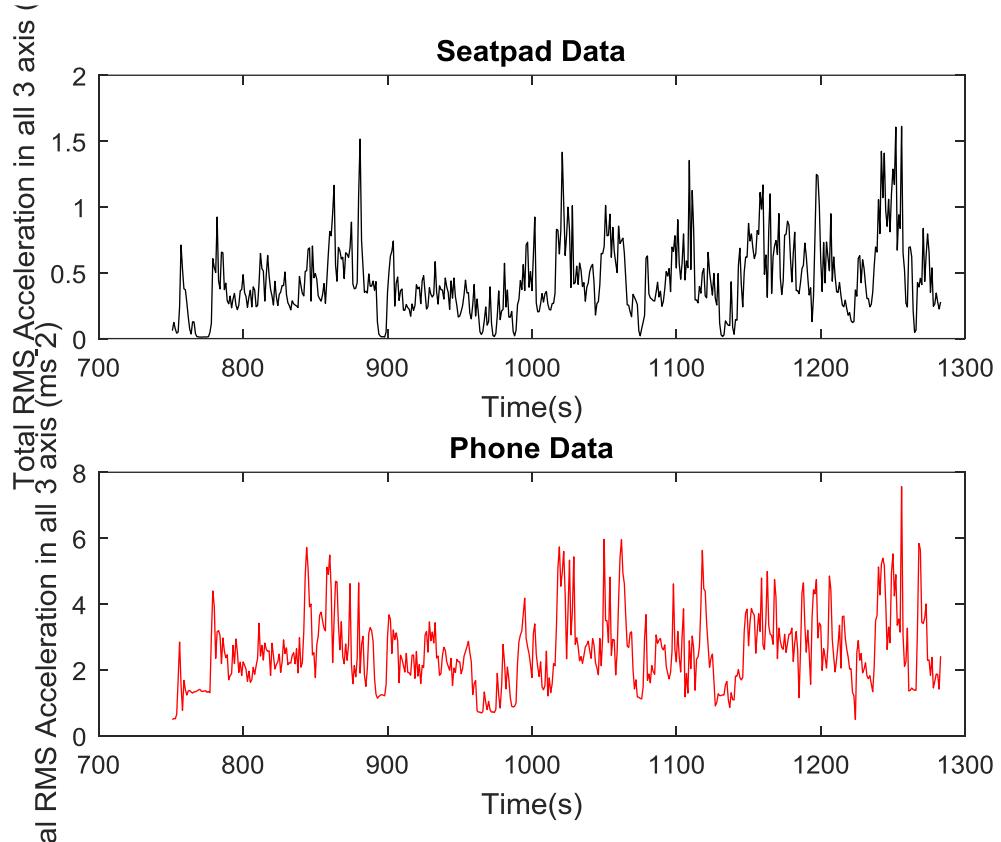


Figure C.4 – Time aligned Total RMS Accelerations in all 3 axis obtained from the seatpad and the phone app

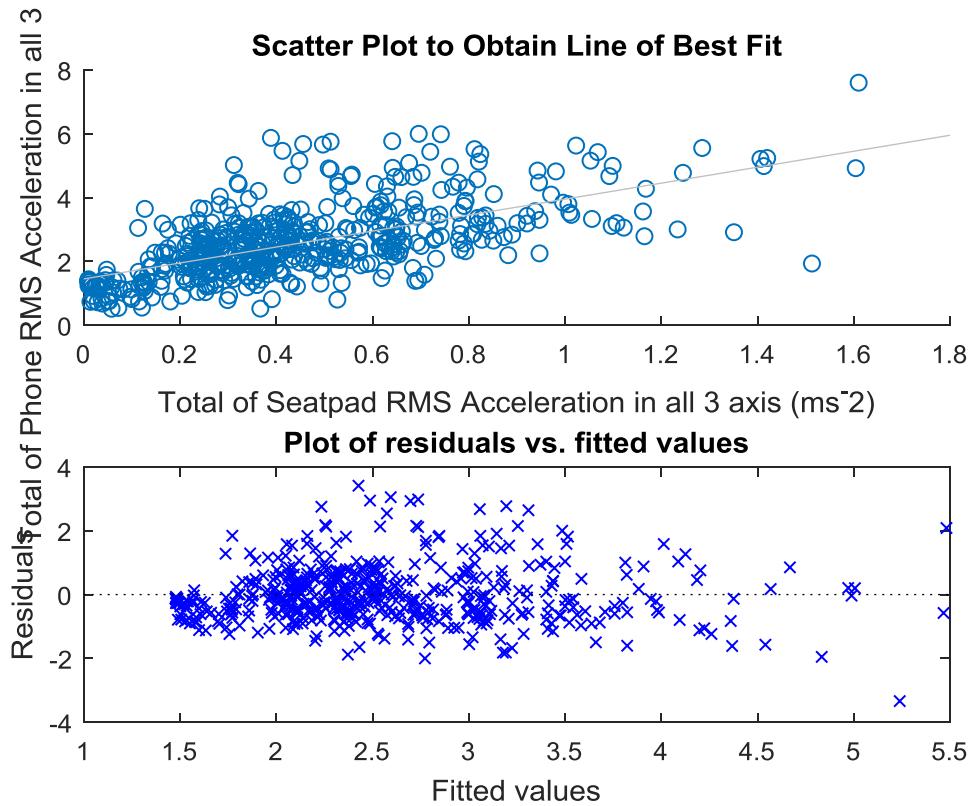


Figure C.5 – Line of best fit fitted to total RMS Accelerations in all 3 axis & Residual plot to show stochastic nature of error

Linear regression model:

$$y \sim 1 + x_1$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	1.4491	0.071049	20.395	1.0299e-68
x1	2.5011	0.13515	18.506	2.2448e-59

```
Number of observations: 533, Error degrees of freedom: 531
Root Mean Squared Error: 0.877
R-squared: 0.392, Adjusted R-Squared 0.391
F-statistic vs. constant model: 342, p-value = 2.24e-59
```

Figure C.6 – Linear regression model applied to total RMS Accelerations in all 3 axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

C.3 – Rural 2

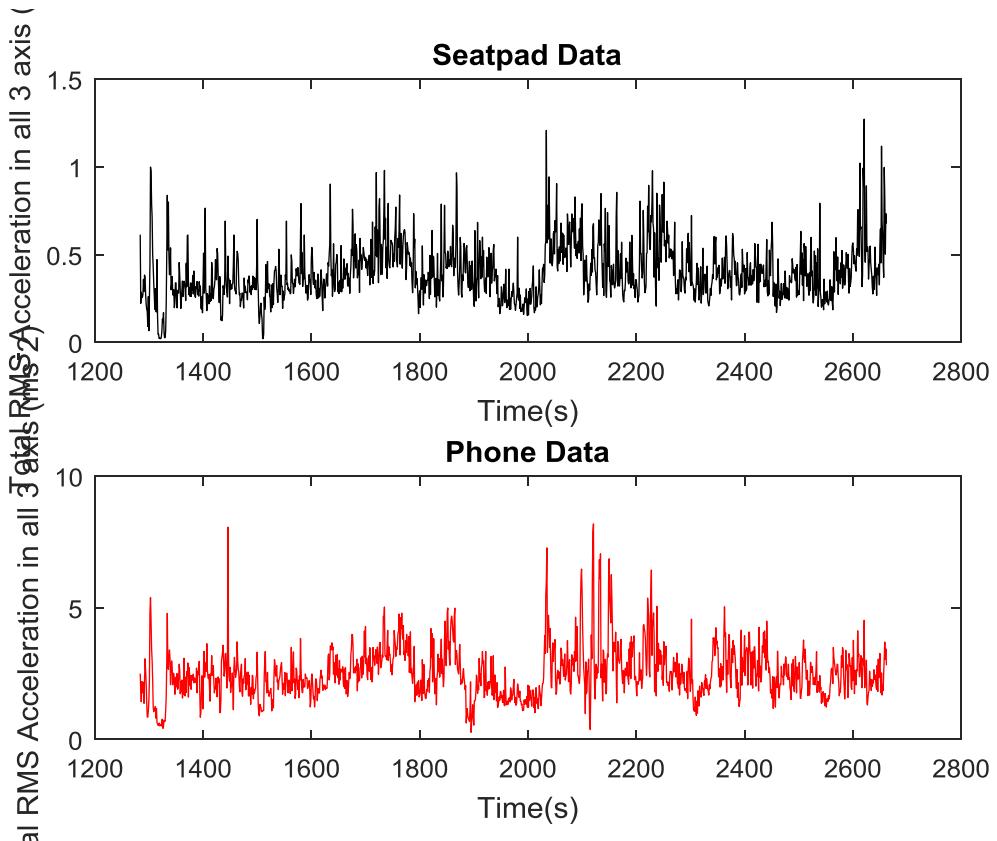


Figure C.7 – Time aligned Total RMS Accelerations in all 3 axis obtained from the seatpad and the phone app

