A complete hardware/software application for the characterization of car driving behaviours

Rodolfo Borges Dutra

Thesis to obtain the Master of Science Degree in Aerospace Engineering

Supervisors: Prof. Alexandre José Malheiro Bernardino
Eng. Carlos Alberto Fernandes Arantes

Examination Committee

Chairperson: Prof. José Fernando Alves da Silva
Supervisor: Prof. Alexandre José Malheiro Bernardino
Member of the Committee: Prof. Bertinho Manuel D’Andrade da Costa

November 2017
Acknowledgments

First and foremost, I would like to thank my supervisors, Engineer Carlos Arantes and Professor Alexandre Bernardino for their support, expertise, and advice on the development of this thesis. I am very grateful for their guidance and value brought to the work.

I express my sincere appreciation to the whole CEiiA community, for receiving me as one of their own. Special thanks to the Academia CEiiA for providing the opportunity and all the work conditions needed to focus and achieve my objectives.

To all my colleagues of the Mobility Team, for sharing the workplace and their challenges with me, especially my desk neighbour Tiago Macedo. I look forward to more opportunities to work with you all.

To the Engineers José Silva and Carlos Arantes, for the inspiration and motivation to work even harder.

To my internship colleagues, Pedro Reguenga, Carlos Lopes and Miguel Leal, for all the encouragement given and for sharing their experience.

To my Técnico Lisboa friends, Tiago Nunes, Rui Pires, Vítor Mendonça and João Filipe, that accompanied my whole university degree. For all the hard work and deadlines completed with the perseverance required, for all the good and bad times I am grateful I could share them all with you.

To my friends, Elia Niemczyk, Jamil Heneni and Diogo Braga, for hearing my ideas, giving always their sincere advice and encouraging my goals.

To my old friends, Isaac Vargas, Gustavo Garcia and Karina Silveira.

To my parents, grandparents, and brother, for their unconditional support and love. They gave me the strength and happiness I needed to pursue the degree and thesis, independent of every setback.

To each and every one of you - Thank you.
Abstract

According to the World Health Organization, road safety is a global problem. In 2015, there were 1.25 millions of traffic related deaths and, with these accidents, billions of associated costs on each country. One of the main reasons are the behaviours of drivers, which impact, not only the vehicle integrity, but also the fuel consumption and CO₂ emissions. Therefore, a driver characterization solution could bring significant value to insurance companies as well as car sharing services like the one proposed by CEiiA, included in the MOBI.ME platform, to reduce emissions.

An embedded system was developed to collect data from the behaviour of drivers. The PCB, designed with Autodesk® Eagle™, was projected to contain a microcontroller, programmed to save the vehicle acceleration, from an accelerometer at 40Hz and data from a GPS module at 1Hz, to a µSD card. Using Python, the acceleration was filtered by a low-pass Butterworth Filter, and the data from four drivers were labelled into seven different maneuvers. Four maneuver classification models were built using a Linear SVM and three non-linear machine learning algorithms, K-NN, Random Forest and SVM with RBF kernel. The SVM, with RBF kernel, obtained the best performance with 89% accuracy and an F1-score of 75%.

The solution for driver characterization was finally presented in the form of a website, developed with Django, with a Google Maps API. It is possible to observe, not only the number of times each maneuver was detected but, as well, which levels of acceleration, were more common. Based on this metrics it is possible to observe different behaviours that, with more data, will give a clearer characterization of each unique driver.

Keywords

Driving Behavior; Embedded System; Printed Circuit Board; Data Handling; Machine Learning.
Resumo

De acordo com a Organização Mundial de Saúde, a segurança na estrada é um problema global. Em 2015, houveram 1.25 milhões de mortes na estrada e, com estes acidentes, bilhões de custos em cada país. A principal razão é o comportamento dos condutores, que influenciam, não só a integridade do veículo, como também o consumo de combustível e emissões de CO₂. Portanto, uma solução de caracterização de condutores poderá trazer valor a seguradores bem como serviços de "car sharing", como o proposto pelo CEiiA, incluído na plataforma MOBI.ME, para redução de emissões.

