A Real-Time Motion Based Adaptive Fuel Monitoring Technique For Vehicle Tracking Systems

Md. Miraz Hossain; ID:011 151 072
Mohammad Maruf Imtiaze; ID: 011 151 269
Md. Tafsir Hossain; ID: 011 151 274
Faizah Farzana; ID: 011 153 028

Department of Computer Science and Engineering
United International University

A project report submitted for the degree of

BSc in Computer Science & Engineering

May 2019

Declaration

We, Md. Miraz Hossain, Mohammad Maruf Imtiaze, Md. Tafsir Hossain and Faizah Farzana, declare that this thesis titled, **A Real-Time Motion Based Adaptive Fuel Monitoring Technique For Vehicle Tracking Systems** and the work presented in it are our own. We confirm that:

- This work was done wholly or mainly while in candidature for a BSc degree at United International University.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely our own work.
- We have acknowledged all main sources of help.
- Where the thesis is based on work done by ourselves jointly with others, We have made clear exactly what was done by others and what we have contributed ourselves.

(Md. Miraz Hossain) [ID: 011151072]

(Mohammad Maruf Imtiaze) [ID: 011151269]

(Md. Tafsir Hossain) [ID: 011151274]

(Faizah Farzana) [ID: 011153028]

Certificate

I do hereby declare that the research works embodied in this thesis entitled **A Real-Time Motion Based Adaptive Fuel Monitoring Technique For Vehicle Tracking Systems** is the outcome of an original work carried out by Md. Miraz Hossain, Mohammad Maruf Imtiaze, Md. Tafsir Hossain and Faizah Farzana, under my supervision.

I further certify that the dissertation meets the requirements and the standard for the degree of BSc in Computer Science and Engineering.

Dr. Salekul Islam
Professor & Head of the Dept.
Department of Computer Science and Engineering,
United International University,
United City, Madani Avenue, Badda, Dhaka 1212, Bangladesh.

Abstract

The thriving economic growth in Bangladesh has greatly promoted the demand for motor vehicles and escalated the consumption of fuel. As a result, fuel prices are constantly on the increase, making monitoring of fuel usage and theft prevention a dire need of consumers. As a solution, vehicle tracking system has been a welcome trend in our country. An integral part of VTS is fuel monitoring, which allows the client to accurately monitor fuel status, consumption behavior and offer saving opportunities that altogether help optimize operational costs, manage assets efficiently and improve profitability. In this project, we have worked on improving the fuel monitoring service of Prohori, a vehicle tracking system developed by Pi Labs Bangladesh Ltd, by integrating motion parameters into the system. In this project we design a compact module to observe the eects of a vehicles orientation, acceleration and vibration on its fuel level and process the data accordingly to detect and decrease errors in the fuel monitoring reports, that in turn, improves the eciency of a vehicle tracking system.

Dedicated to our family and in our friends, who have supported and encouraged us to complete this work.

Acknowledgements

We would like to take this opportunity to express our gratitude to our Supervisor of this Thesis, Dr. Salekul Islam, Professor & Head, and our Mentor Abu Shafin Mohammad Mahdee Jameel, Assistant Professor, of the Dept. of Computer Science and Engineering for providing us guidance, support and valuable advice in accomplishing our thesis and thanks for his constructive criticism, valuable suggestions and constant encouragement at all stages of thesis work and writing thesis book. They taught us how to work hard and growing confidence in us when we doubted ourselves and brought out the good ideas in us. Without their encouragement and constant guidance, we could not have finished this thesis. Without their contribution we would not be able to conduct this thesis. Our special thanks also to our parents and all other teachers who supported us throughout our academic year. At the last, special thanks to all friends and individuals who have directly or indirectly offered help, suggestions and support in bringing towards the completion of this thesis.

Contents

List of Figures **xi**

List of Tables **xii**

1 Project Overview **1**

1.1 Introduction 1

1.2 Purpose of the Project 2

1.3 Problem Statement 2

1.4 Project Targets 2

2 Background **3**

2.1 Introduction 3

2.2 Market Research 3

2.3 Related Work 4

3 Theory **5**

3.1 Introduction 5

3.2 Fuel Level Measurement Using Level Sensor 5

3.3 Effect of Motion Parameters on Fuel Level 7

3.3.1 Orientation 7

3.3.2 Acceleration 8

3.3.3 Vibration 9

3.4 Data Analysis 10

3.4.1 Data Filtering 10

3.4.2 Correlation Among Parameters 10

4 Complex Engineering Decisions **11**

4.1 Introduction 11

4.2 Comparison of Hardware Modules 11

4.2.1 Development Board 11

4.2.1.1 Decision 11

4.2.2 Orientation and Accelerometer Sensor 12

4.2.2.1 Decision 12

4.2.3 Vibration Sensor 12

4.2.3.1 Decision . . . 12
4.2.4 Time Keeping Module . . . 12
4.2.4.1 Decision . . . 12
4.2.5 Storage Module . . . 13
4.2.5.1 Decision . . . 13
4.2.6 Communication Module . . . 13
4.2.6.1 Decision . . . 13
4.2.7 Power Supply . . . 13
4.2.7.1 Decision . . . 13
4.3 Comparison of IDE and Programming Language . . . 14
4.3.1 Hardware Interfacing . . . 14
4.3.1.1 Decision . . . 14
4.3.2 Circuit Design . . . 14
4.3.2.1 Decision . . . 14
4.3.3 Data Analysis . . . 15
4.3.3.1 Decision . . . 15
4.3.4 Mobile Application . . . 15
4.3.4.1 Decision . . . 15
4.4 Documentation Tools . . . 16
4.4.1 Comparison . . . 16
4.4.1.1 Decision . . . 16

5 System Structure 17
5.1 Introduction . . . 17
5.2 Hardware System . . . 18
5.2.1 Overview . . . 18
5.2.2 Circuit Diagram . . . 19
5.3 Software System . . . 20
5.3.1 Overview . . . 20
5.3.2 Hardware Interfacing Program . . . 20
5.3.3 Run-Time Data Observation Mobile Application . . . 21
5.3.4 Data Analysis Program . . . 22

6 Experimental Analysis And Results 23
6.1 Overview . . . 23
6.2 Attributes and Instances . . . 23
6.3 Filtering . . . 24
6.3.1 Moving Average Filtering . . . 24

6.3.2 Savitzky-Golay Filtering . . . 25
6.3.3 Local Regression Smoothing . . . 26
6.3.3.1 LOWESS (Locally Weighted Scatter plot Smoothing) . . 27
6.3.3.2 Robust LOWESS . . . 27
6.3.3.3 LOESS (Locally Estimated Scatterplot Smoothing) . . . 28
6.3.3.4 Robust LOESS . . . 28
6.4 Analyzing Filtering Results . . . 29
6.5 Correlation Observation . . . 29
6.5.1 Correlation Equation . . . 30
6.6 ADC Ratio . . . 32
6.7 Unusual Activity Detection from ADC Ratio . . . 33
6.8 Fuel Refill Detection from ADC Ratio . . . 34
6.9 Accuracy Analysis . . . 35

7 Standards 37
7.1 Introduction . . . 37
7.2 Compliance With standards . . . 37
7.2.1 Technology . . . 37
7.2.2 Ethics . . . 37
7.2.3 Licenses . . . 37
7.2.3.1 License for Device . . . 37
7.2.3.2 License for Programming Languages . . . 38
7.2.3.3 License for IDE . . . 38
7.2.3.4 License for Documentation Tools . . . 38
7.2.4 Safety . . . 38

8 Impacts And Constraints 39
8.1 Impacts . . . 39
8.1.1 Economic Impact . . . 39
8.1.2 Social Impact . . . 39
8.1.3 Environmental Impact . . . 39
8.1.4 Ethical Impact . . . 39
8.2 Constraints . . . 40
8.2.1 Political Constraint . . . 40
8.2.2 Real Time Response . . . 40
8.2.3 Manufacturability . . . 40
8.2.4 Portability . . . 40
8.2.5 Sustainability . . . 40

8.2.6 Cost Effectiveness 40

9 Conclusion, Limitation And Future Work 41

9.1 Conclusion 41

9.2 Limitation 41

9.3 Future Work 42

Bibliography 43

A Project Timeline 44

A.1 Timeline of Trimester 1 44

A.2 Timeline of Trimester 2 44

B Work Division 45

List of Figures

2.1 Product Comparison Study 3

3.1 Fuel Level Sensor 6
3.2 Effect of Orientation on Fuel 7
3.3 Effect of Acceleration on Fuel 8
3.4 Effect of Vibration on Fuel 9

5.1 System Overview 17
5.2 Hardware System Breakdown 18
5.3 Prototype 18
5.4 Circuit Diagram 19
5.5 Software System Breakdown 20
5.6 .CSV Output File Containing Dataset. 20
5.7 App To Observe Data In Real Time 21
5.8 Raw Data of Fuel and Sensors 22

6.1 Moving Average Filtering- Fuel, Gyroscope, Accelerometer, Vibration Data 24
6.2 Savitzky-Golay Filtering- Fuel, Gyroscope, Accelerometer and Vibration Data 25
6.3 LOWESS Filtering- Fuel, Gyroscope, Accelerometer and Vibration Data 27
6.4 Robust LOWESS Filtering- Fuel, Gyroscope, Accelerometer, Vibration Data 27
6.5 LOESS Filtering- Fuel, Gyroscope, Accelerometer and Vibration Data . . 28
6.6 Robust LOESS Filtering- Fuel, Gyroscope, Accelerometer, Vibration Data 28
6.7 Effect of GyroY on Fuel ADC Value 29
6.8 Fuel ADC Value Computed From Correlation Equation: Example- 1 . . . 30
6.9 Fuel ADC Value Computed From Correlation Equation: Example- 2 . . . 31
6.10 Fuel ADC Value Computed From Correlation Equation: Example- 3 . . . 31
6.11 Fuel ADC Ratio 32
6.12 Unusual Increase In Fuel ADC Ratio Indicating Suspicious Incident . . . 33
6.13 Refill Detection Through Drop In Fuel ADC Ratio 34
6.14 Incorrect Result From Fuel ADC Value Computation: Example- 1 35
6.15 Incorrect Result From Fuel ADC Value Computation: Example- 2 36
6.16 Incorrect Result From Fuel ADC Value Computation: Example- 3 36

List of Tables

4.1 Comparison Between Development Boards 11
4.2 Comparison Between Sensors 12
4.3 Comparison Between Sensors 12
4.4 Comparison Between Time Keeping Modules 12
4.5 Comparison Between Storage Modules 13
4.6 Comparison Between Communication Modules 13
4.7 Comparison Between Power Supply Components 13
4.8 Comparison Between Hardware Interfacing IDE 14
4.9 Comparison Between Programming Languages for Hardware Interfacing . 14
4.10 Comparison Between Circuit Design IDE 14
4.11 Comparison Between IDEs and Programming Languages for Data Analysis 15
4.12 Comparison Between Mobile Application Development IDE 15
4.13 Comparison Between Programming Languages for Mobile App Development 15
4.14 Comparison Between Documentation Tools 16

6.1 Datasets and Instances 23
6.2 Attributes of Dataset 23

7.1 Technology Standards 37

B.1 Work Distribution 45

Chapter 1
Project Overview

1.1 Introduction

Vehicle Tracking System (VTS) is a growing technology worldwide. It incorporates a number of techniques to allow continuous tracking, monitoring and controlling of a vehicle. The system uses both hardware and software technologies to collect and analyze real time data of vehicles and provide the clients with a range of services, such as live tracking, mileage report, fuel monitoring, theft prevention and many more, that help optimize operational costs, manage assets efficiently and improve profitability. As the era of automated vehicles is gradually taking over consumer life, the system is gaining popularity day by day in both personal and industrial sectors.

In Bangladesh, the spiraling economic development has greatly promoted the usage of motor vehicles and hence, the use of fuel. The ever increasing demand of motor vehicles has resulted in rapid decrease in our natural resources while the price of the fuel is constantly on the rise. Over and above that, stealth of fuel is also increasing day by day for the purpose of quick and low-risk illegal income. All in all, the vehicle owners are facing a large economic loss that puts a highly negative effect on our national resources and revenues. Therefore, monitoring and ensuring proper use of fuel is a paramount issue in our country.

As a solution, Vehicle tracking system has been a welcome trend in our country. It is an emerging technology with more than 20 companies competing in the national market. The VTS products offer the customers a number of different facilities, though only a few offer fuel monitoring service, which helps the clients to monitor real time fuel status and consumption rate and allow them to adopt saving opportunities to cut down cost and unnecessary usage. Different companies use different methods to monitor fuel consumption and consequently, the nature of the monitoring process varies and yielded results differ from each other as well.

For our Capstone Project, we have worked on improving the fuel monitoring service of **Prohori**, a vehicle tracking system developed by **Pi Labs Bangladesh Ltd.** Prohori has been competing in the VTS market with top-notch innovative processes and quality assurance. Fuel monitoring is an integral part of this system. We hope to integrate an innovative solution to this system for achieving a better monitoring procedure.

1.2 Purpose of the Project

The purpose of the project is to integrate our own innovation into the fuel monitoring system of **Prohori** to improve the accuracy of fuel level measurement. The current system can measure the fuel level with an accuracy of ± 2 Liters. Our idea is to increase the parameters of the fuel monitoring system so it can take more elements into consideration and produce more accurate results. To accomplish that, we have built a new hardware module that hosts a number of sensors in order to collect data from the vehicles and we have also developed an algorithm to process and analyze the data sets using different methods that yield less erroneous fuel monitoring data.