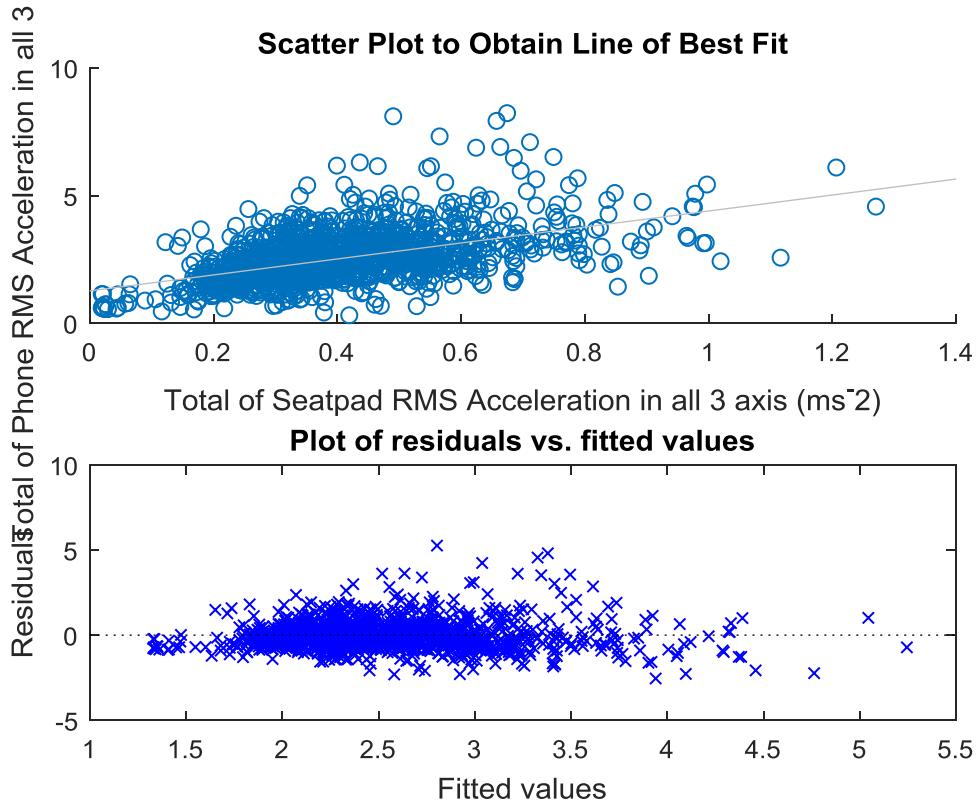


Figure C.8 – Line of best fit fitted to total RMS Accelerations in all 3 axis & Residual plot to show stochastic nature of error

Linear regression model:

$y \sim 1 + x1$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	1.2642	0.060569	20.872	2.7994e-84
x1	3.129	0.14005	22.342	1.3376e-94

Number of observations: 1379, Error degrees of freedom: 1377  
Root Mean Squared Error: 0.834  
R-squared: 0.266, Adjusted R-Squared 0.266  
F-statistic vs. constant model: 499, p-value = 1.34e-94

Figure C.9 – Linear regression model applied to total RMS Accelerations in all 3 axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

#### C.4 – Motorway 1

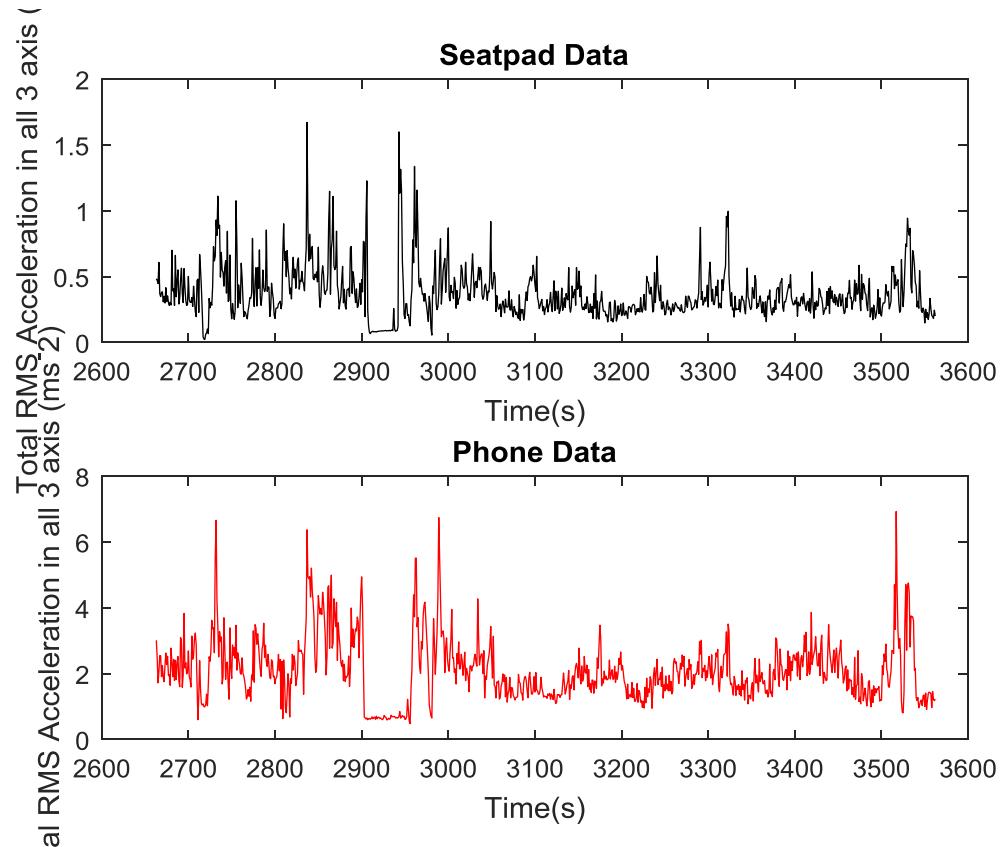


Figure C.10 – Time aligned Total RMS Accelerations in all 3 axis obtained from the seatpad and the phone app