Um Sistema Embebido foi desenvolvido para reunir dados do estilo de condução. Um PCB, desenvolvido com Autodesk® Eagle™, foi projetado para conter um microcontrolador, programado para guardar num cartão µSD a aceleração, de um acelerômetro a 40Hz, e a velocidade e posição de um módulo GPS a 1Hz. Usando Python, filtrou-se a aceleração com um filtro Butterworth passa-baixo, e os dados de quatro condutores foram etiquetados em sete manobras diferentes. Foram construídos quatro modelos de classificação usando um SVM Linear e três algoritmos não-lineares, K-NN, Random Forest e SVM com kernel RBF. O SVM, com kernel RBF, obteve a melhor performance com 89% precisão e 75% F1-score.

A caracterização de condutores foi completada com um website desenvolvido em Django, com Google Maps API. Nele observa-se, não apenas o número de detecções de cada manobra, mas também que níveis de aceleração são mais comuns. Baseado nestas métricas é possível observar diferentes comportamentos que, com mais data, conseguirão caracterizar condutores com maior clareza.

Palavras Chave

Estilo de Condutores; Sistemas Embebidos; Placa de Circuito Impresso; Eagle; Análise de Dados; Machine Learning.
## Contents

1 Introduction ........................................ 1
   1.1 Motivation ........................................ 3
   1.2 CEIIA-Centre of Engineering and Product Development ........................................ 4
   1.3 Related Work .................................... 5
      1.3.1 DriveSafe: an App for Alerting Inattentive Drivers and Scoring Driving Behaviors ........................................ 5
      1.3.2 Mobile Phone Based Drunk Driving Detection ........................................ 6
      1.3.3 Driving Scoring for Fleet Management and Insurance Applications using Fuzzy Logic ........................................ 6
      1.3.4 Driving Style Recognition with Dynamic Time Wrapping ........................................ 7
      1.3.5 Hidden Markov Method for detection of Driving Behavior ........................................ 7
      1.3.6 SVM in driver identification ........................................ 8
      1.3.7 Deep Learning in driver identification ........................................ 9
   1.4 Objectives and Innovation .......................... 9
   1.5 Context of this work in Aeronautic Industry ........................................ 10
   1.6 Thesis Structure .................................. 10

2 Embedded System Concepts ................. 11
   2.1 Microcontroller .................................. 13
   2.2 Accelerometer ................................... 15
   2.3 GPS Module ..................................... 17
      2.3.1 NMEA 0183 Standard ........................................ 19
   2.4 Low-Level Communication Protocols .................. 19
      2.4.1 Universal Asynchronous Receiver/Transmitter (UART) Protocol ........................................ 20
      2.4.2 Serial Peripheral Interface (SPI) Protocol ........................................ 21
      2.4.3 Inter-Integrated Circuit (I²C) Protocol ........................................ 21

3 PCB Design .......................................... 23
   3.1 Autodesk® EAGLE™8 ........................................ 25
   3.2 Schematic ......................................... 26
      3.2.1 Power Supply ........................................ 27
3.2.2 Level Shifthing .................................................. 27
3.2.3 LED’s .............................................................. 28
3.3 Board ................................................................. 28
3.3.1 Components Position .......................................... 29
3.4 Manufacturing ..................................................... 30
3.4.1 Gerber Files and Excellon ................................. 30
3.4.2 PCB Quotation .................................................. 30
3.4.3 Components Selection ....................................... 31
3.4.4 Soldering ........................................................ 31

4 System Set-Up ....................................................... 33
4.1 Arduino Setup ...................................................... 35
4.2 Embedded System Setup ........................................ 36
4.2.1 Code Adaptation .............................................. 37
4.3 Installation of the Device in a car ........................... 38

5 Initial Data Analysis ................................................ 39
5.1 Python Language .................................................. 41
5.2 Preliminary Observation ......................................... 41
5.3 Correction and Completion ..................................... 42
5.4 Smoothing .......................................................... 43
5.4.1 Butterworth Filter .......................................... 44
5.4.2 Savitzky-Golay Algorithm .............................. 45
5.4.3 Custom Douglas–Peucker algorithm ................. 46
5.5 Comparison between Butterworth and Savitzky-Golay .................................................. 47