The observations from the processed data can be vital in finding out relationship between the motion parameters of a vehicle and it's fuel consumption pattern. If any relationship exists and can be proved in this project, the findings can be used in the VTS business models as accuracy improvement measure. As business perspective has to be kept in mind, the prototype developed in this project must be cost friendly, compact and easy to manufacture.

1.3 Problem Statement

In this project, we hope to design a compact module combining both hardware and software to observe the effects of a vehicle's orientation, acceleration and vibration on its fuel level and process the data accordingly to detect and decrease errors in the fuel monitoring reports, that will in turn, improve the efficiency of a vehicle tracking system.

1.4 Project Targets

In this project, we aim to achieve **three** specific targets:

1. Find out correlation between fuel data and motion parameters (Orientation, Acceleration, Vibration) of a vehicle.
2. Detect unusual consumption activity (such as tank leaks, fuel theft).
3. Detect fuel refill.

Chapter 2

Background

2.1 Introduction

In recent years, fuel prices have been increasing at a high rate worldwide. Before automated VTS, the fuel management analysis was done by scanning paperwork and mileage of vehicles. This method could not provide instantaneous and precise results due to lack of real time data and unlawful activities such as fuel theft, unauthorized trips, unregistered load etc. VTS aptly deals with all these problems and provides the user with real time data, schedule management, consumption analysis and provide effective ways to save cost and energy.

2.2 Market Research

As VTS is a relatively new technology in Bangladesh, it is somewhat tough to gather information on the existing systems and how they work. We have looked into a few other popular VTS providers with fuel monitoring service in Bangladesh and conducted a comparative benchmark study on them. However, technical details about how these products monitor the fuel status is highly confidential information and therefore we could only work with the information provided by Pi Labs.

Company	GPS Tracking	Fuel Monitoring	Speed Analysis	Engine & AC Control	SMS Alert	Report
EasyTrax	✓	Real Time	✓	✓	✓	✓
Finder	✓	Distance Based	✓	✗	✓	✗
NTrack	✓	Distance Based	✓	✓	✓	✓
Grameenphone VTS	✓	Distance Based	✓	✗	✓	✓
Robi Tracker	✓	Distance Based	✓	✓	✓	✓
MobiTrack	✓	Distance Based	✓	✗	✓	✓
Prohori	✓	Real Time	✓	✓	✓	✓

Figure 2.1: Product Comparison Study -

From the comparison chart, we can see some companies use real time fuel level monitoring while others use distance based monitoring method. The distance based method often do not provide proper fuel consumption data as it does not consider scenarios such as traffic jam, temperature, ac status etc. that have major influence on the consumption rate. Real time monitoring takes such scenarios into account and therefore provides much more accurate data. For this reason, we choose to work on **Prohori** that uses real time monitoring.

2.3 Related Work

In 2018, Stratis Kanarachos et al [1] designed a system where instantaneous fuel consumption of a vehicle was estimated using Recurrent Neural Networks that process a smartphones GPS position, speed, altitude, acceleration and number of visible satellites.

In 2018, Rajesh Bose et al [2] explained the working principles of a real time fuel monitoring system that used capacitive fuel level sensor. They used a GPS-GPRS fuel monitoring controller to collect and remotely store the fuel consumption data.

In 2017, Safa Abd elmonem. Yosif et al [3] explained the working principles of a real time fuel monitoring system that used capacitive fuel level sensor. They used a GPS-GPRS fuel monitoring controller to collect and remotely store the fuel consumption data.

In 2016, Ahmet Gurcan Capraz et al [4] conducted a comparison of statistical fuel consumption models applied to automobiles using real time data.

In 2012, Sachin S. Aher et al [5] developed a bus tracking and fuel monitoring system based on Arduino, GPS and fuel level sensor to measure the fuel level and store the data on the server.

Chapter 3
Theory

3.1 Introduction

A vehicle tracking system collects various data whose combination provides the status of a vehicle, such as location, fuel consumption, route history, AC usage, speed analysis and many more. The trouble with the fuel level measurement is that since fuel is a liquid material, its level may fluctuate depending on the orientation, acceleration and vibration of the vehicle.

The current systems measure the level with data from the float switch only- this system fails to differentiate the scenarios that cause sudden and short-lived changes in the fuel level, such as when the vehicle is in a tilted position, has hit a hard brake or when the fuel tank is refilled. Our theory is to take into consideration all these parameters that affect the fuel level in the fuel tank and combine all these data to detect and observe the effects at any given time.

In this chapter, we discuss the concepts we make use of to plan and implement our project.

3.2 Fuel Level Measurement Using Level Sensor

Fuel tanks inside vehicles have level sensors installed in them for measuring the level of fuel. Fuel level sensor has a simple operating model. It consists of three components: a float, a resistor and an actuating rod. The variable resistor has a resistive material strip that is connected to the ground. A wiper is connected to the fuel gauge and it conducts current from the gauge to the resistor by moving along the material strip. The distance of the wiper from the grounded part of the strip determines the resistance level. When fuel level is high, the float is closer to the top. So, the wiper is closer to the grounded part and resistance is small and vice versa when the fuel level is low. Therefore, resistance is the most when the tank is empty and the least when the tank is full. This way,the level sensor sends a variable signal to the gauge, where smaller current makes the needle move towards the **empty** point and larger current swings it towards the **full** point.

Figure 3.1: Fuel Level Sensor -

As fuel is a liquid material, it seldom stays in a stable position during the movement of a vehicle. Together with the structure and mechanism of the level sensor to that, the fuel gauge often provides inaccurate reading.

3.3 Effect of Motion Parameters on Fuel Level

3.3.1 Orientation

When a vehicle goes over an upward or downward slope, it's orientation along the X, Y and Z axes changes respectively. As fuel tank is in a fixed position inside the vehicle, the fuel in the tanks fluctuates, causing the float position to change accordingly. The orientation of a vehicle can be measured using a Gyroscope.

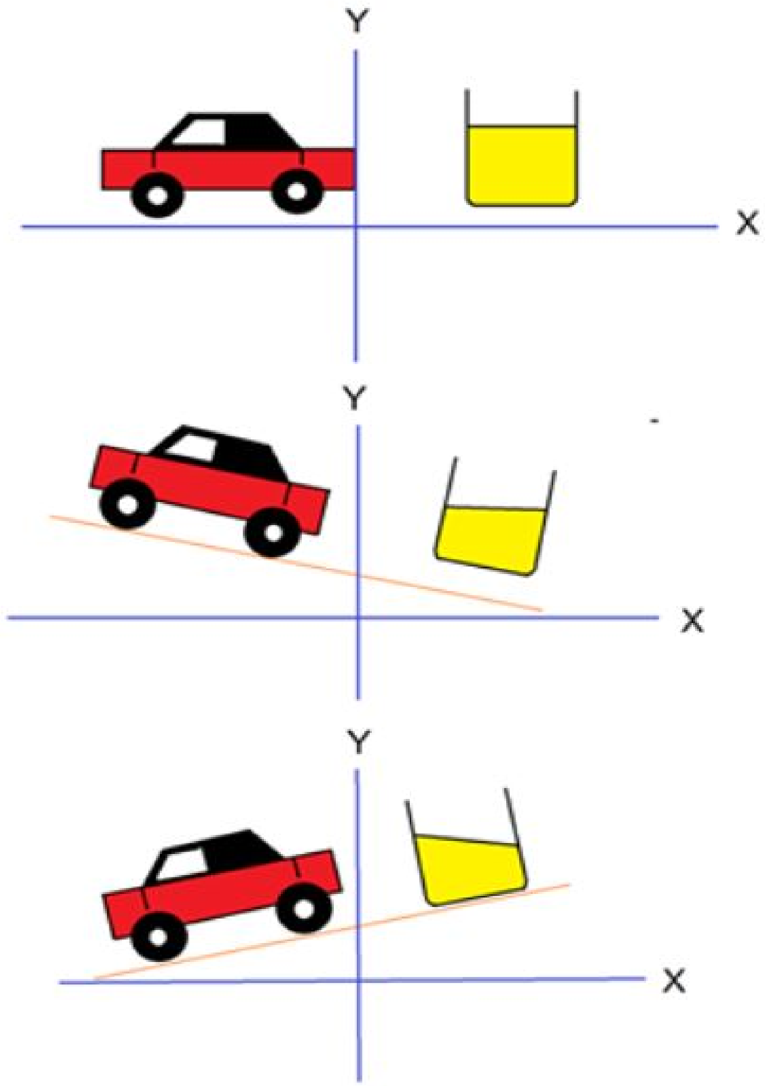

Figure 3.2: Effect of Orientation on Fuel -

3.3.2 Acceleration

Sudden movements such as hard brakes or speed sprints cause the fuel inside the tank to slosh around strongly. Therefore, the fuel level fluctuates for a time before it gets stable again, causing the float position to change as well. The acceleration of a vehicle can be measured using a Accelerometer.

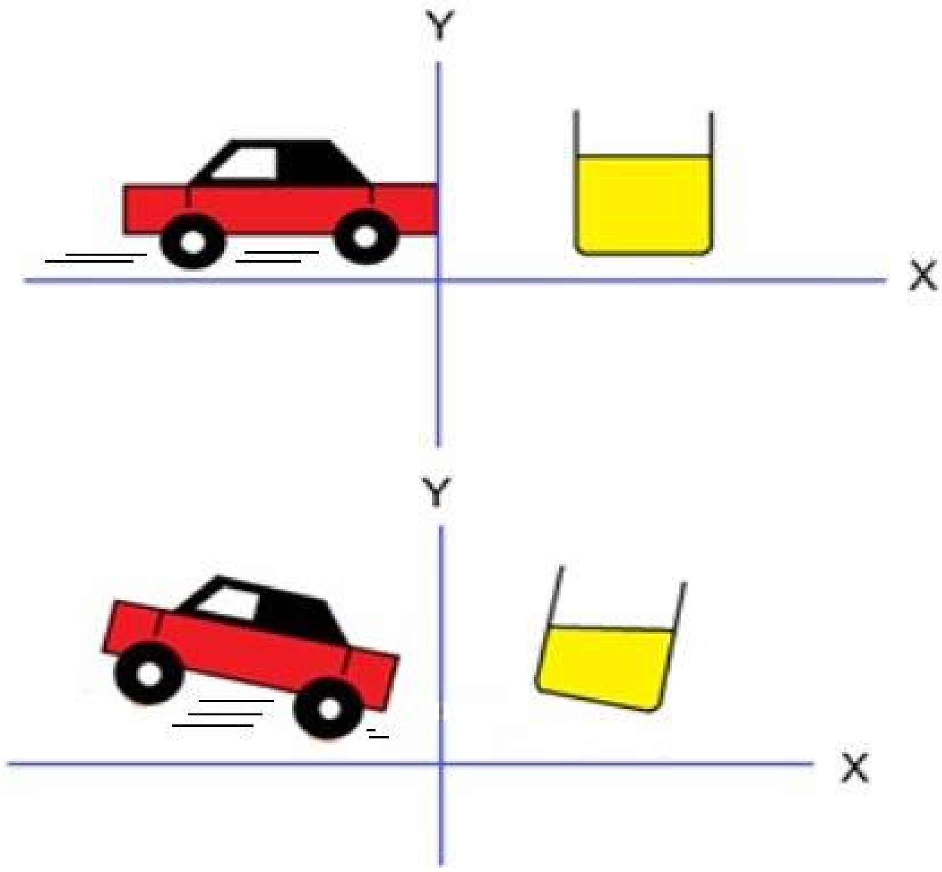

Figure 3.3: Effect of Acceleration on Fuel -

3.3.3 Vibration

Uneven roads, speed breakers, potholes etc. cause the vehicle to vibrate on different scales. Through the vehicle body and fuel tank, the vibration might propagate to the fuel as well. In that case, fuel level will fluctuate and the float position will move too. The vibration of the vehicle can be measured using a Vibration Sensor.