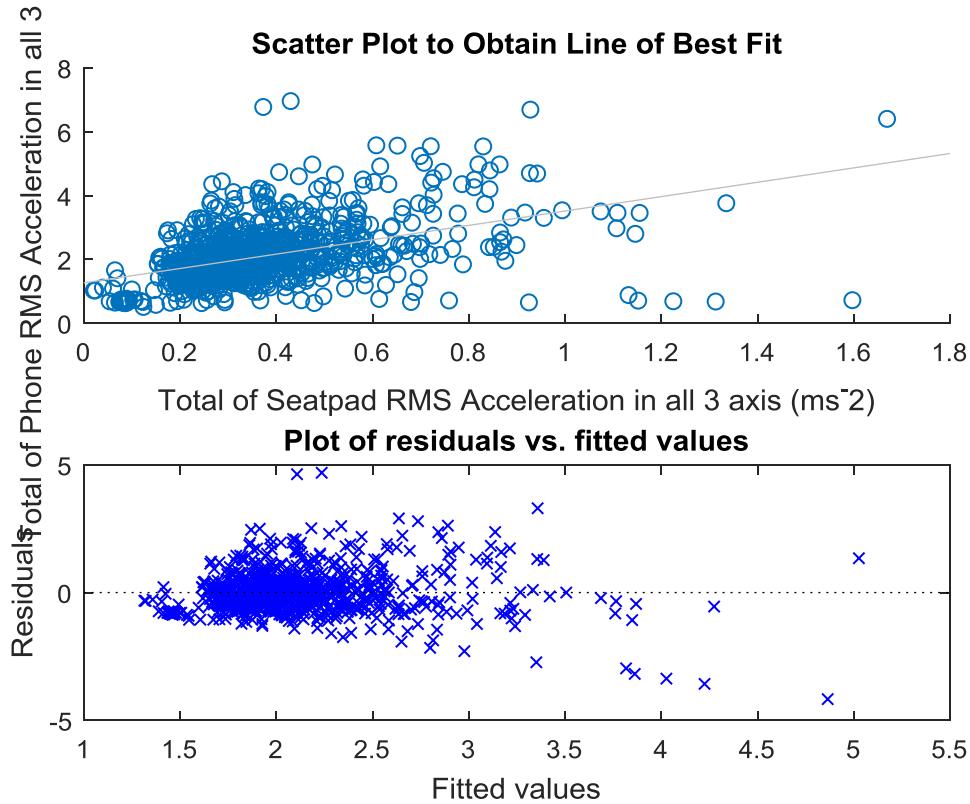


Figure C.11 – Line of best fit fitted to total RMS Accelerations in all 3 axis & Residual plot to show stochastic nature of error

Linear regression model:

$$y \sim 1 + x_1$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	1.2633	0.060032	21.044	3.1043e-80
x1	2.2519	0.14325	15.72	2.2538e-49

Number of observations: 900, Error degrees of freedom: 898

Root Mean Squared Error: 0.832

R-squared: 0.216, Adjusted R-Squared 0.215

F-statistic vs. constant model: 247, p-value = 2.25e-49

Figure C.12 – Linear regression model applied to total RMS Accelerations in all 3 axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

### C.5 – Rural 3

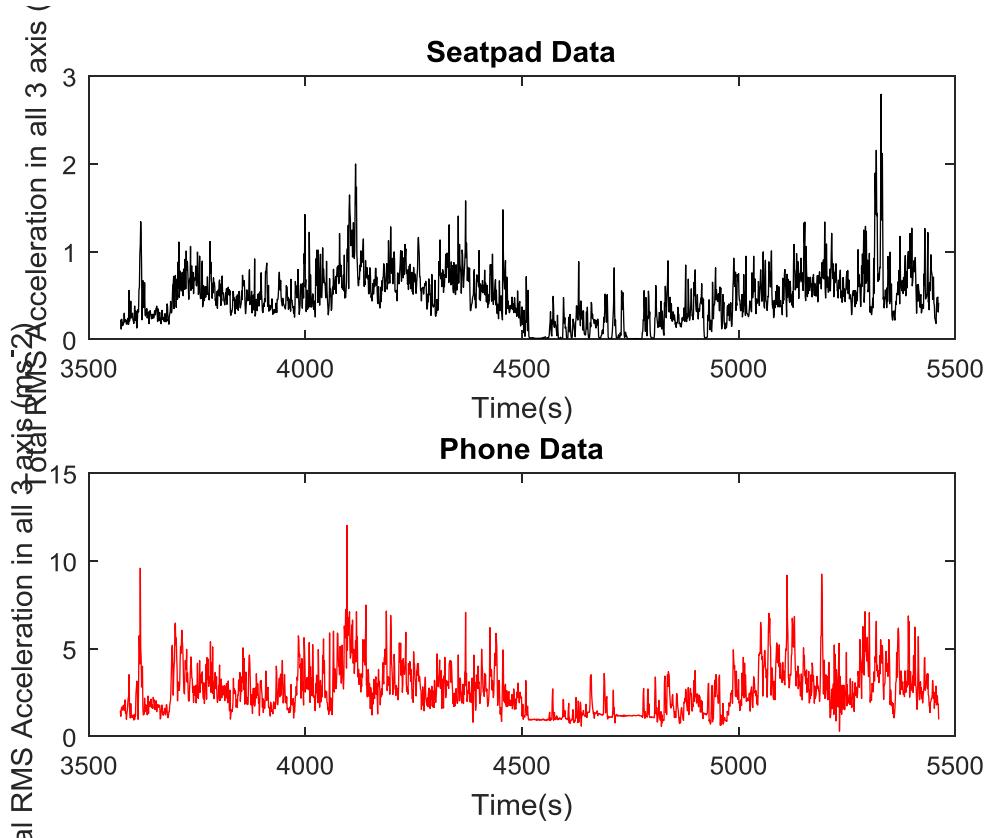


Figure C.13 – Time aligned Total RMS Accelerations in all 3 axis obtained from the seatpad and the phone app

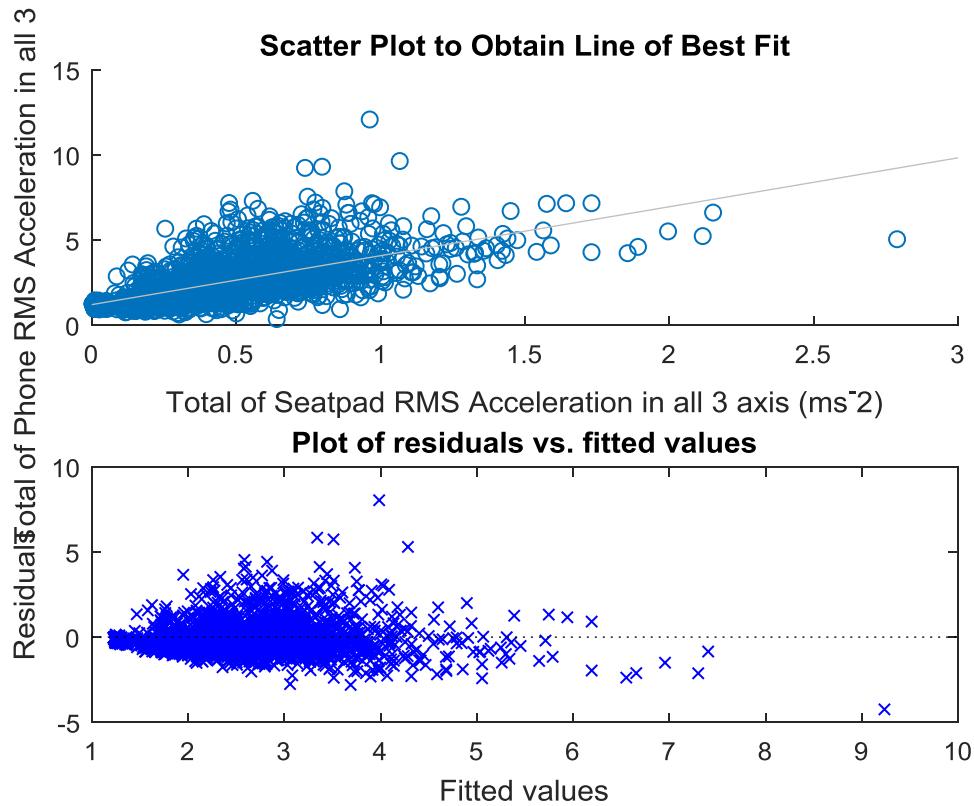


Figure C.14 – Line of best fit fitted to total RMS Accelerations in all 3 axis & Residual plot to show stochastic nature of error

Linear regression model:

$y \sim 1 + x_1$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	1.2091	0.045872	26.357	1.2517e-130
x1	2.8745	0.0796	36.112	1.4462e-217

Number of observations: 1890, Error degrees of freedom: 1888

Root Mean Squared Error: 1.04

R-squared: 0.409, Adjusted R-Squared 0.408

F-statistic vs. constant model: 1.3e+03, p-value = 1.45e-217

Figure C.15 – Linear regression model applied to total RMS Accelerations in all 3 axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