6 Dataset Characteristics and Labeling ...................... 49
6.1 Driver Maneuver Labeling ....................................... 51
6.1.1 E3 Actions and Maneuvers before an Accident .... 51
6.1.2 E4 Complementary information to the actions and maneuvers .... 52
6.1.3 Maneuvers and Labels ..................................... 53
6.2 Dataset Characteristics ........................................ 54
6.3 Data Transformation and Features ........................ 54

7 Machine Learning Algorithms for Maneuver Classification .................................................. 57
7.1 Bias-Variance Trade-Off ........................................ 59
7.1.1 Stratified K-Fold ............................................. 59
7.2 Validation and Learning Curves ............................ 60
**List of Figures**

1.1 Road fatality rates per million population by country, 2014. In Portugal there were over 600 deaths per year on the road. ............................................................. 3
1.2 DriveSafe iPhone GUI. ........................................................................ 6
1.3 Fuzzy Inference System. ..................................................................... 7
1.4 Overview of a framework based on symbolization, recognition and generation of driving pattern. .......................................................... 8

2.1 Simplified scheme of the desired embedded system components and which protocols are needed for the implementation. ........................................... 13
2.2 Atmel 8-bit AVR Microcontroller ATmega328P, PDIP package. .................. 14
2.3 ATmega328P Block Diagram. ................................................................. 14
2.4 Scheme of the inner workings of a semiconductor accelerometer. .................. 16
2.5 Scanning Electron Microscope (SEM) photo of an accelerometer by Analog Devices Inc. ............................................................... 16
2.6 MMA8452Q digital accelerometer (QFN). ................................................. 17
2.7 Gtop.013, GPS module. ....................................................................... 17
2.8 Illustration of the two types of serial communication. .................................. 19
2.9 Data Framing with voltages, start and stop bit of usual **TTL** serial. .......... 20
2.10 SPI configuration with 1 master and 3 slaves. ............................................. 21
2.11 **I²C** configuration with 2 masters and 2 slaves. ....................................... 22
2.12 **I²C** signal protocol for 8bits devices. ................................................... 22

3.1 Layout of a **PCB** with 2 copper layers. ............................................... 25
3.2 EAGLE project example. Schematic window on the left and Board window on the right. ................................................................. 26
3.3 Embedded System Board Top Layer. ...................................................... 28
3.4 Embedded System Board Bottom Layer. ................................................ 29
3.5 Thesis author soldering Through-Hole pins on the developed board. .......... 32
3.6 Final appearance of the projected embedded system. ................................ 32
4.1 Diagram of the flow of the software/firmware implemented in the Atmega328P. Setup and Loop are the same functions present in every Arduino Program. The .csv files are all stored in the µSD card. 36

4.2 Programming the developed system Atmega328P, using an Arduino Board as Programmer. 37

4.3 Demonstration of the alternatives for the allocation of the device in a car. The allocation nº1 has the prototype system, composed by the arduino and the GPS logger shield. 38

5.1 Python Logo. 41

5.2 Sample of raw accelerometer data on the X, Y and Z axis in g units. 43

5.3 X, Y and Z axis in the car. 43

5.4 Frequency Domain of the X axis acceleration data. 44

5.5 Comparison between raw accelerometer data and data filtered by a 9th order Butterworth filter with cutoff frequency of 1Hz. 44

5.6 Comparison between raw accelerometer data and Savitzky-Golay filtered data. 45

5.7 Diagram demonstrating the 5 steps of the Douglas-Peucker algorithm, highlighting, in the last step, the main difference between normal Douglas-Peucker(left) and custom version(right). 46

5.8 Comparison between raw accelerometer data and data filtered with custom algorithm for two different allowable distances. 46

5.9 Comparison between raw accelerometer data filtered by a 9th order Butterworth filter with cutoff frequency of 1 Hz and Savitzky-Golay algorithm 47/2. 47

5.10 Comparison of the magnitude Bode diagram of the 9th order Butterworth filter with cutoff frequency of 1Hz and Savitzky-Golay algorithm 47/2. The green line represents the point where \( f = 1 \text{Hz} \). 48