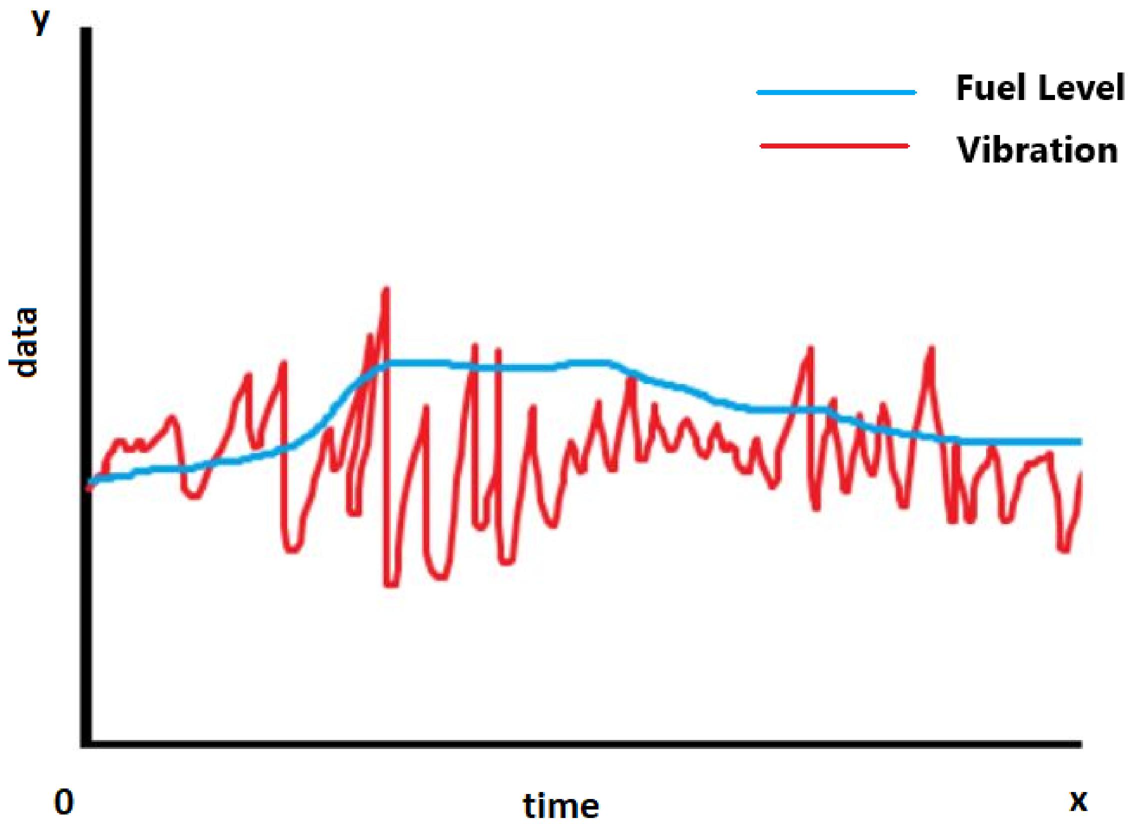

Figure 3.4: Effect of Vibration on Fuel -

3.4 Data Analysis

Data analysis is the procedure of defining, demonstrating and evaluating data using logical or statistical techniques. By analyzing data, it's possible to detect patterns, fit models and infer various other kinds of information through distinguishing actual signals from noise or fluctuations. In our project, the prototype installed on the vehicles collects all the necessary data using the motion sensors. Naturally, the raw data can not be used directly for experimentation as it contains a lot of noise. We have to eliminate or reduce the noise to a tolerable limit by analyzing the collected data using different methods and run experiments to find out the dependency of the fuel level changes on the motion parameters of the vehicle.

3.4.1 Data Filtering

The data provided by the fuel sensor are actually ADC values which change over time depending on the fuel level. Therefore, the ADC values are often noisy and do not actual fuel level correctly. To derive proper results, filtering of the data is needed. There are various filtering methods available for reducing noise in datasets. We will attempt a number of these methods to compare the results and determine the best one for our model.

3.4.2 Correlation Among Parameters

Correlation measures the relationship between two quantitative variables. If fluctuations of the values of one variable can be predicted from those of another variable, that usually reasons that change in one causes change in the other. A positive correlation refers to parallel change in the values of the variables whereas a negative correlation refers to opposite changes in the values. For our project, after the dataset is collected and filtered, we can hope to have removed unwanted noises from the initial datasets. The next step is to illustrate and observe the filtered datasets and find out existence of correlation between the motion parameters **(Orientation, Acceleration and Vibration)** and the **fuel level**.

Chapter 4
Complex Engineering Decisions

4.1 Introduction

We have conducted extensive and comparative studies on the tools and platforms required to build the project and make engineering decisions about which options are the most appropriate. The decisions were taken based on properties such as usage purpose, ease of configuration, available resources, cost effectiveness etc. In this chapter, we discuss in detail about these complex engineering decisions.

4.2 Comparison of Hardware Modules

4.2.1 Development Board

Name	Board Type	OS	Current Consumption	ADC Support	Cost (Approximate)
Arduino Mega 2560	Microcontroller Board	No	Low	Yes	800 BDT
Raspberry Pi 3	Single Board Computer	Yes	High	No	4000 BDT

Table 4.1: Comparison Between Development Boards

4.2.1.1 Decision

We choose Arduino Mega 2560 for our project, as it suits our project requirements sufficiently whereas Raspberry Pi 3 exceeds them by a wide margin.

4.2.2 Orientation and Accelerometer Sensor

Name	**Gyroscope Range**	**Accelerometer Range**	**Cost (Approximate)**
MPU 6050 6 Axis Sensor	±250, ±500, ±1000, ±2000 dps	$\pm 2/\pm 4/\pm 8/\pm 16$ g	250 BDT
Grove 6 Axis Sensor	±125, ±245, ±500, ±1000, ±2000 dps	$\pm 2/\pm 4/\pm 8/\pm 16$ g	1500 BDT

Table 4.2: Comparison Between Sensors

4.2.2.1 Decision

We choose MPU 6050 as it's almost 6 times cheaper.

4.2.3 Vibration Sensor

Name	**Sensitivity**	**I/O**	**Cost (Approximate)**
DFRobot Sensor	high	digital	450 BDT
SW 420	high	digital	105 BDT

Table 4.3: Comparison Between Sensors

4.2.3.1 Decision

We choose SW 420 is it's more cost effective for our purpose.

4.2.4 Time Keeping Module

Name	**In-built Temperature Sensor**	**Accuracy**	**Cost (Approximate)**
DS3231 Module	yes	more accurate	BDT 90
DS1307 Module	no	less accurate	BDT 130

Table 4.4: Comparison Between Time Keeping Modules

4.2.4.1 Decision

We choose DS3231 module as gives more accurate time which is vital for our project.

4.2.5 Storage Module

Board	Card Compatibility	Cost (Approximate)
Waveshare Micro SD Card Module	SD, SDHC, TF Card	BDT 350
Seeed Studio SD Card Shield	SD, SDHC, TF Card	1500

Table 4.5: Comparison Between Storage Modules

4.2.5.1 Decision

Since both modules offer similar features needed for our project, we choose the Waveshare module which is 5 times cheaper.

4.2.6 Communication Module

Name	Range	Configuration	Cost (Approximate)
WiFi Module ESP8266	very long, depends on obstacles	complicated	BDT 450
Bluetooth Module HC-05	9 meters	simple	BDT 250

Table 4.6: Comparison Between Communication Modules

4.2.6.1 Decision

We decide to use HC-05 as it's sufficient for our use, we do not need the advance features of ESP8266.

4.2.7 Power Supply

Board	Connector Type	Safety	Cost (Approximate)
LiPo Battery 2200 mAh	T connectors	caution needed	BDT 2500
Power Bank 5000 mAh	USB charger cable	safer to use	BDT 1200

Table 4.7: Comparison Between Power Supply Components

4.2.7.1 Decision

We decide to use Xiaomi 5000 mAh power bank as it is easier to setup and also safer.

4.3 Comparison of IDE and Programming Language

4.3.1 Hardware Interfacing

Name	Available Resources	Connectivity	cost
Arduino	widely available	offline/online	free
Programino	limited	offline/online	free

Table 4.8: Comparison Between Hardware Interfacing IDE

Name	Complexity	Object Orientated	Execution Speed
C	simple to use	no	faster
Java	complex, extra libraries needed	yes	slower

Table 4.9: Comparison Between Programming Languages for Hardware Interfacing

4.3.1.1 Decision

We decide on Arduino as the interfacing platform as there is a huge number of resources available for it and community support is much greater than Programino. For programming language, we choose C, as it's easier to implement and has faster execution.

4.3.2 Circuit Design

Name	Circuit Simulation	Microcontroller Simulation	PCB Design	Cost
Proteus Design Suit	yes	yes	no	free trial
LTspice	yes	no	no	free

Table 4.10: Comparison Between Circuit Design IDE

4.3.2.1 Decision

We choose Proteus Design Suit as it offers the features that we need to use for our project.

4.3.3 Data Analysis

Name	Data Handling Capability	Resources	Cost
MATLAB	better; huge libraries, functions	widely available	free trial
Julia	worse; few libraries, functions	limited	open source

Table 4.11: Comparison Between IDEs and Programming Languages for Data Analysis

4.3.3.1 Decision

We choose MATLAB as it has more available resources, libraries and functions that help with our data analysis.

4.3.4 Mobile Application

Name	Cross Platform	Language	Cost
Android Studio	no	java, kotlin	free
Xamarin	yes	c#	free trial

Table 4.12: Comparison Between Mobile Application Development IDE

Name	Exceution Speed	Available Resources	Team Proficiency
Java	faster	widely available	better
Kotlin	slower	limited	worse

Table 4.13: Comparison Between Programming Languages for Mobile App Development

4.3.4.1 Decision

We choose Android Studio as IDE because we do not need cross platform app at this moment. For programming language, we work with Java, as it has more resources and our team has better skills in it.

4.4 Documentation Tools

4.4.1 Comparison

Name	**Connectivity**	**Illustration Tools**	**Cost**
Microsoft Office	Offline	easy to use	free trial
LaTeX	online	complex to use	free

Table 4.14: Comparison Between Documentation Tools

4.4.1.1 Decision

We choose both Microsoft Office and LaTeX, as both have features that help us prepare the reports, presentations and journals.

Chapter 5

System Structure

5.1 Introduction

The system has been designed and developed to ensure it can be implemented practically and serve its purpose efficiently. The entire system was designed by our team from scratch and tested at every step for detecting and debugging problems and making required modifications. We have had to revise the design a few times till we came up with a model that suits our requirements and executes the data collection and analysis processes just as planned. The complete system can be divided into two segments- hardware and software. The hardware part collects and stores the sensor data from the vehicle and the software part analyzes the collected datasets.

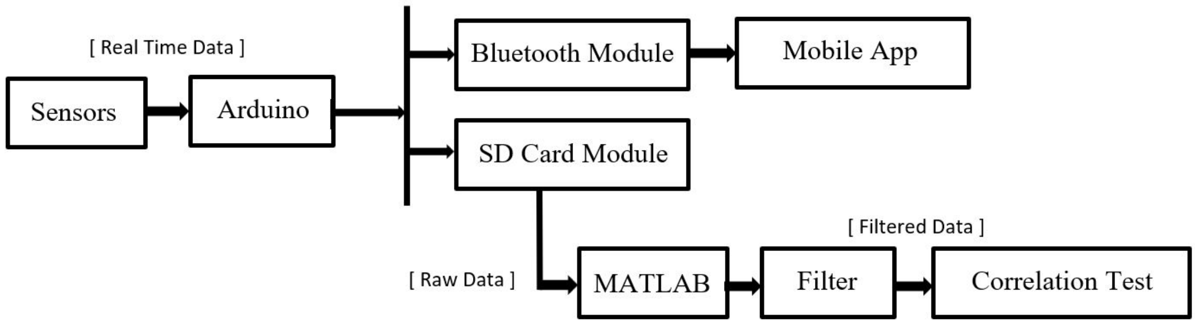

Figure 5.1: System Overview -

In this chapter, we run a breakdown of the whole system through visualization of the design, description of equipment and tools and demonstration of different functionalities.

5.2 Hardware System

5.2.1 Overview

The hardware system consists of two sensor modules (Vibration Sensor and Gyroscope and Accelerometer Sensor), an SD card module, a Bluetooth module, a an real time clock module and an Arduino Mega microcontroller board and finally, a power bank. The sensors, SD card module, RTC module and Bluetooth module are all connected to the Arduino Mega. The sensors provide respective status data of the vehicle body they are attached to and the microcontroller stores the data on the SD Card. The Bluetooth module sends the data to a mobile phone for run-time observation.

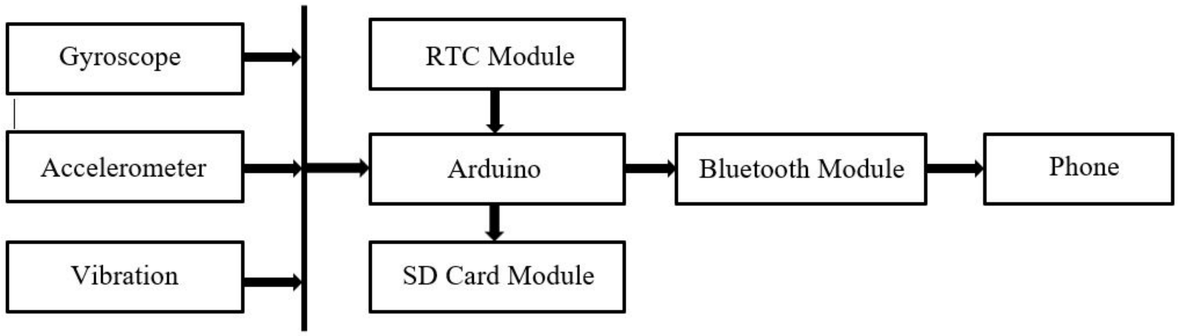

Figure 5.2: Hardware System Breakdown -

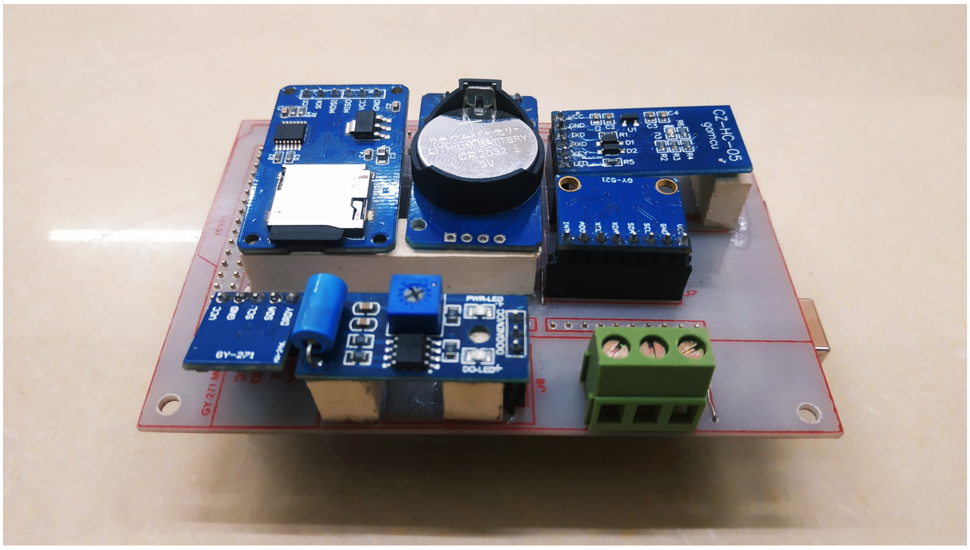

Figure 5.3: Prototype -

5.2.2 Circuit Diagram

The circuit diagram was designed for building the prototype professionally. The diagram shows the electrical connections of all the modules of our prototype. It was designed using Proteus Design Suite.