### C.6 – Motorway 2

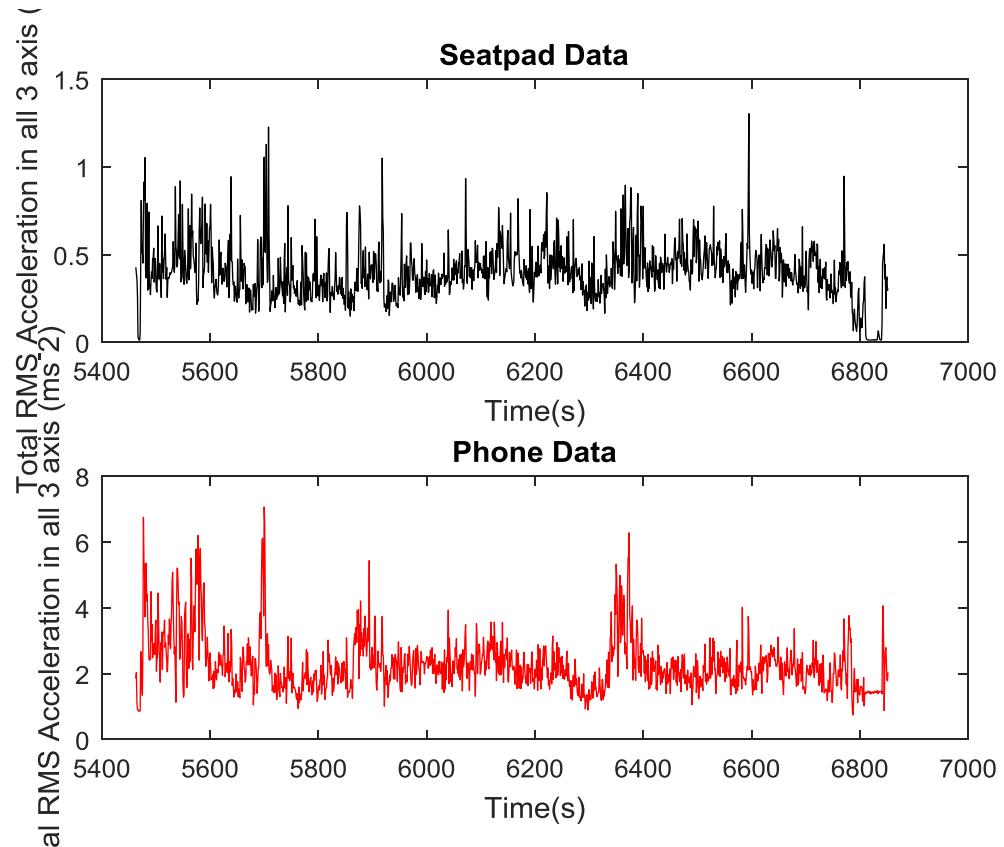


Figure C.16 – Time aligned Total RMS Accelerations in all 3 axis obtained from the seatpad and the phone app

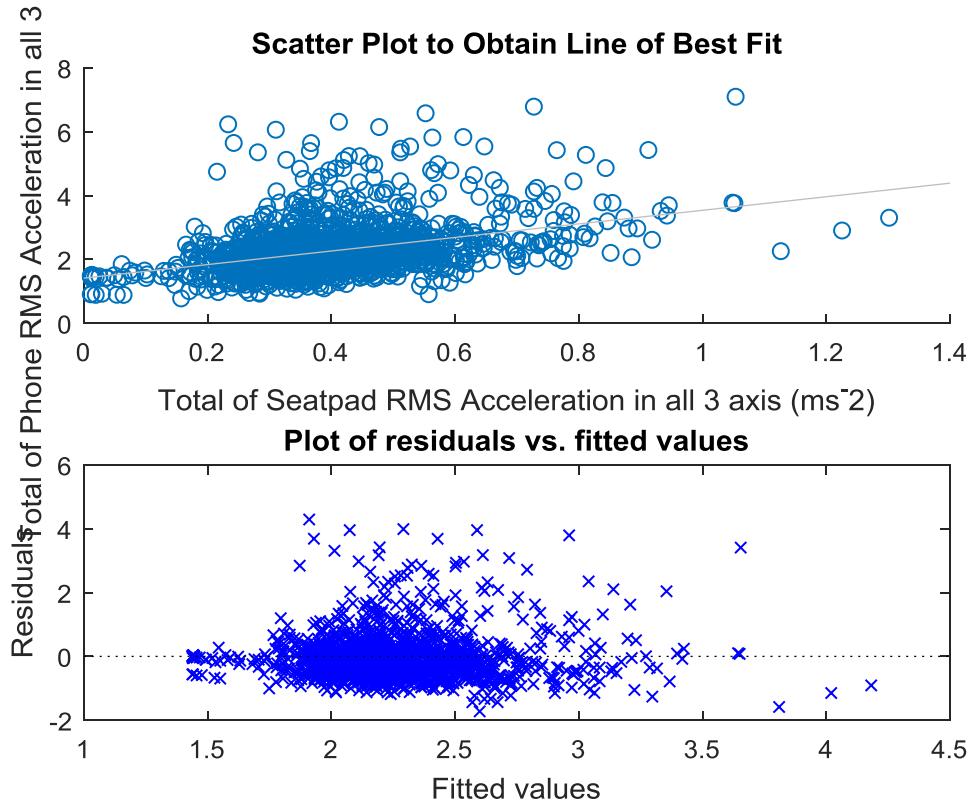


Figure C.17 – Line of best fit fitted to total RMS Accelerations in all 3 axis & Residual plot to show stochastic nature of error

Linear regression model:

$y \sim 1 + x1$

Estimated Coefficients:

	Estimate	SE	tstat	pValue
(Intercept)	1.4113	0.056589	24.939	9.8876e-114
x1	2.1258	0.13149	16.167	5.3551e-54

Number of observations: 1390, Error degrees of freedom: 1388

Root Mean Squared Error: 0.759

R-squared: 0.158, Adjusted R-Squared 0.158

F-statistic vs. constant model: 261, p-value = 5.36e-54

Figure C.18 – Linear regression model applied to total RMS Accelerations in all 3 axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

C.7 – Urban 2

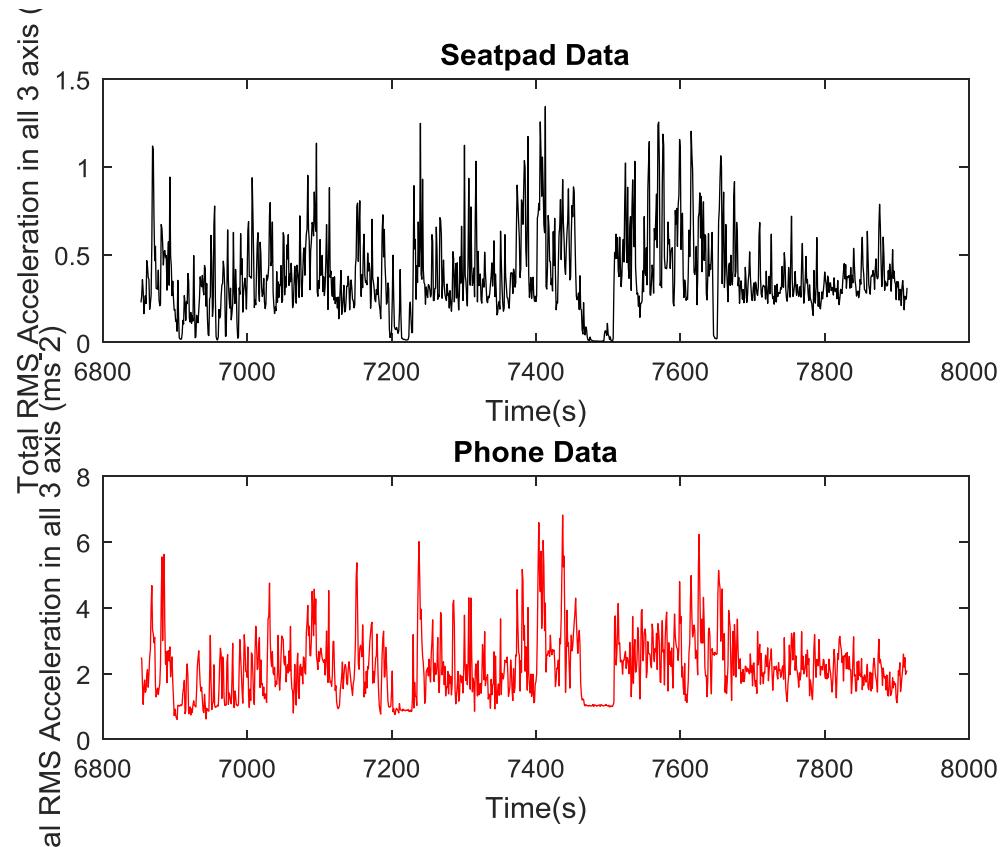


Figure C.19 – Time aligned Total RMS Accelerations in all 3 axis obtained from the seatpad and the phone app