6.1 Number of drivers that lost their lives in 2016 compared by type of maneuver. 51

6.2 Number of drivers that lost their lives in 2016 compared by complementary information. 52

6.3 Relationships between longitudinal and lateral acceleration with speed during a curve. 53

6.4 Example of classification of with 3 different maneuvers. Shaded regions show the time considered part of the maneuver. 54

6.5 Data Transformation. After transformation, each frame of data will contain 21 features. 9 from a 1s window and the other 12 from an 11s window. 56

7.1 Evolution of generalization error with the evolution of the capacity. 59

7.2 K-Fold Cross Validation with k=10 folds. 59
7.3 Validation Curve showing the Linear Support Vectors Machine model accuracy in function of value $C$, related to the slack variable. kernel='linear', decision_function_shape="one-vs.-one" ........................................... 60

7.4 Learning Curve showing the Linear Support Vector Machines model performance improvement as the size training samples grows. kernel='linear', $C = 10$, decision_function_shape="one-vs.-one" ........................................... 62

7.5 Validation Curve showing the Support Vectors Machine model accuracy in function of value $C$, related to the kernel dissimilarity. Kernel = 'rbf', $gamma = 1/21$ ('default' value), decision_function_shape="one vs one". ........................................... 63

7.6 Validation Curve showing the Support Vectors Machine model accuracy in function of value $\gamma$, related to the slack variable. Kernel = 'rbf', $C = 4$, decision_function_shape="one vs one". ........................................... 64

7.7 Learning Curve showing the Support Vector Machines model performance improvement as the size training samples grows. Kernel = 'rbf', $C = 4$, $\gamma = 0.04$, decision_function_shape="one vs one". ........................................... 64

7.8 Learning Curve showing the K-NN model performance improvement as the size training samples grows. Nº of neighbours $K = 8$, NN algorithm = "kd tree", weights= "uniform". ........................................... 65

7.9 Simplified scheme of the Random Forest Algorithm. ........................................... 65

7.10 The training samples circled are the support vectors and the distance between two classes of support vectors is the margin. ........................................... 65

7.11 Validation Curve showing the Random forest model variance getting lower as the minimum number of samples per child node grows. Nº of estimators = 200 , Max $m = 2$. ........................................... 66

7.12 Learning Curve showing the Random forest model performance improvement as the size training samples grows. Nº of estimators=220 , Max. $m = 8$, Min. Nº Samples in Child Nodes = 15. ........................................... 66

7.13 Confusion Matrix. ........................................... 67

7.14 Strip from classified route taken by a unseen driver. Developed with Google Maps API . 68

8.1 Radar Graphs showing the maneuver index values for each maneuver to different drivers. The names are fake to maintain the anonymity of the drivers in the study. The graphs were created using Chart.js. ........................................... 72

8.2 Driver Profile for the Yellow Driver. Under the name a quick description of the driver is presented. The table shows for each maneuver the number of occurrences in each categorie and the maneuver index. ........................................... 73
8.3 Doughnut Graph on the left illustrates and compares the number of occurrences of each maneuver. The radar graph displays the maneuver index for each maneuver. 73
8.4 Map division of the BeDriver webpage, demonstrating a quick strip of a colour coded classified route. Black = Normal, Red = Brake, Orange = Roundabout. 74
A.1 Schematic of the microcontroller and some peripherals. 88
A.2 Schematic of the ICSP connector and crystal connected to the XTAL pins of the microcontroller. 88
A.3 Schematic of the MicroB USB power supply and voltage regulator. 89
A.4 Schematic of the LEDs implemented in the system. 89
A.5 Schematic of the accelerometer. 90
A.6 Schematic of the Global Positioning System (GPS) module and its peripherals. 91
A.7 Schematic of the microSD. 92
A.8 Schematic of the SPI level shifting. 92
A.9 Schematic of the UART and I2C level shifting. 92
A.10 Image of the whole final schematic of the developed embedded system. 93
B.1 Impedance Controlled Routing Parameters. 96
B.2 Micro B symbol and package specially made for the connector with manufacturer part number 10118192 – 0001LF. 97
List of Tables

1.1 Collection of State of the Art papers searched. ........................................... 5

3.1 Comparison between the output and input signal voltages of two components powered at different TTL voltages. .......................................................... 27