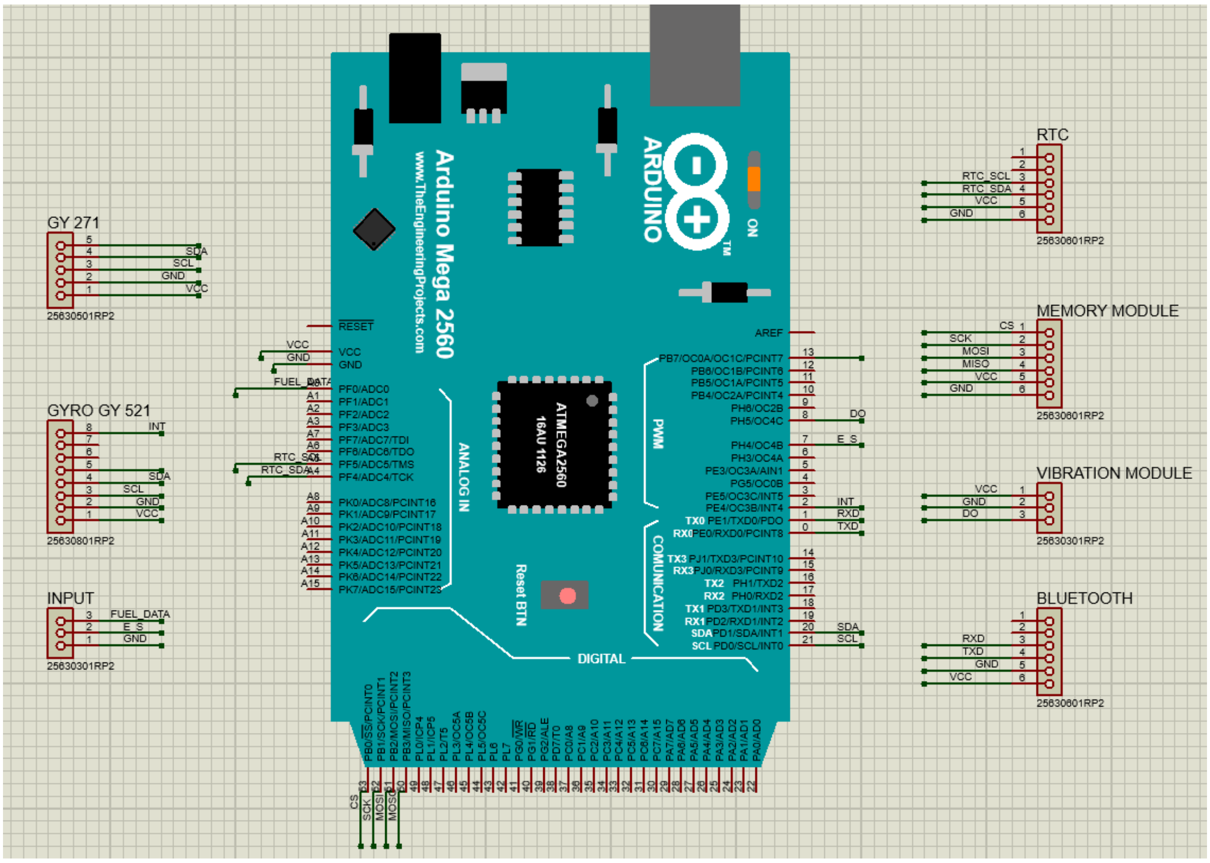

Figure 5.4: Circuit Diagram -

5.3 Software System

5.3.1 Overview

The software system is developed from base keeping in mind the requirements and purpose of the project. The software system programs the hardware device, enables instantaneous inspection of the sensor data and analyses the collected datasets.

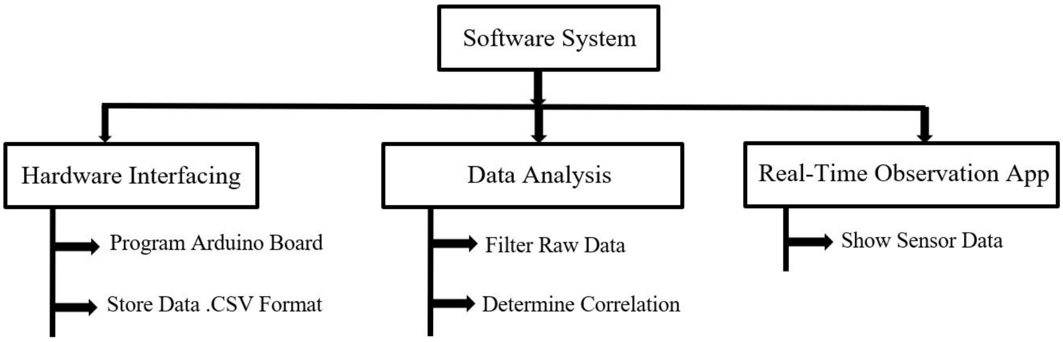

Figure 5.5: Software System Breakdown -

5.3.2 Hardware Interfacing Program

For collection, transmission and storing of data, the hardware modules have to interfaced accordingly. Each hardware module has been programmed individually in the beginning and after successful testing the programs have been merged into one single code.

1	Date	Time	Temp	GyroX	GyroY	GyroZ	AccX	AccY	Vibration	Direction	Engine	FuelDATA
2	1/2/2019	16:11:10	31.75	-11	-2	-473	-3	0	0	202	0	288
3	1/2/2019	16:11:10	31.75	-11	-2	-473	-2	1	0	202	0	288
4	1/2/2019	16:11:10	31.75	-11	-2	-473	-2	1	0	203	0	288
5	1/2/2019	16:11:10	31.75	-11	-2	-473	-3	0	0	203	0	288
6	1/2/2019	16:11:10	31.75	-11	-2	-473	-3	1	0	202	0	288
7	1/2/2019	16:11:11	31.75	-11	-2	-473	-3	1	0	202	0	288
8	1/2/2019	16:11:11	31.75	-11	-2	-473	-3	1	0	202	0	288
9	1/2/2019	16:11:11	31.75	-11	-2	-473	-3	0	0	202	0	288
10	1/2/2019	16:11:11	31.75	-11	-2	-473	-3	1	0	202	0	288
11	1/2/2019	16:11:11	31.75	-11	-2	-473	-3	0	0	203	0	288
12	1/2/2019	16:11:11	31.75	-11	-2	-473	-3	1	0	203	0	288
13	1/2/2019	16:11:11	31.75	-11	-2	-473	-3	0	0	202	0	288
14	1/2/2019	16:11:11	31.75	-11	-2	-473	-3	1	0	202	0	288
15	1/2/2019	16:11:12	31.75	-11	-2	-473	-3	1	0	202	0	288

Figure 5.6: .CSV Output File Containing Dataset. -

5.3.3 Run-Time Data Observation Mobile Application

While testing the device and collecting data, it is immensely helpful if we can see the live values of data during vehicle run-time. For this purpose, we developed an Android app that shows the values of the sensors that the microcontroller board send over to the phone through the Bluetooth Module.

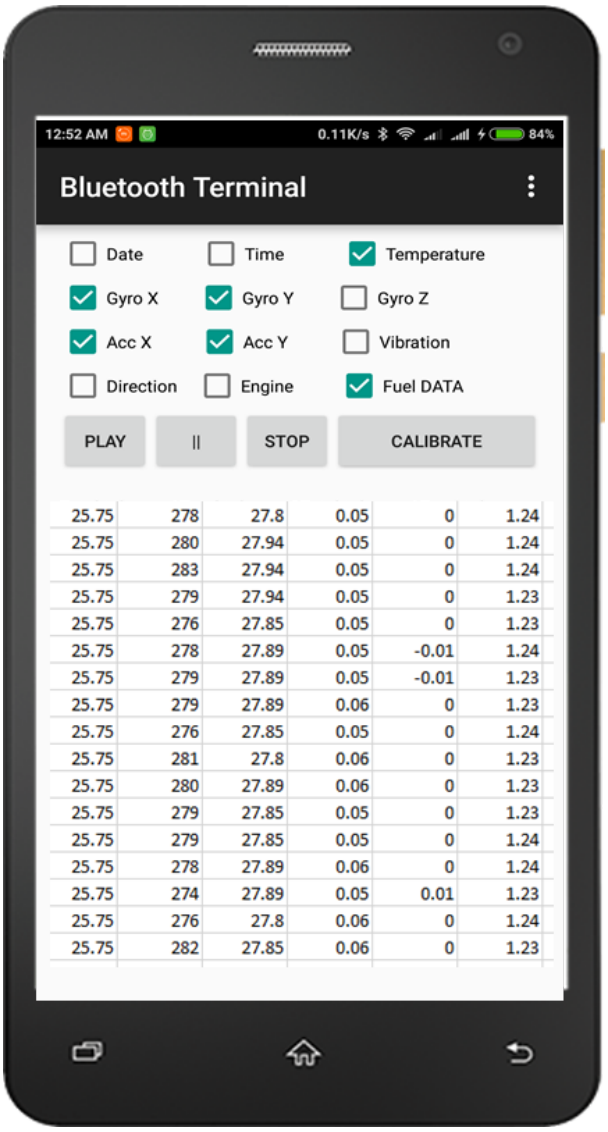

Figure 5.7: App To Observe Data In Real Time -

5.3.4 Data Analysis Program

We used MATLAB to plot the raw and filtered datasets and determine correlations among the parameters.

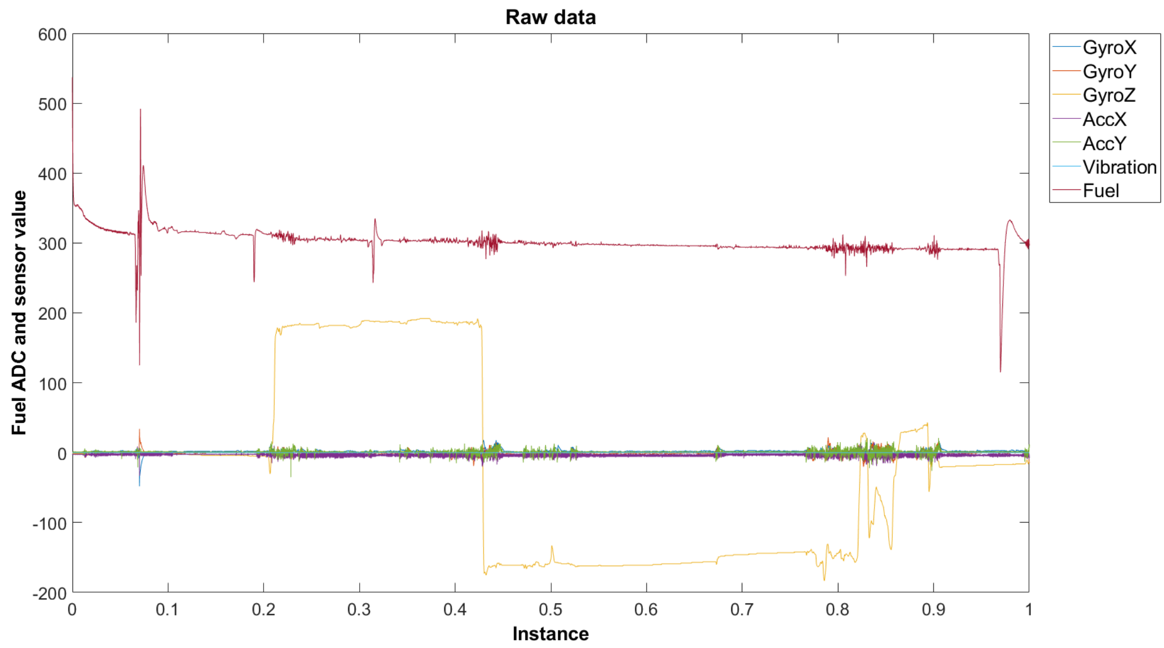

Figure 5.8: Raw Data of Fuel and Sensors

Chapter 6

Experimental Analysis And Results

6.1 Overview

To achieve our goals, we experiment with the datasets in a number of ways, observe the respective outcomes and from them, decide which methods provide the best results. In this chapter, we discuss in detail the experiments we have conducted.

6.2 Attributes and Instances

No of Dataset	15 datasets
No of Instances	20, 000 per dataset
Duration of Dataset	30 minutes to 60 Minutes per dataset
No of Attributes	12 per dataset
Total No of Instances	300, 000 Instances

Table 6.1: Datasets and Instances

1. Date and Time	7. AccX
2. Temperature	8. AccY
3. GyroX	9. AccZ
4. GyroY	10. Vibration
5. GyroZ	11. Engine Status
6. Direction	12. Fuel Data

Table 6.2: Attributes of Dataset

Out of these attributes, we make use of GyroX, GyroY, GyroZ, AccX, AccY, AccZ, Vibration and Fuel Data for our analysis.