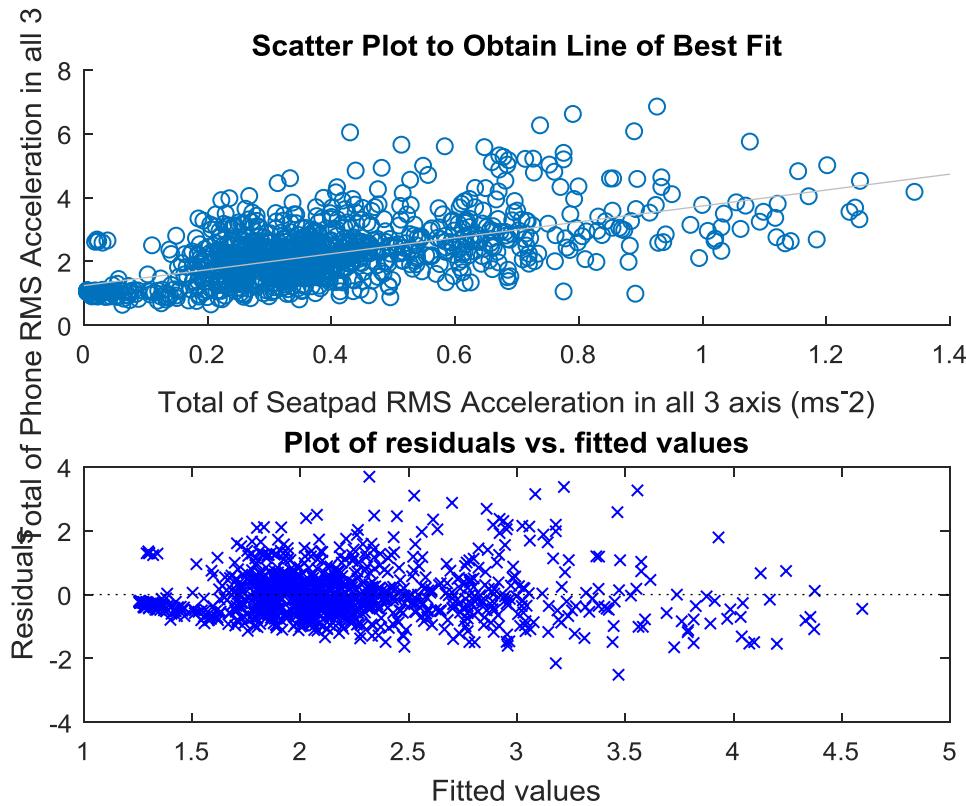


Figure C.20 – Line of best fit fitted to total RMS Accelerations in all 3 axis & Residual plot to show stochastic nature of error

Linear regression model:

$$y \sim 1 + x_1$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	1.2407	0.045697	27.15	1.1727e-123
x1	2.4953	0.10247	24.352	2.1672e-104

```
Number of observations: 1062, Error degrees of freedom: 1060
Root Mean Squared Error: 0.767
R-squared: 0.359, Adjusted R-Squared 0.358
F-statistic vs. constant model: 593, p-value = 2.17e-104
```

Figure C.21 – Linear regression model applied to total RMS Accelerations in all 3 axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

C.8 – Rural 4

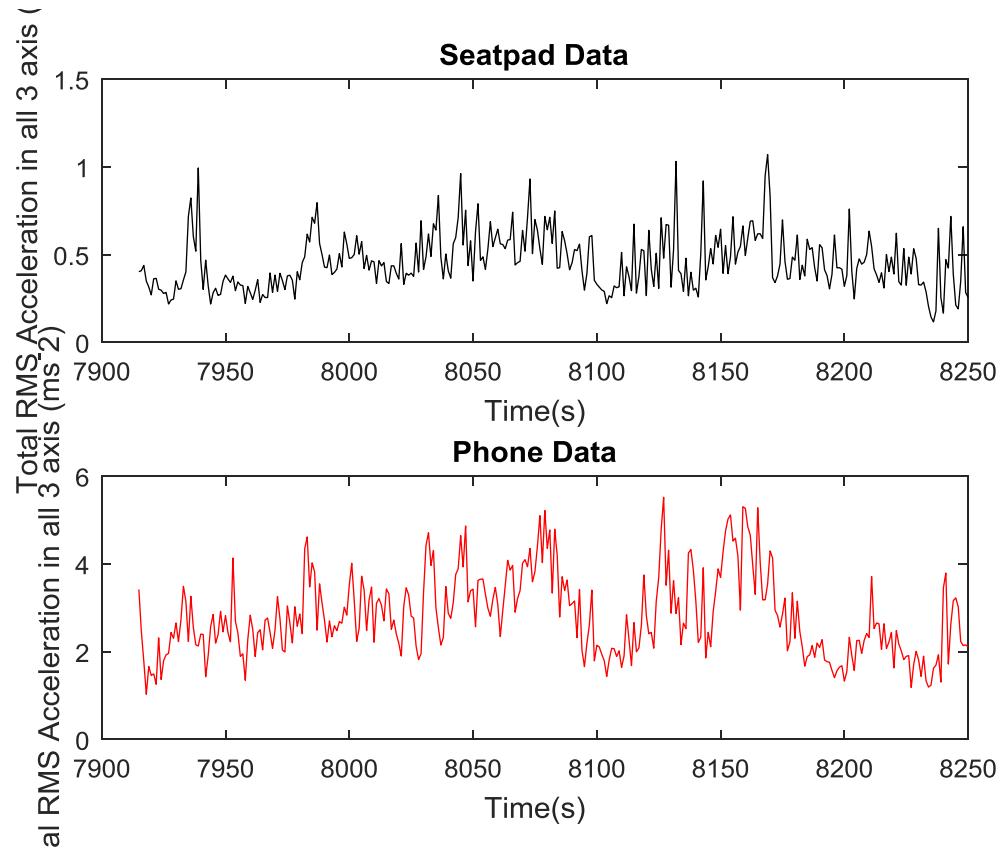


Figure C.22 – Time aligned Total RMS Accelerations in all 3 axis obtained from the seatpad and the phone app

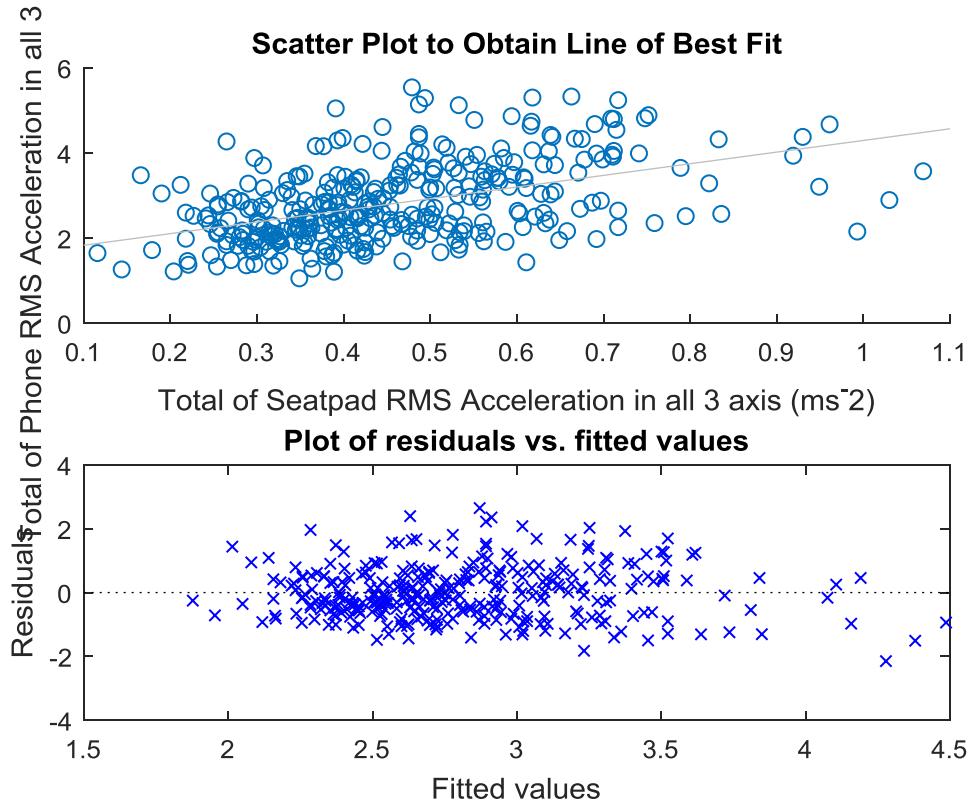


Figure C.23 – Line of best fit fitted to total RMS Accelerations in all 3 axis & Residual plot to show stochastic nature of error