4.1 Highlight of code related differences between MMA8452Q and MMA8451Q accelerometers. 38

5.1 Noise energy comparison in both low-pass techniques with the same data sample. ........ 48

6.1 Maneuvers and correspondent numeric labels. .................................................. 55

6.2 Total number of maneuvers of each type in the dataset with respective mean time. .......... 55

6.3 Features Selected to characterize each time frame. Acc. or Acceleration includes the X and Y axis. ................................................................. 56

7.1 Comparison of the 4 models, through Accuracy, Precision, Recall and F1-Score obtained with the test set. ................................................................. 68

8.1 Acceleration Categories with the corresponding ranges for each label. ....................... 71

B.1 Impedance Controlled Routing Parameters Influence on the impedance value. ........... 96
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSR</td>
<td>Autoridade Nacional Segurança Rodoviária</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>BEAV</td>
<td>Boletim Estatístico de Acidentes de Viação</td>
</tr>
<tr>
<td>C/A</td>
<td>Coarse/Acquisition Code</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer Aided Design</td>
</tr>
<tr>
<td>CAM</td>
<td>Computer Aided Manufacturing</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary Semi-Oxide-Semiconductor</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>DTW</td>
<td>Dynamic Time Warping</td>
</tr>
<tr>
<td>DUI</td>
<td>Driving Under the Influence</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Electrically Erasable Programmable Read-Only Memory</td>
</tr>
<tr>
<td>EGNOS</td>
<td>European Geostationary Navigation Overlay Service</td>
</tr>
<tr>
<td>ERSO</td>
<td>European Road Safety Organization</td>
</tr>
<tr>
<td>GLONASS</td>
<td>Global’naya Navigatsionnaya Sputnikovaya Sistema</td>
</tr>
<tr>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GPIO</td>
<td>General Purpose Input Output</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile communication</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Models</td>
</tr>
<tr>
<td>(I^2C)</td>
<td>Inter-Integrated Circuit</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated Circuit</td>
</tr>
<tr>
<td>K-NN</td>
<td>K-Nearest Neighbors</td>
</tr>
<tr>
<td>LED</td>
<td>Light Emitting Diode</td>
</tr>
<tr>
<td>MDC</td>
<td>Mobility Device Controller</td>
</tr>
<tr>
<td>MEO</td>
<td>Medium Earth Orbit</td>
</tr>
<tr>
<td>MISO</td>
<td>Master In/Slave Out</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>MOSI</td>
<td>Master Out/Slave In</td>
</tr>
<tr>
<td>NAVSTAR-GPS</td>
<td>Navigation System with Time and Ranging Positioning System</td>
</tr>
<tr>
<td>NMEA</td>
<td>National Marine Electronics Association</td>
</tr>
<tr>
<td>PAYD</td>
<td>Pay-As-You-Drive</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
</tr>
<tr>
<td>PDIP</td>
<td>Plastic Duo-In-Line Package</td>
</tr>
<tr>
<td>PPS</td>
<td>Precise Positioning Service</td>
</tr>
<tr>
<td>QFN</td>
<td>Quad Flat No Leads</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>RoHS</td>
<td>Restriction of Certain Hazardous Substances</td>
</tr>
<tr>
<td>RTC</td>
<td>Real Time Clock</td>
</tr>
<tr>
<td>SBAS</td>
<td>Satellite-Based Augmentation Systems</td>
</tr>
<tr>
<td>SCK</td>
<td>Synchronization Clock</td>
</tr>
<tr>
<td>SEM</td>
<td>Scanning Electron Microscope</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>SMD</td>
<td>Surface Mount Device</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise ratio</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>SPS</td>
<td>Standard Positioning Service</td>
</tr>
<tr>
<td>SRAM</td>
<td>Static Random Access Memory</td>
</tr>
<tr>
<td>SS</td>
<td>Slave-Select/Chip Select</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>THT</td>
<td>Through-Hole Technology</td>
</tr>
<tr>
<td>TTL</td>
<td>Transistor-Transistor Logic</td>
</tr>
<tr>
<td>TWI</td>
<td>Two Wire Interface</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver/Transmitter</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned Aerial Vehicle</td>
</tr>
<tr>
<td>UBI</td>
<td>Usage Based Insurance</td>
</tr>
<tr>
<td>WHO</td>
<td>World Health Organization</td>
</tr>
</tbody>
</table>
Introduction