6.3 Filtering

Filters are data processing methods used for smoothing out high frequency fluctuations in datasets. To remove unwanted noise and prepare it for further analysis, we used three major types of filtering on the raw data.

6.3.1 Moving Average Filtering

Moving Average filter is a simple Low Pass Finite Impulse Response filter. It's generally used to smooth random noise in data. [6] For a time series $\{y_t\}$, t = 1,...,N, a symmetric moving average filter of window length $2q + 1$ is given by:

$$m = \sum_{j=-q}^{+q} b_j y_{t+j}; \quad q < t < N - q$$

In MATLAB, the filtering process is:

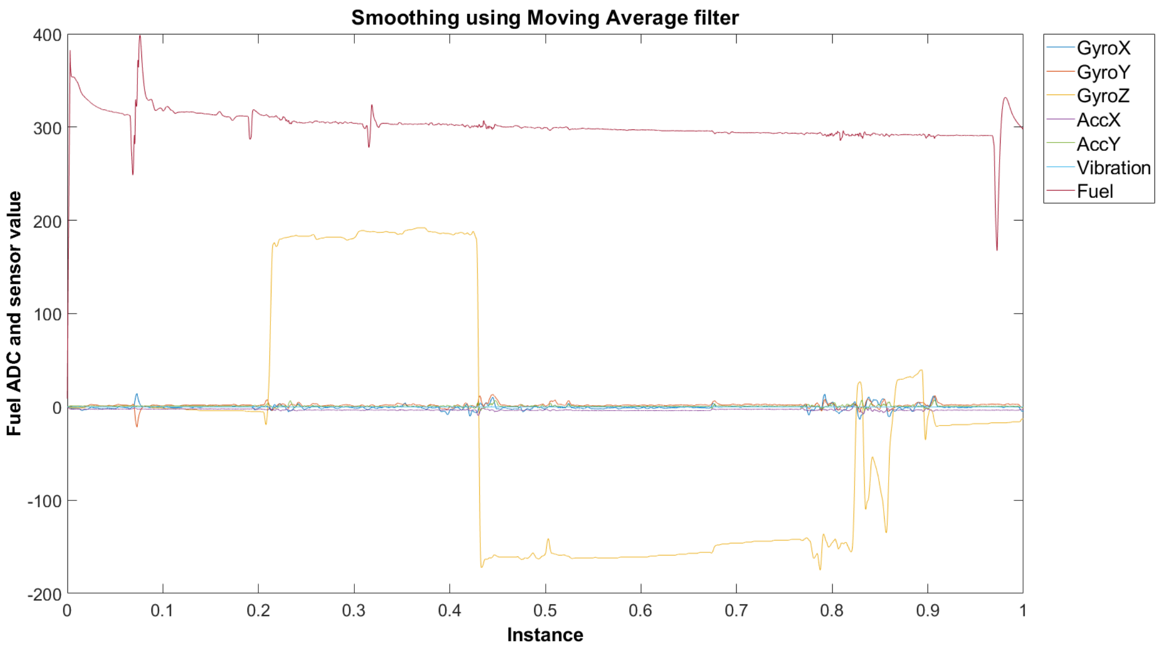

Figure 6.1: Moving Average Filtering- Fuel, Gyroscope, Accelerometer, Vibration Data

6.3.2 Savitzky-Golay Filtering

Savitzky-Golay filter is usually used to smooth a noisy a signal with large frequency window. This filter minimizes least-squares error when fitting a polynomial to frames of noisy data. [7] The data consists of a set of n $\{x_j, y_j\}$ points (j = 1, ..., n), where x is an independent variable and y is an observed value. They are treated with a set of m convolution coefficients, $\{C_i\}$, according to the expression:

$$Y_j = \sum_{i=\frac{1-m}{2}}^{\frac{m-1}{2}} C_i\, y_{j+i}, \qquad \frac{m-1}{2} \leq j \leq n - \frac{m-1}{2}$$

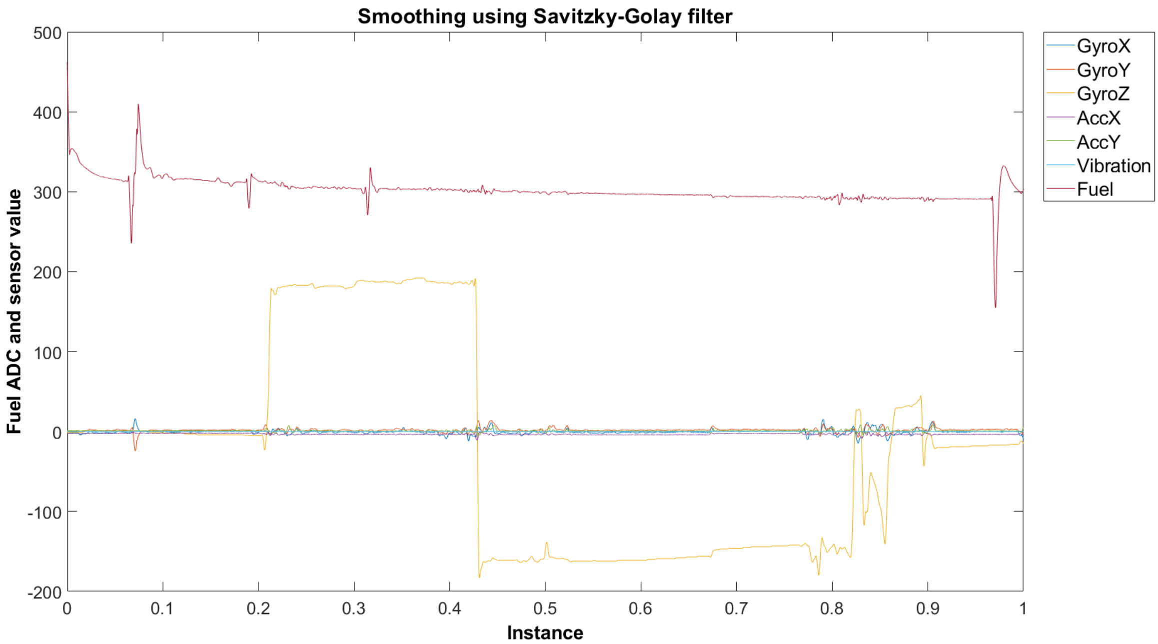

Figure 6.2: Savitzky-Golay Filtering- Fuel, Gyroscope, Accelerometer and Vibration Data

6.3.3 Local Regression Smoothing

Local Regression or Moving Regression is a generalized form of polynomial regression and moving average. [8] Local Regression computes weights using this expression:

$$w_i = (1 - |\frac{x - x_i}{d(x)}|^3)^3$$

The most used techniques of local regression are LOESS (Locally Estimated Scatterplot Smoothing) and LOWESS (Locally Weighted Scatter plot Smoothing). For datasets with outliers, robust versions of these methods can be used. [9] The robust versions use the function below to calculate robust weights for the datapoints:

$$w_i = \begin{cases} (1 - (\frac{r_i}{6MAD})^2)^2, & |r_i| < 6MAD \\ 0, & |r_i| > 6MAD \end{cases}$$

We will apply local regression techniques and their robust versions on the datasets in the subsequent sections and compare the results.

6.3.3.1 LOWESS (Locally Weighted Scatter plot Smoothing)

LOWESS is a non-parametric method that uses a weighted linear polynomial to smooth datapoints of a dataset.

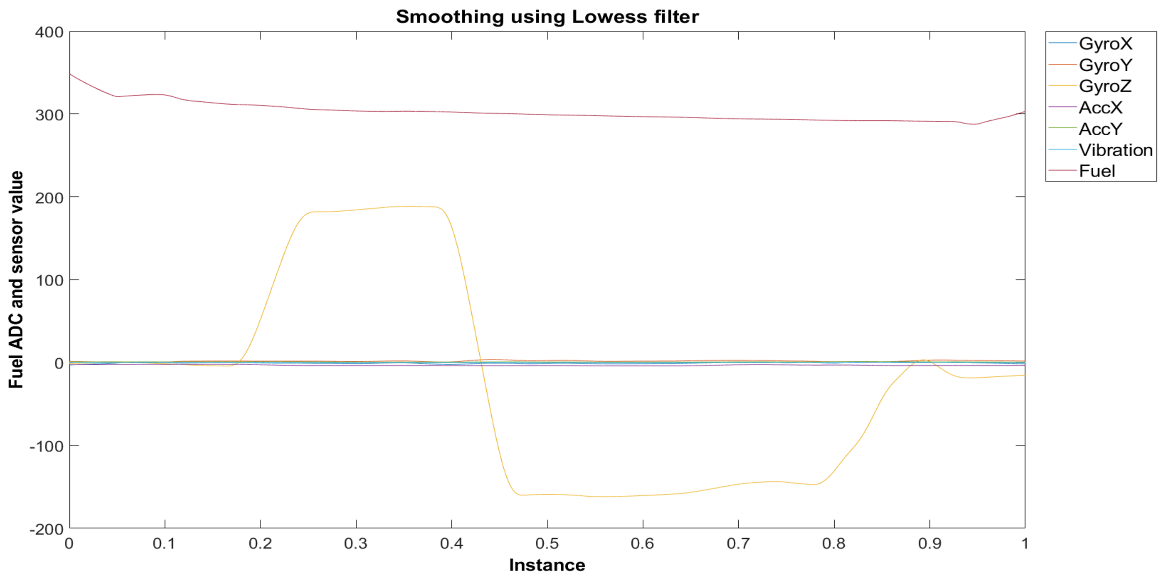

Figure 6.3: LOWESS Filtering- Fuel, Gyroscope, Accelerometer and Vibration Data

6.3.3.2 Robust LOWESS

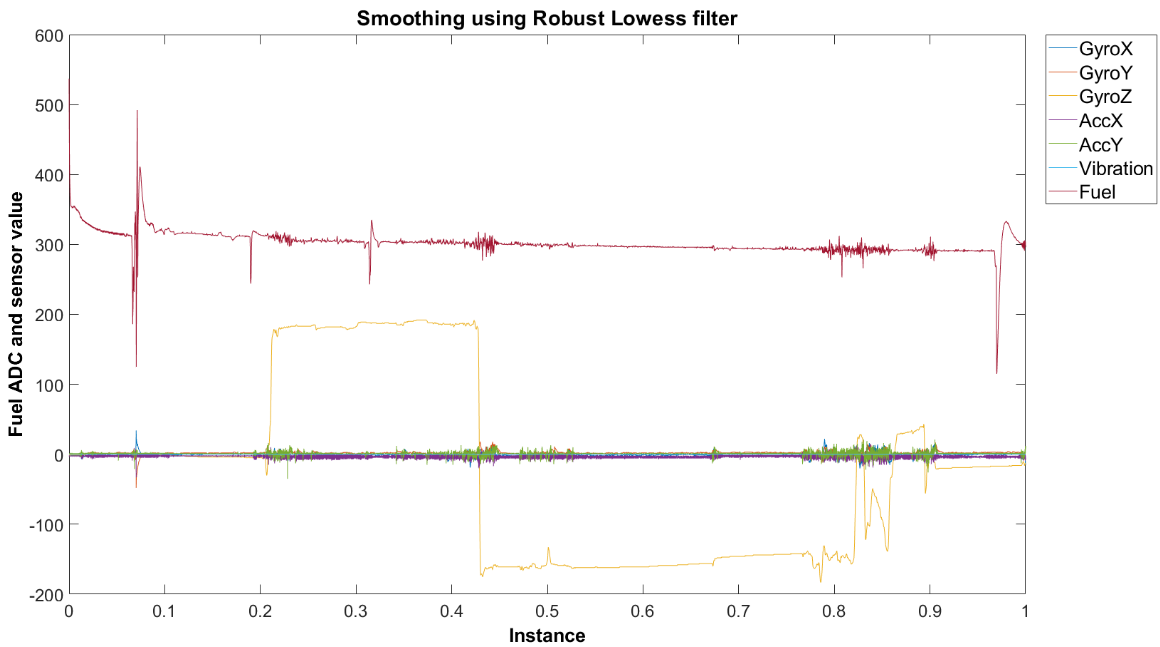

Figure 6.4: Robust LOWESS Filtering- Fuel, Gyroscope, Accelerometer, Vibration Data

6.3.3.3 LOESS (Locally Estimated Scatterplot Smoothing)

LOESS is a non-parametric method that uses a weighted quadratic polynomial to smooth datapoints of a dataset.