Linear regression model:

$$y \sim 1 + x_1$$

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	1.5574	0.13628	11.429	9.4113e-26
x1	2.7368	0.28049	9.7571	5.9359e-20

Number of observations: 336, Error degrees of freedom: 334

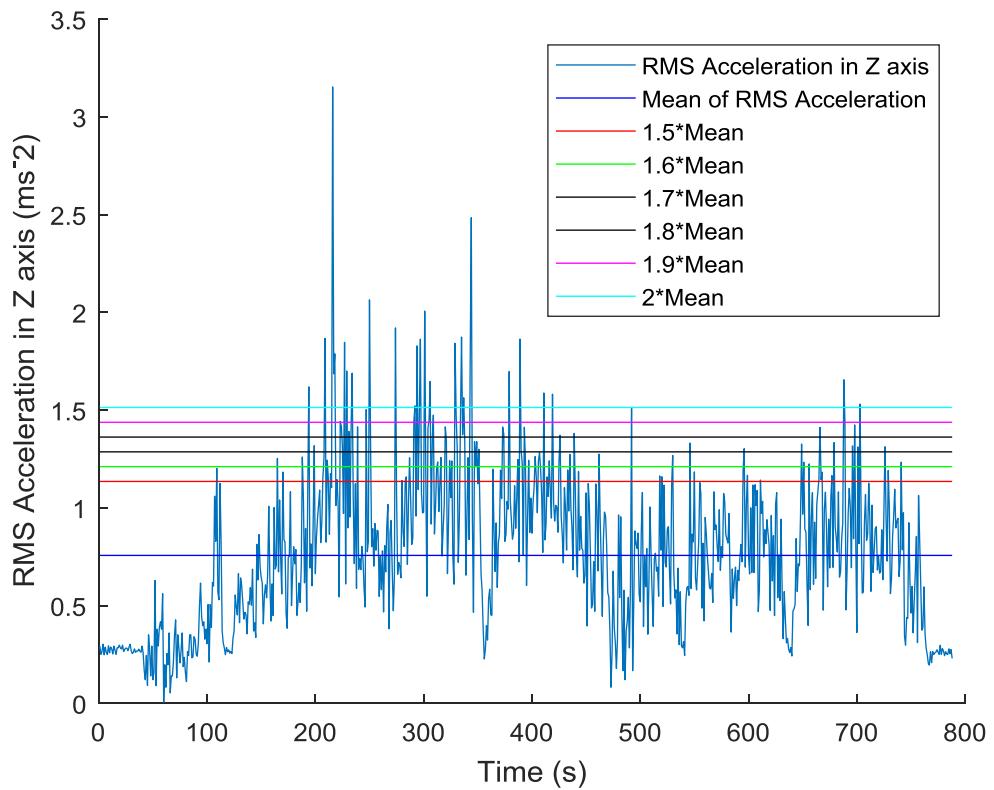
Root Mean Squared Error: 0.821

R-squared: 0.222, Adjusted R-Squared 0.219

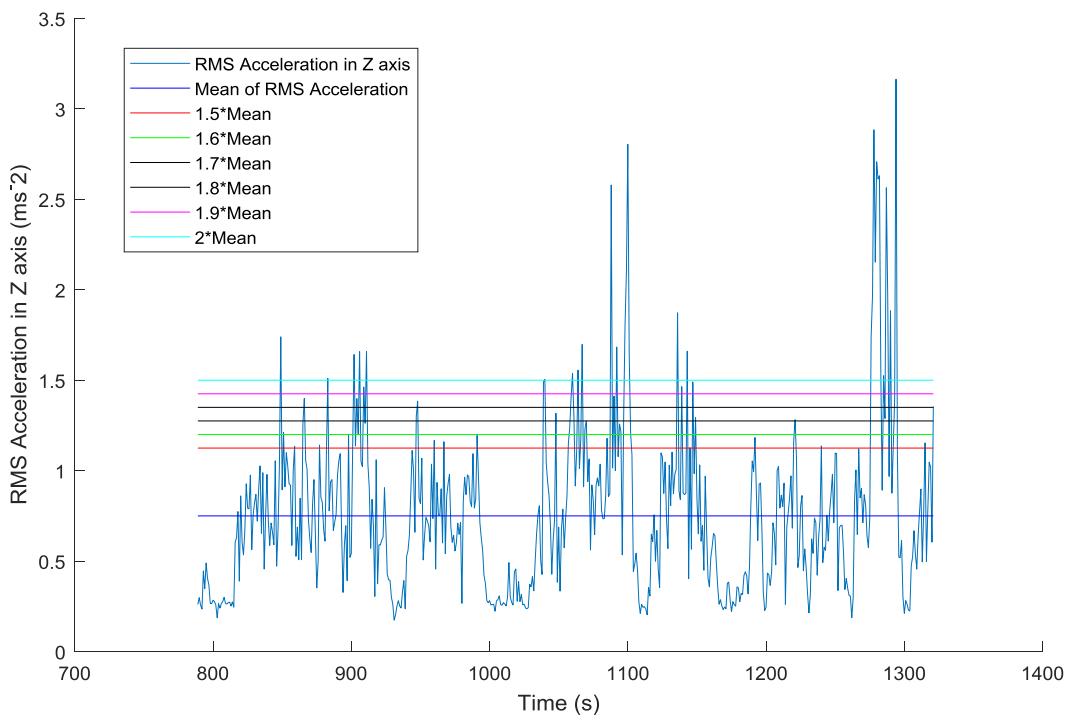
F-statistic vs. constant model: 95.2, p-value = 5.94e-20

Figure C.24 – Linear regression model applied to total RMS Accelerations in all 3 axis. Provides data on the coefficient of determination and gradient of line of best fit along with the errors associated with it