Contents

1.1 Motivation ................................................................. 3
1.2 CEIIA-Centre of Engineering and Product Development ................. 4
1.3 Related Work ............................................................ 5
1.4 Objectives and Innovation ............................................. 9
1.5 Context of this work in Aeronautic Industry ........................... 10
1.6 Thesis Structure ......................................................... 10
1.1 Motivation

According to the World Health Organization (WHO), road safety is a major and global problem. In their Global Status Report of 2015 [1] is stated that, just in 2013, there were 1.25 millions of traffic-related fatalities, being the main cause of death for people aged between 15-29 years old.

In Europe, according to the Annual Accident Report by the European Road Safety Organization (ERSO), from 2005 to 2014 57.000 young people lost their lives on the road, making up to 1/5 of the total fatalities. Deaths per million population due to road-related accidents are shown in Figure 1.1.

Even when fatalities are not observed, substantial accident-related costs exist. Just in the US in 2011, 299.5 billion dollars were estimated as costs [3]. Independently of the dimension of a country, such costs can highly affect the economic growth.

To fight this phenomenon the "Decade of Action in Road Safety(2011-2020)" has been launched by the WHO and has already taken place in order to mitigate damage [1]. Despite the population growth and increase in the number of vehicles, the fatalities have stabilized worldwide and lowered in the European Union [4]. The reasons for traffic accidents can either be external to the driver or dependent of the human behaviour. External to the driver, it is included the pavement roughness and vehicle conditions. Pavement quality affects the ability to drive, the safety of the driver and fuel consumption of the vehicle, as stated by Du et al. [5]. Vehicle condition is also a big factor, since according to WHO, in 80% of countries worldwide, the sold vehicles do not meet basic safety standards [1].

Figure 1.1: Road fatality rates per million population by country, 2014 [2]. In Portugal there were over 600 deaths per year on the road.
Human driving behaviour is also one important factor. According to WHO a 30% reduction in fatal accidents could be achieved with a simple cut of 5% in speed [6]. An enforcement of drink-driving laws could also result in a 20% reduction of accidents [7]. Driver's behaviour and judgment influence his overall safety can deteriorate the vehicle condition and raise the fuel consumption. It is important to study the different driving patterns for a wide range of applications like car sharing and eco-driving.

Car sharing, as explained by Bardhi et al. [8], can be described as a group of individuals who have access to a car fleet for which they periodically pay for. These programs generate billions in revenues annually and can be directed to companies or consumers that live in cities. Existing car sharing companies, like Zipcar [9], demand a rigorous new member check of driving history and monitoring data for each driver. Since multiple drivers will use the same vehicle, becomes important to be able to distinguish between each one and control users with hazardous behaviour.

Eco-driving consists in maintaining a smoother driving. This profile, according to [10], [11], can contribute up to 10% reduction in fuel consumption and thereby lower CO₂ emissions.

1.2 CEIIA-Centre of Engineering and Product Development

CEIIA is the Centre of Engineering and Product Development, headquartered in Matosinhos, that innovates in the design, implementation and operation of their products in areas such as Aeronautic, Mobility, Naval and Automotive [12]. It is a non-profit organization and one of the main innovators in Portugal, employing more than 200 people in 7 different countries. The centre is very concerned with sustainable mobility and development of smarter cities. For these reasons, CEIIA participated in the COP21, presenting mobi.me to UN Secretary-General Ban Ki-moon and US Secretary of State John Kerry [13]. MOBI.ME is a smart management system for urban mobility able to measure CO₂ emissions and it manages the electric vehicle fleet of the UN Brazil in Brasilia since mid-2015.