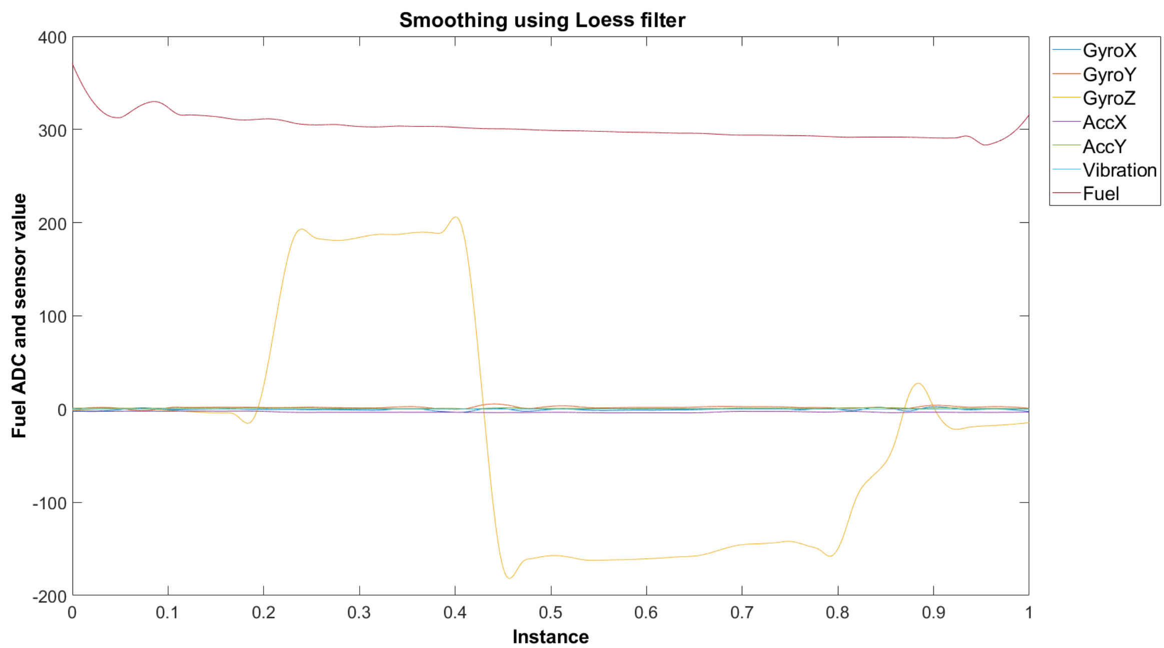

Figure 6.5: LOESS Filtering- Fuel, Gyroscope, Accelerometer and Vibration Data

6.3.3.4 Robust LOESS

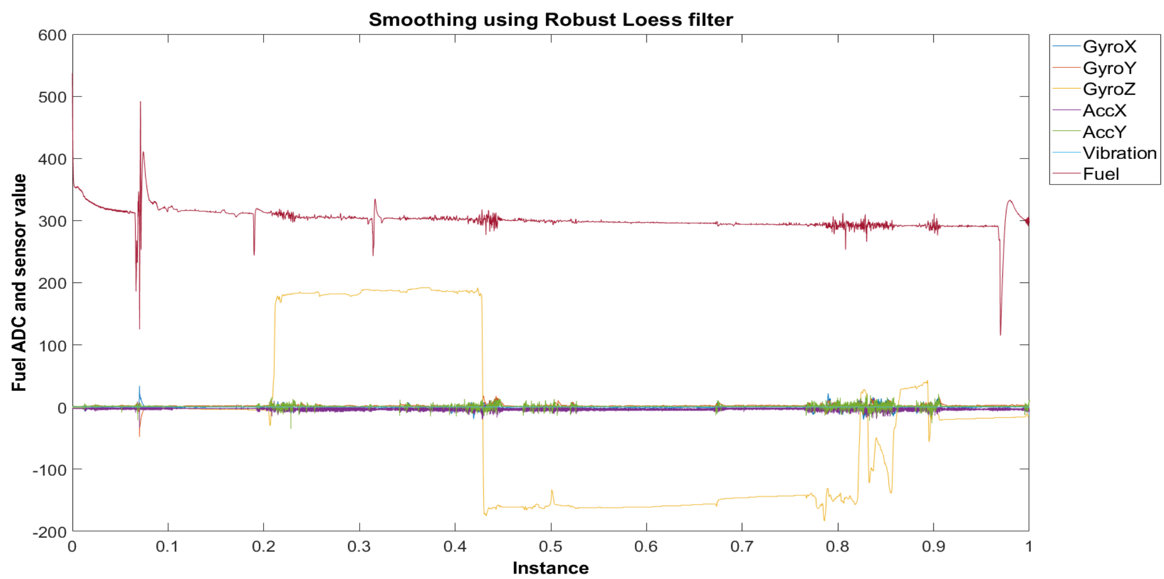

Figure 6.6: Robust LOESS Filtering- Fuel, Gyroscope, Accelerometer, Vibration Data

6.4 Analyzing Filtering Results

Studying all these graphs, we can say, that LOWESS and LOESS methods provide the best results. These results are almost similar, therefore, we decide to use these methods for next steps.

6.5 Correlation Observation

From our collected data of 300, 000 instances, we have plotted the fuel data against each of GyroX, GyroY, GyroZ, AccX, AccY, AccZ and Vibration data. The effect of motion parameters on Fuel data that we found out from the observations are listed below:

- Vibration sensor data has no effect
- accX, accY and accZ data has little to no significant effect
- gyroY has a very significant effect, gyroX has little to none and gyroZ has none.

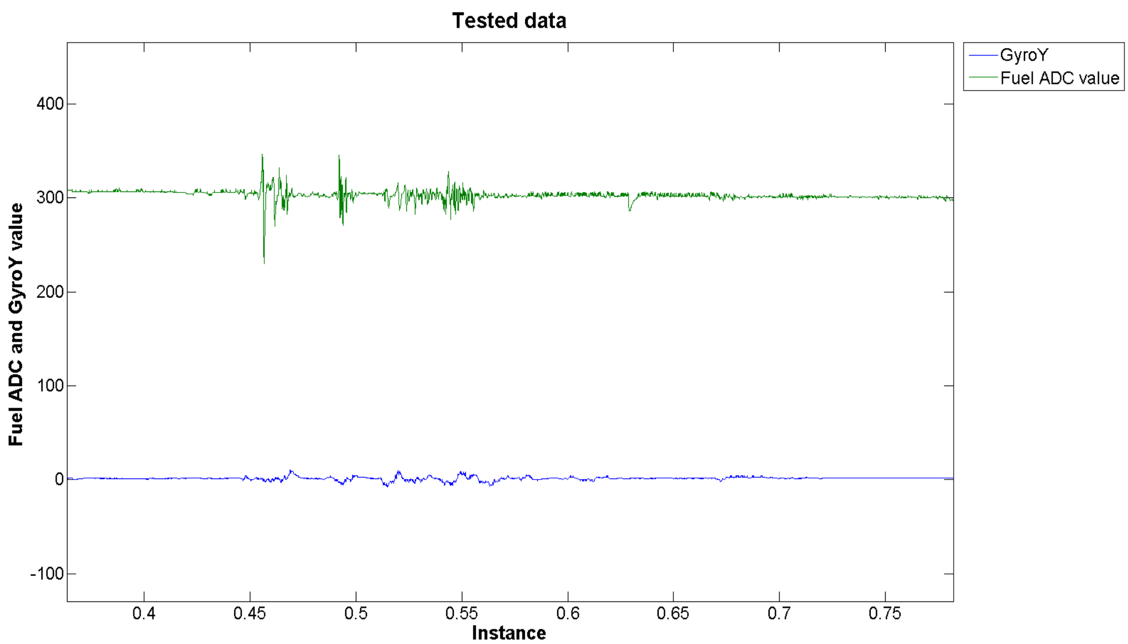

Figure 6.7: Effect of GyroY on Fuel ADC Value

In this graph, we can see that ADC Value fluctuates in the places where GyroY value fluctuates. This observation stays consistant through the other datasets.

6.5.1 Correlation Equation

Using the filtered data, we derive a correlation equation from MATLAB to determine fuel ADC value from the value of gyroY:

$$Fuel(computed) = 0.22061 \times (gyroY)^2 - 6.5145 \times (gyroY) \\ +ADC \quad value \times 0.7 + 90$$

We have taken 70% of the previous ADC value is this equation, to make sure the graph is consistant with the actual ADC values and does not become distorted. We decided on the constant value 90 by testing different values for each dataset, observing the effects of these values and taking a average of those that provide the best results.

This equation gives accurate ADC values eliminating the noise originated from the orientation of the car along the Y axis. The results from applying this equation to the datasets are given below:

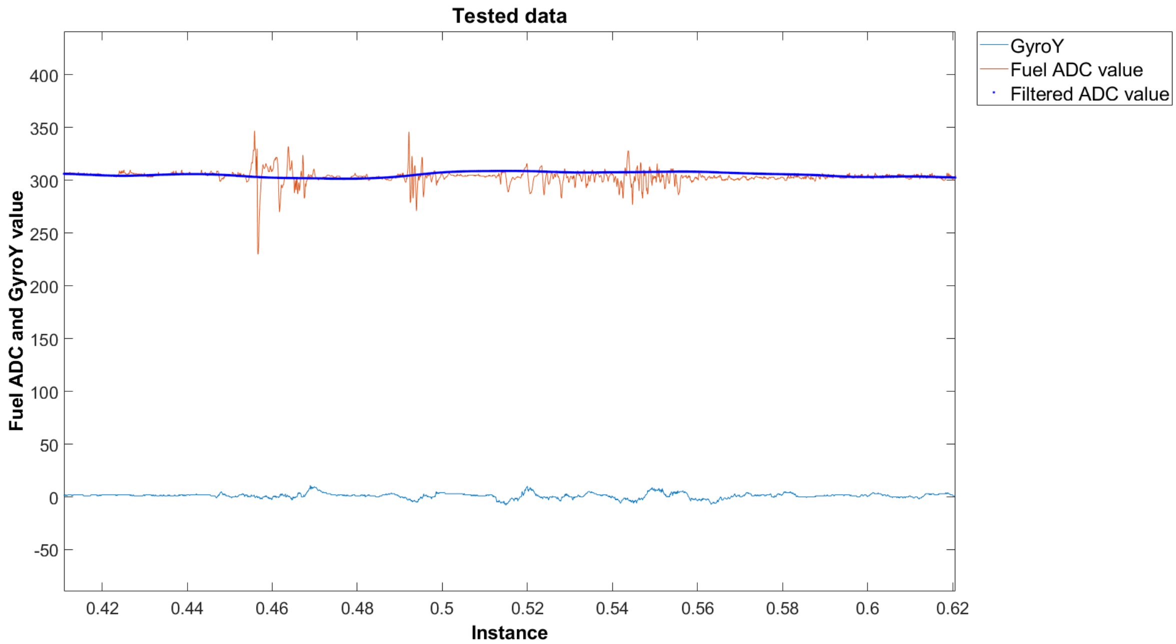

Figure 6.8: Fuel ADC Value Computed From Correlation Equation: Example- 1

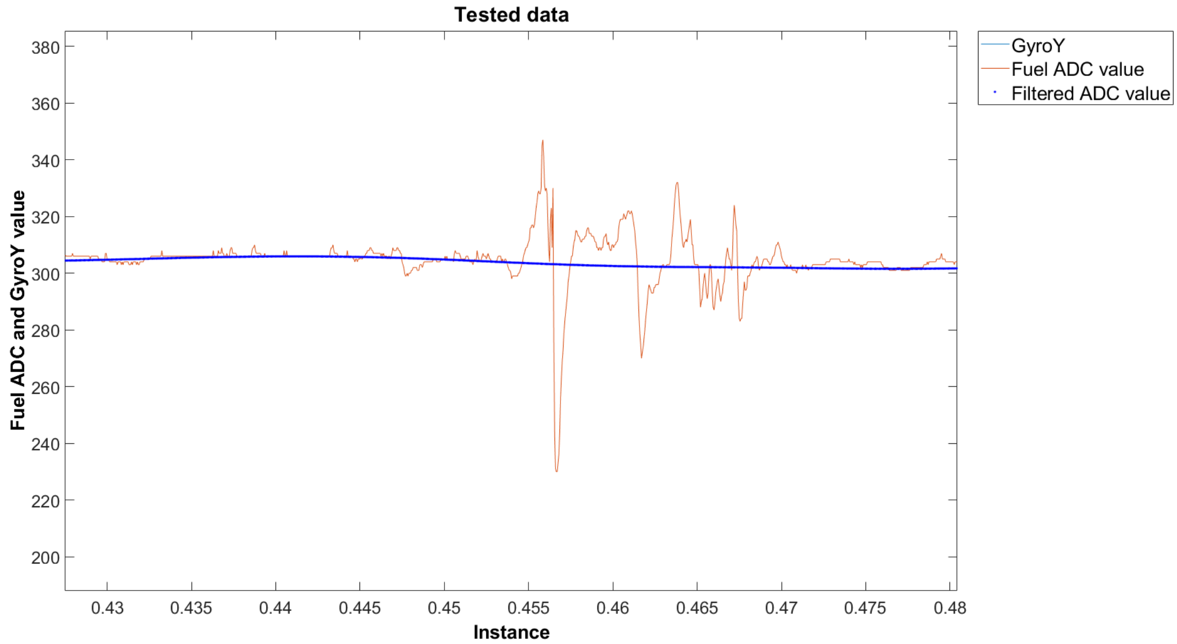

Figure 6.9: Fuel ADC Value Computed From Correlation Equation: Example- 2

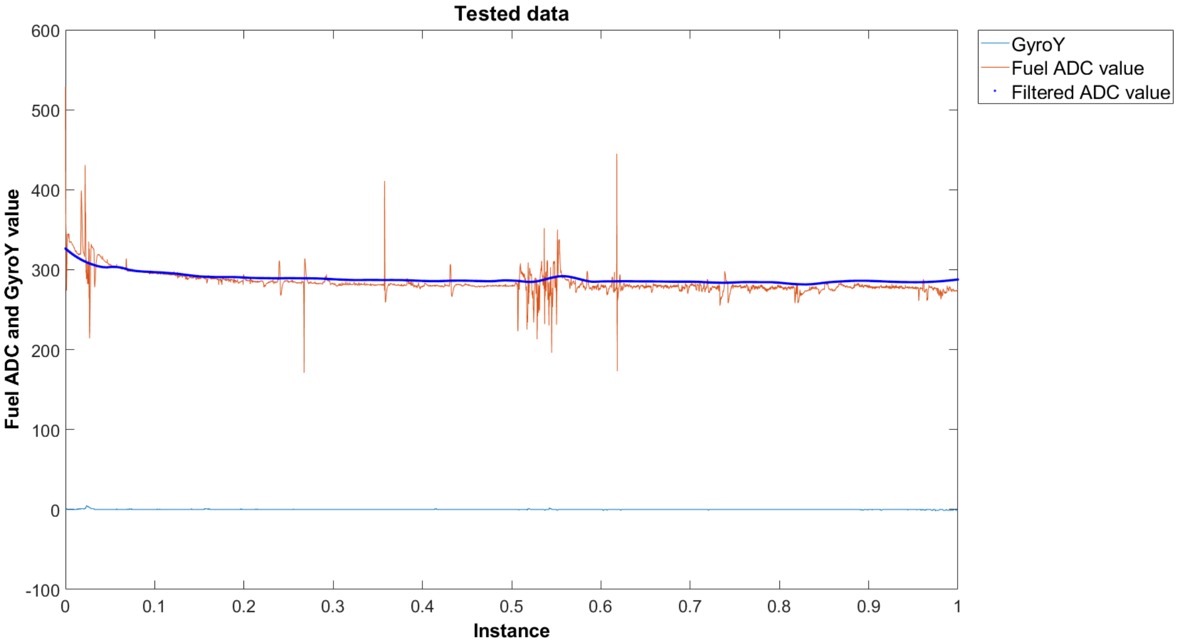

Figure 6.10: Fuel ADC Value Computed From Correlation Equation: Example- 3

6.6 ADC Ratio

We have calculated the ADC Ratio this way:

$$ADC \quad Ratio = \frac{old \quad ADC \quad value}{new \quad ADC \quad value}$$

while a vehicle is moving, the fuel level gradually goes down. So in normal circumstances, new ADC value will be lower than the old one. Studying all the datasets, we find that the ADC ratio stays within the range **[1 - 1.05]** during normal movement of a vehicle.