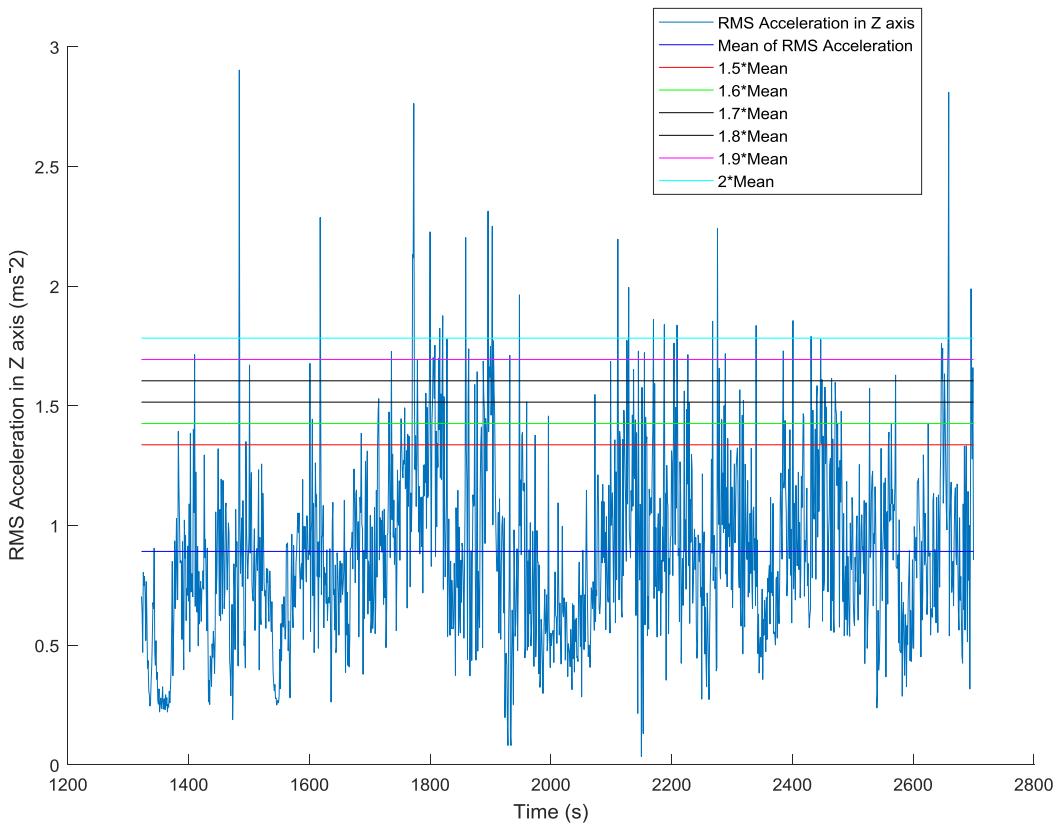
## Appendix D



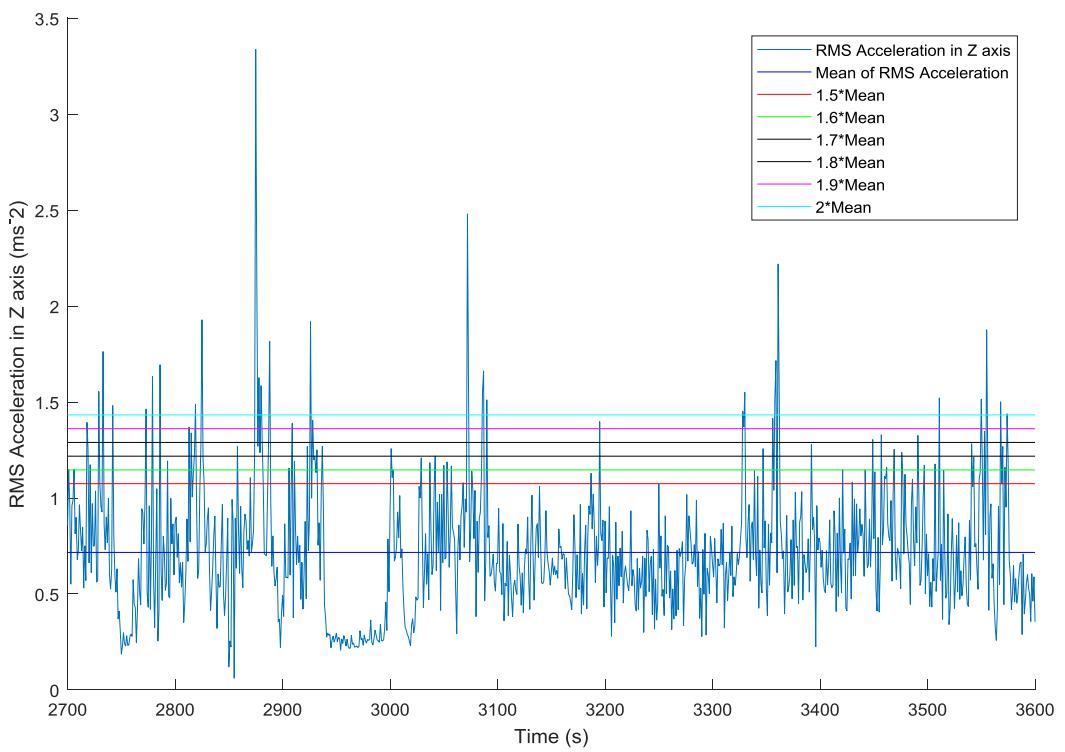
D.1 – RMS Accelerations in the Z axis for Rural 1 split into several interval ranges shown by the multicolored horizontal lines.



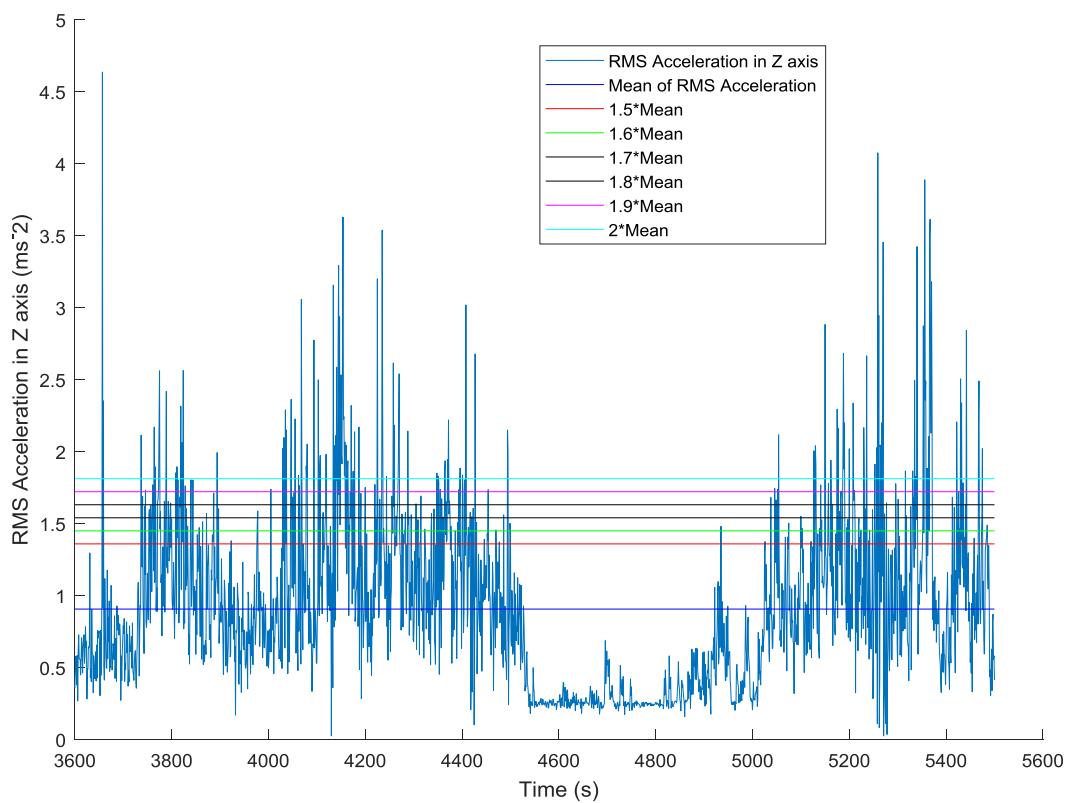
D.2 – RMS Accelerations in the Z axis for Urban 1 split into several interval ranges shown by the multicolored horizontal lines.



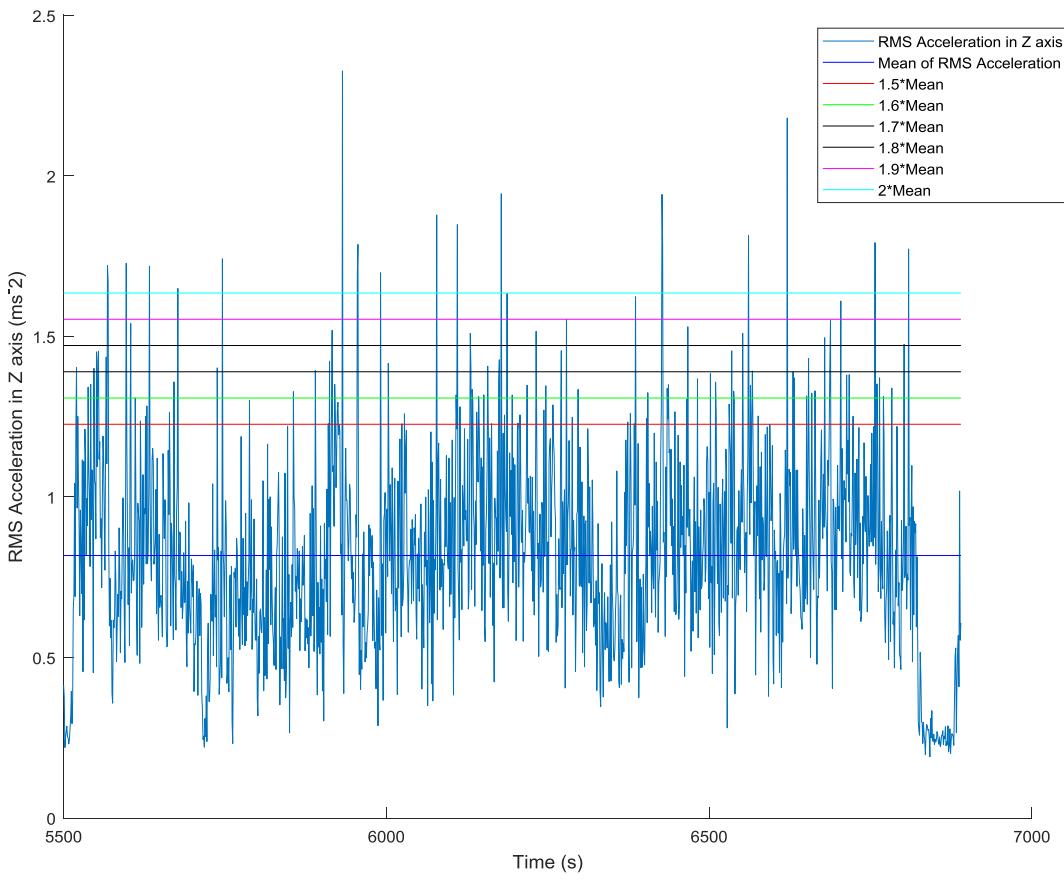
D.3 – RMS Accelerations in the Z axis for Rural 2 split into several interval ranges shown by the multicolored horizontal lines.



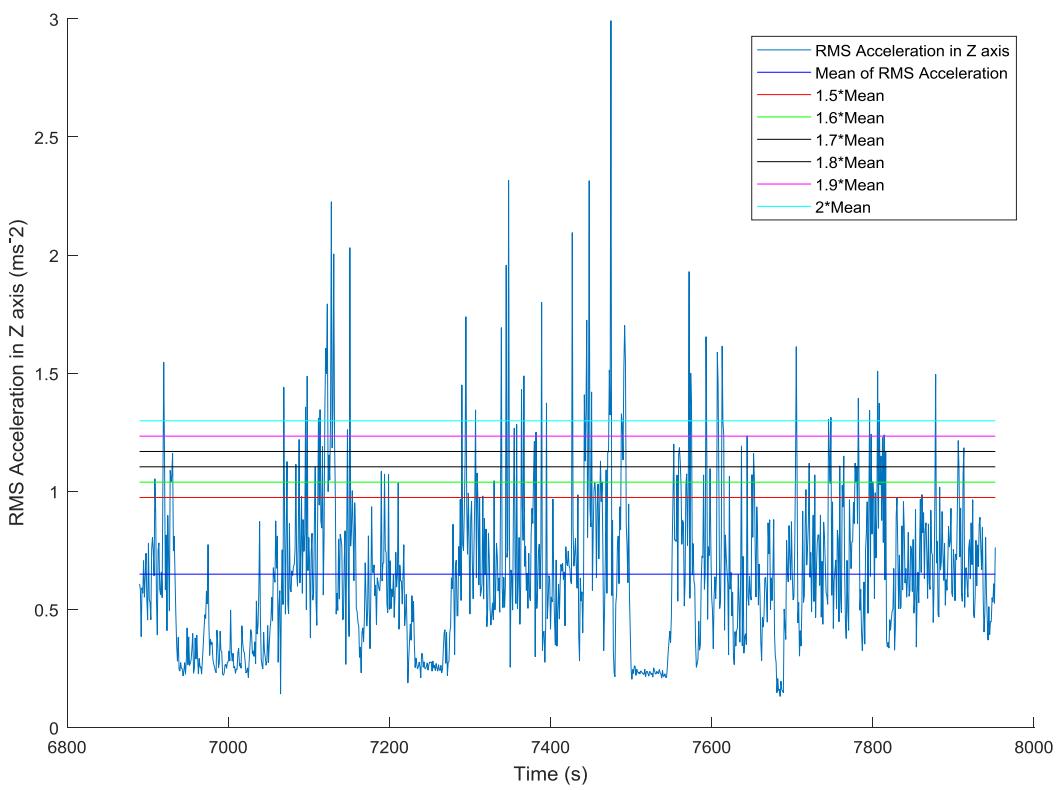
D.4 – RMS Accelerations in the Z axis for Motorway 1 split into several interval ranges shown by the multicolored horizontal lines.



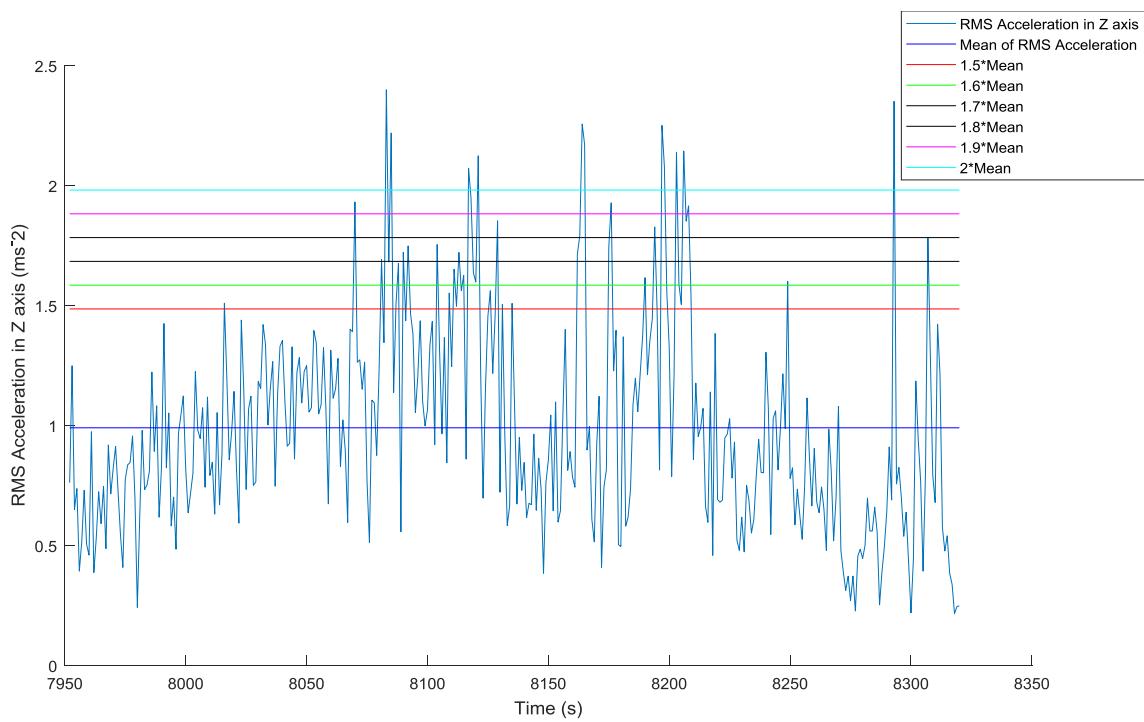
D.5 – RMS Accelerations in the Z axis for Rural 3 split into several interval ranges shown by the multicolored horizontal lines.



D.6 – RMS Accelerations in the Z axis for Motorway 2 split into several interval ranges shown by the multicolored horizontal lines.



D.7 – RMS Accelerations in the Z axis for Urban 2 split into several interval ranges shown by the multicolored horizontal lines.



D.8 – RMS Accelerations in the Z axis for Rural 4 split into several interval ranges shown by the multicolored horizontal lines.