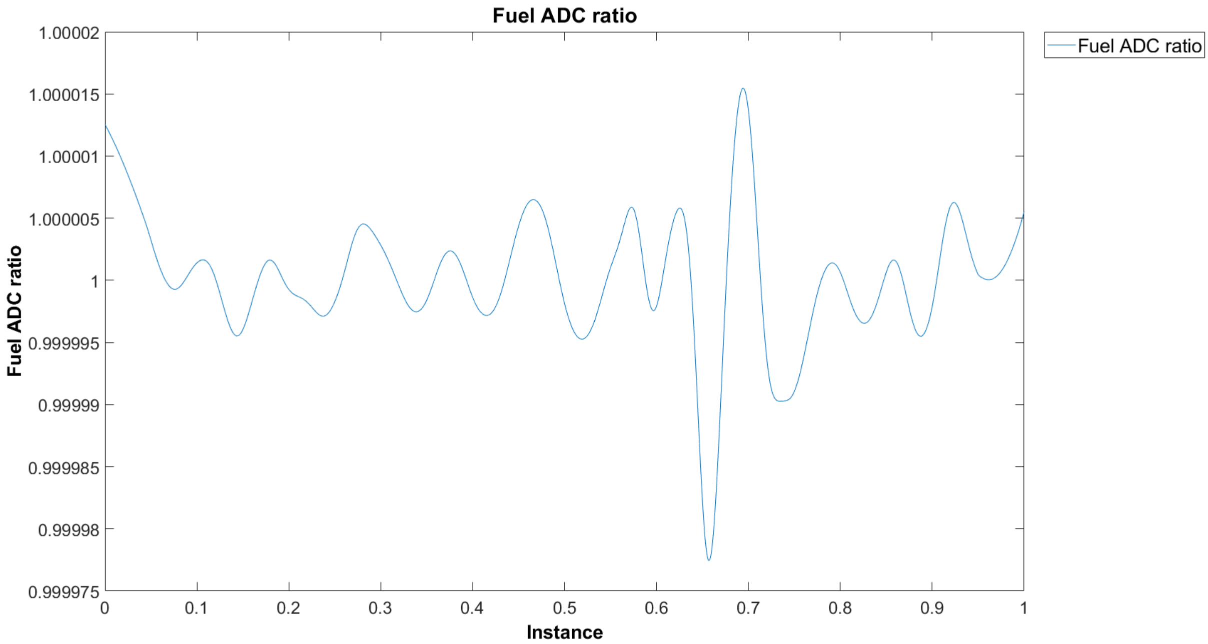

Figure 6.11: Fuel ADC Ratio

6.7 Unusual Activity Detection from ADC Ratio

Using the ADC ratio, we can detect unusual activity of a vehicle. If the ratio rises above the usual range of **1 - 1.05** consistently for a large number of instances, that would mean new ADC value is much lower than the old value than it should be in normal cases. Therefore, this behavior is unusual- meaning it could be fuel leak or theft, which causes sudden decrease in fuel level and consequent drop in the respective ADC value.

$$ADC \quad Ratio = \frac{old \quad ADC \quad value}{new \quad ADC \quad value} > [1 - 1.05]$$

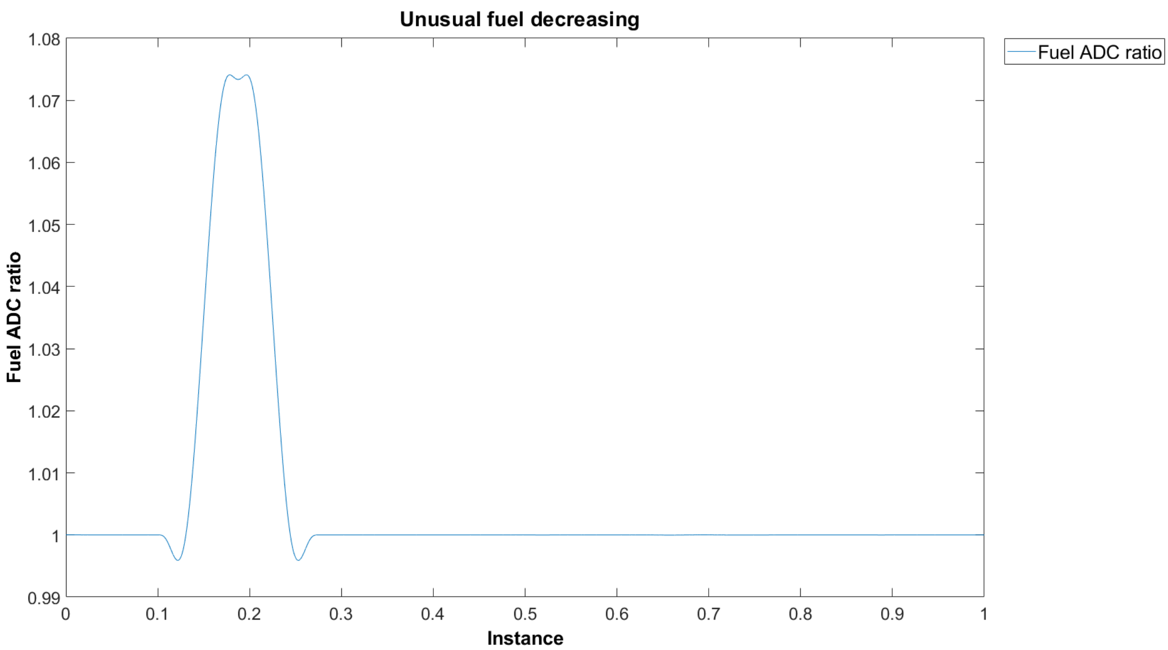

Figure 6.12: Unusual Increase In Fuel ADC Ratio Indicating Suspicious Incident

6.8 Fuel Refill Detection from ADC Ratio

We can detect fuel refill from the ADC ration as well. If the ratio falls below the usual range of **1 - 1.05** consistently for a large number of instances, that would mean new ADC value is much higher than the old value than it should be in normal cases. This behavior can happen when there is a sudden increase in the fuel level and consequent boost in the respective ADC value, meaning fuel tank has been refilled.

$$ADC \quad Ratio = \frac{old \quad ADC \quad value}{new \quad ADC \quad value} < [1 - 1.05]$$

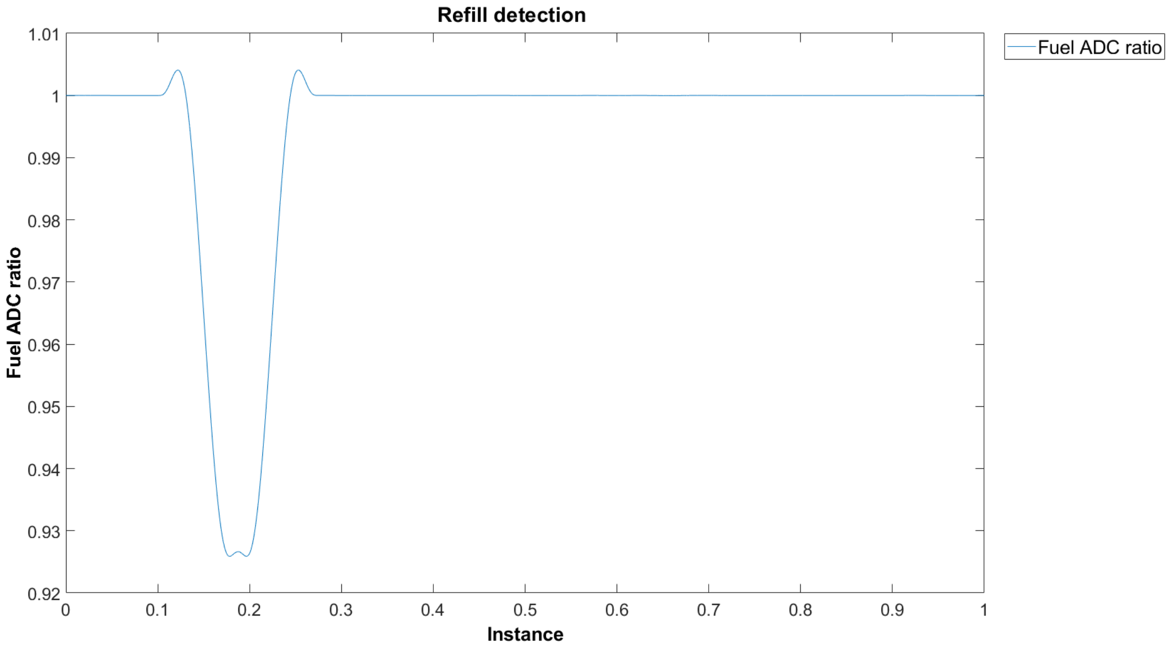

Figure 6.13: Refill Detection Through Drop In Fuel ADC Ratio

6.9 Accuracy Analysis

Out of 15 datasets containing 300, 000 instances, we have found out the correct ADC value and detected suspicious behavior and refill incidents for 12 datasets. Our success rate is therefore 80%.

In the remaining datasets, the computed ADC values differ from the original value by upto 30 points. Some examples are given below:

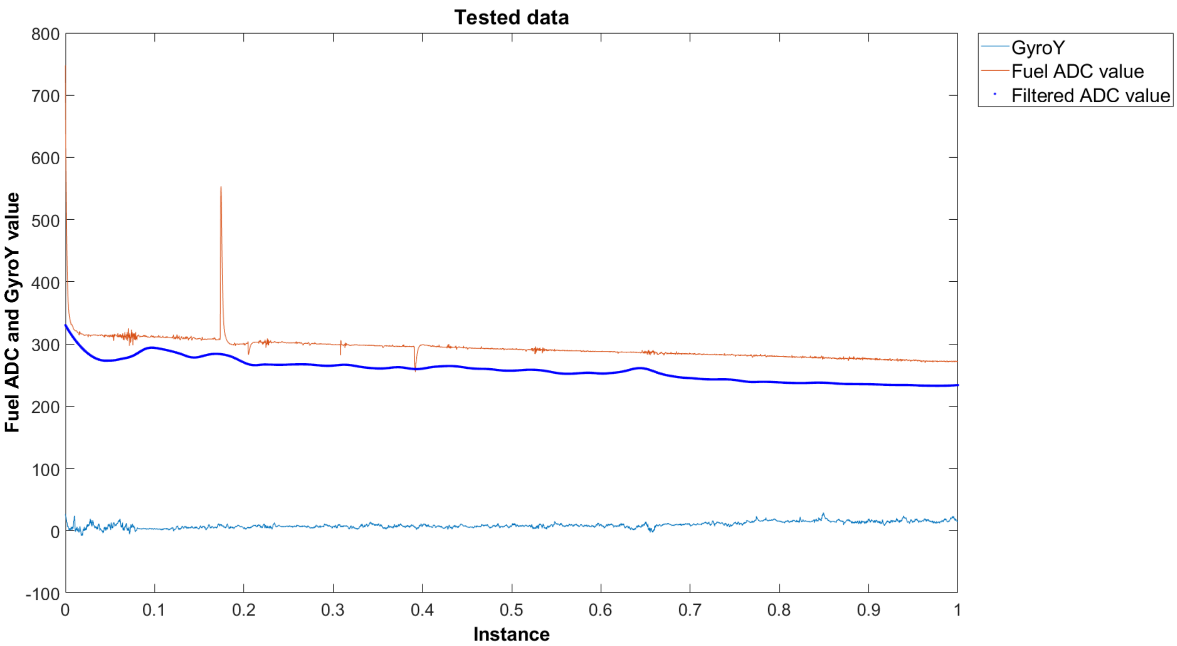

Figure 6.14: Incorrect Result From Fuel ADC Value Computation: Example- 1

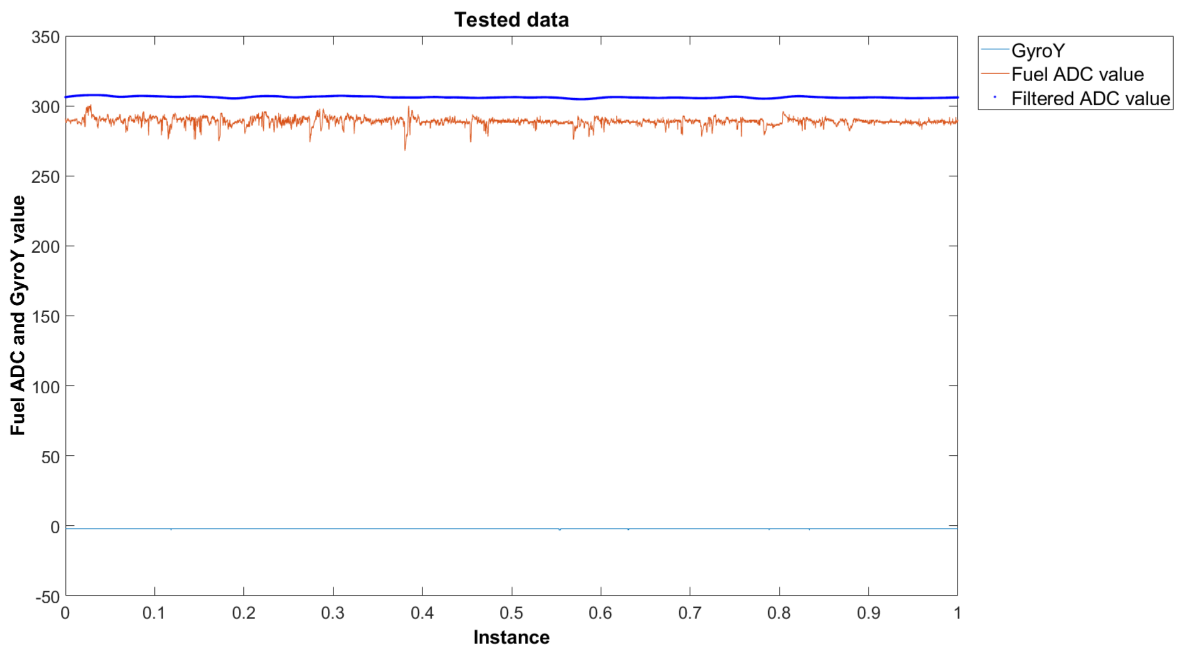

Figure 6.15: Incorrect Result From Fuel ADC Value Computation: Example- 2

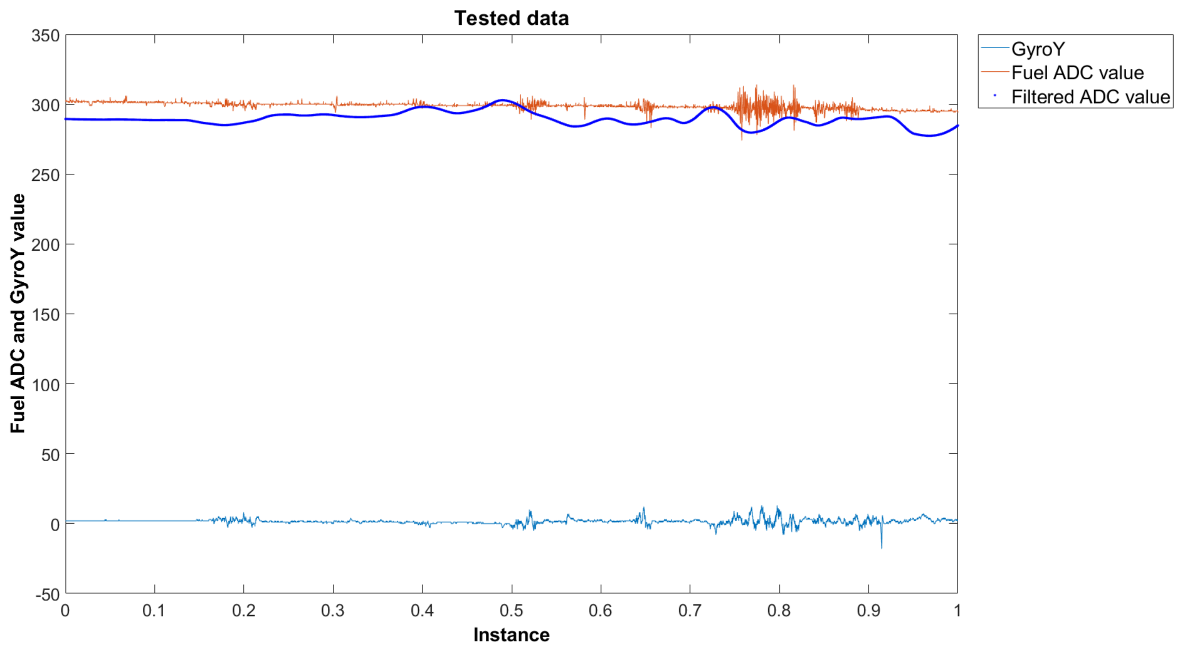

Figure 6.16: Incorrect Result From Fuel ADC Value Computation: Example- 3

Chapter 7
Standards

7.1 Introduction

In the era of information, gaining legally permitted model and program is highly significant. We have to conform to certain laws, policies and regulations to conduct the project and make sure our final prototype maintains all necessary standards properly.

7.2 Compliance With standards

7.2.1 Technology

Programming Language	C/C++
Documentation Tool	LaTeX, Microsoft Word
IDE	Arduino, MATLAB, Proteus Design Suite, Android Studio
Data Formats	Comma Separated Value (CSV)

Table 7.1: Technology Standards

7.2.2 Ethics

ISO/IEC 12207 - International Organization for Standardization and International Electrotechnical Commission combinedly published this standard for software engineers. This standard promotes not to develop any software which may harm mankind, society and environment economically, physically or in any other form.

7.2.3 Licenses

7.2.3.1 License for Device

Pi Labs Bangladesh Ltd. they have a license from BRTA for using VTS device, that also extends to our ptototype:

BTRC/LL/Vehicle
Tracking(16) Pilabs/2011-13
Dated: 06/02/2012

7.2.3.2 License for Programming Languages

We have used C, Arduino and MATLAB programming languages in our project. These languages are licensed as mentioned below:

- **C:** Open Source License
- **Java:** Open Source License
- **Arduino:** Creative Commons Attribution-Share Alike 3.0 License, GNU GPL (General Public license), GNU LGPL (General Public license)
- **Matlab:** Creative Commons Attribution-Share Alike 3.0 License, BSD License.

7.2.3.3 License for IDE

We have used Arduino and MATLAB IDE and they are licensed as mentioned below:

- **Arduino:** GNU GPL (General Public license)
- **Matlab:** Commercial License.
- **Proteius Design Suit:** Commercial License.
- **Android Studio:** Commercial License.

7.2.3.4 License for Documentation Tools

We have used Arduino and MATLAB IDE and they are licensed as mentioned below:

- **LaTeX:** AGPL (Affero General Public license)
- **Microsoft Word:** Commercial License.

7.2.4 Safety

Our prototype is built using off-the-shelf modules and there is no explosive or harmful material that could pose any threat to the users. The prototype is therefore safe to use.

Chapter 8
Impacts And Constraints

8.1 Impacts

8.1.1 Economic Impact

Precisely calculated data will allow the users to track their usage behavior, therefore will detect fuel theft, refill and unauthorized usage of vehicle by drivers. On one hand, this will eliminate economic loss of the users while on the other hand, it may increase the salary of the drivers as any unlawful income regarding the fuel usage will diminish.

8.1.2 Social Impact

By helping to reduce fuel theft and unauthorized use of vehicles, the project will influence positive social values among drivers, renters, fleet management companies, refilling stations and many other related communities.

8.1.3 Environmental Impact

Since overall report will encourage the users to implement cost and fuel effective utilization of vehicles, unnecessary fuel consumption may go down and help curb wastage of resources as well as pollution.

8.1.4 Ethical Impact

This project will greatly enforce ethics in use of fuel and so will uplift individual and collective sense of moral.

8.2 Constraints

8.2.1 Political Constraint

We do not have to deal with any political constraint. Pi Labs Bangladesh Ltd. already owns all the necessary permits to implement the design and algorithm.

8.2.2 Real Time Response

For successful observation, the prototype needs real time data from the vehicle. The RTC module (real time clock) ensures this constraint can be met.

8.2.3 Manufacturability

Our prototype is easy to produce, cost friendly and small in size. So the hardware and software both can be produced in large numbers with ease.

8.2.4 Portability

As vehicles have limited space to spare, we have designed the prototype to be compact and lightweight. It can be plugged into the float sensor line very conveniently and removed just as easily when necessary.

8.2.5 Sustainability

Since we have designed the model considering accidental cases, the design is sustainable and durable.

8.2.6 Cost Effectiveness

Our prototype is meant to be a business solution to improve existing systems. Therefore, we have to keep in mind that the solution does not impose heavy costs and cut down on the profit. We have used low cost quality materials that serve both our purpose and the business perspective.

Chapter 9
Conclusion, Limitation And Future Work

9.1 Conclusion

Fuel Monitoring is a crucial part of Vehicle tracking systems. Therefore, it's imperative that the current services provided by the local companies are improved much further to avoid loss of financial and natural resources. In this project, we have worked on a novel model that could contribute significantly in improving the accuracy of fuel level monitoring. We have-

- proved a strong correlation between fuel ADC value and the orientation of a vehicle and derived a correlation equation to determine the a more accurate ADC value using the orientation value.
- detected suspicious consumption events such as fuel stealth, leak etc. in real time. In such cases, the ADC ratio of the previous and current ADC values rises above the usual range [**1 - 1.05**].
- detected fuel tank refill incidents in real time. In this case, the ADC ratio falls below the usual range [**1 - 1.05**].

9.2 Limitation

Our research yields results in terms of the ADC value provided by the fuel level sensor. It does not provide the results in terms of Fuel level units such as litres. For measuring fuel in liters, we have to install our device in a car and measure the ADC value when refilling and the emptying the tank and have to repeat this process a few times for calibration. However, to install the device, we have to take apart the dashboard of a car, cut open the float sensor cable and connect it to our device. For this reason, it was not possible to loan a car and unfortunately none of our teammates own a car either.

9.3 Future Work

With more datasets relevant to the unusual consumption scenarios such as leak, stealth or refill, further qualitative analysis could be done to overcome the problems with the current detection model. Also, in the cases when the ADC ratio fluctuates too much because of long time travel on uneven paths such as rocky roads, mountain trails, a more suitable analysis model could be designed with proper datasets of a vehicle's journey through such scenarios.

Bibliography

[1] S. Kanarachos, J. Mathew, and M. Fitzpatrick, "Instantaneous vehicle fuel consumption estimation using smartphones and recurrent neural networks," *Expert Systems with Applications*, vol. 120, 12 2018. 4

[2] R. Bose and S. Roy, "Explaining the workings of a real-time fuel monitoring system," 07 2018. 4

[3] S. Abd Elmonem Yosif, M. Mohamed Abdelwahab, M. Abd Elrahman ALagab, and F. Muhammad, "Design of bus tracking and fuel monitoring system," 01 2017, pp. 1–5. 4

[4] A. Grcan apraz, P. zel, M. evkli, and O. Beyca, "Fuel consumption models applied to automobiles using real-time data: A comparison of statistical models," *Procedia Computer Science*, vol. 83, pp. 774–781, 12 2016. 4

[5] S. S. Aher and K. R.D., "Fuel monitoring and vehicle tracking," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 1, 03 2012. 4

[6] "Moving average filter," http://www.mathworks.com/help/econ/filtering.html, accessed: 2019-05-24. 24

[7] "Savitzky-golay filter," https://en.wikipedia.org/wiki/Savitzky-Golay_filter, accessed: 2019-05-24. 25

[8] "Local regression smoothing," https://www.mathworks.com/help/curvefit/smoothing-data.html#bq_6ys3-3, accessed: 2019-05-24. 26

[9] "Robust local regression," https://www.mathworks.com/help/curvefit/smoothing-data.html#bq_6ys3-3, accessed: 2019-05-24. 26

Appendix A
Project Timeline

A.1 Timeline of Trimester 1

Week 3 Submit Project Proposal.

Week 4 Develop a working model and gather relevant research papers.

Week 5 Collect dataset.

Week 6 Test the dataset against the model.

Week 7 Test larger dataset.

Week 8 Repeat testing with different models.

Week 9 Adopt the best working model.

Week 10 Complete model testing.

A.2 Timeline of Trimester 2

Week 3 Make adjustment to the module.

Week 4 Collect dataset.

Week 5 Design PCB.

Week 6 Install new compact module.

Week 7 Collect Dataset and start analysis.

Week 8 Plot and Filter Dataset.

Week 8 Test the dataset against different methods and compare the results.

Week 9 Find our anomaly, limitations and ways to reduce them.

Week 10 Final goal testing and project completed.

Appendix B

Work Division

Name	Responsibility
Md. Miraz Hossain	Design and interface hardware device.
	Test and debug hardware device.
	Develop Android App for run-time observation.
Mohammad Maruf Imtiaze	Develop algorithm to analyze data sets.
	Test different models against data sets.
	Prepare presentations.
Md. Tafsir Hossain	Market research and benchmark study.
	Test different models against dataset.
	Prepare presentations.
Faizah Farzana	Develop algorithm to analyze data sets.
	Test different models against dataset.
	Prepare reports and posters.

Table B.1: Work Distribution