The mobi.me system also supports the scooter sharing service eCooltra, launched by Cooltra, one of the biggest scooter rental players [14]. It provides real-time communication and integration of 250 electric scooters, as well as the remote control of users. The prototype developed in this thesis may then help to lead the way to a better understanding of current product users, identifying their driving profiles not only on cars but also in the scooter sharing with integration on the Mobility Device Controller (MDC). Data would always be analyzed only with the consent of each and every driver. In a fleet management or sharing service perspective, it is also very advantageous to have profile information. It can be related to fuel consumption, CO₂ emissions, vehicle wear and accident probability, features that can change service costs, depending on the driver behaviour. Therefore, the driving profile could be an important variable in the business model.
1.3 Related Work

Table 1.1 summarizes the state of the art of driver behaviour studies. Efficiency is stated as provided in each paper and reflects the success of the algorithm in its test datasets. It is, therefore, not comparable among papers, since they present very diverse applications. In the following sections, are given more details regarding some of the more relevant papers.

Table 1.1: Collection of State of the Art papers searched.

<table>
<thead>
<tr>
<th>Author</th>
<th>Detection and Prediction</th>
<th>Platform</th>
<th>Sensors</th>
<th>Sampling</th>
<th>Method</th>
<th>Features</th>
<th>Accuracy</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kuge et al. (2000) [15]</td>
<td>Lane Change Maneuver</td>
<td>Simulator</td>
<td>Multiple sensors</td>
<td>5s window</td>
<td>HMM</td>
<td>Lat. Position, Steering angle</td>
<td>98.3%</td>
<td>Determines normal from emergency lane change</td>
</tr>
<tr>
<td>Ichikawa et al. (2005)</td>
<td>Maneuver Detection</td>
<td>Simulator</td>
<td>Multiple sensors</td>
<td>—</td>
<td>HMM</td>
<td>Lat., Orientation, derivatives, entropy</td>
<td>82%</td>
<td>Detection of drivers in analogy to phonemes</td>
</tr>
<tr>
<td>Mitrovic et al. (2005)</td>
<td>Maneuver Detection</td>
<td>Sensor Board+PC</td>
<td>Acc., Gyro and GPS</td>
<td>50ms and 1s</td>
<td>HMM</td>
<td>Speed, Lateral and Longitudinal Acceleration</td>
<td>98.3%</td>
<td>Vector quantization and Event Segmentation</td>
</tr>
<tr>
<td>Lue et al. (2010) [19]</td>
<td>Drunk Driving</td>
<td>Smartphone</td>
<td>Accelerometer</td>
<td>5s window</td>
<td>DTW</td>
<td>Vel., Acc., Image and Audio</td>
<td>97%</td>
<td>Considers HMM training time consuming</td>
</tr>
<tr>
<td>Johnson et al. (2011)</td>
<td>Aggressive Maneuvers</td>
<td>Smartphone (MRoad)</td>
<td>Multiple sensors</td>
<td>25Hz</td>
<td>DTW</td>
<td>Vel., Acc., Image and Audio</td>
<td>97%</td>
<td>Considers HMM training time consuming</td>
</tr>
<tr>
<td>Castignani et al. (2013)</td>
<td>Driving Profile Scoring</td>
<td>Smartphone</td>
<td>Acc. and GPS</td>
<td>5Hz and 1Hz</td>
<td>Fuzzy Logic</td>
<td>Acc., Steering and Overspeed</td>
<td>82%</td>
<td>Normal/Moderate (Aggressive, Urban/ extra-urban)</td>
</tr>
<tr>
<td>Bergasa et al. (2014)</td>
<td>Drowsiness and Distraction</td>
<td>Smartphone</td>
<td>Multiple sensors</td>
<td>250ps, 100Hz</td>
<td>Image Recognition, Thresholds</td>
<td>82%</td>
<td>Compared with AXA app</td>
<td></td>
</tr>
<tr>
<td>Du et al. (2016) [5]</td>
<td>Pavement Roughness</td>
<td>2 Acc. and GPS</td>
<td>20Hz</td>
<td>IRI and RQI</td>
<td>Vertical Acc.</td>
<td>82%</td>
<td>One Acc. for each rear wheel., GPS used for detection of road.</td>
<td></td>
</tr>
<tr>
<td>Dong et al. (2017)</td>
<td>Driving Styles</td>
<td>GPS</td>
<td>—</td>
<td>Deep Learning</td>
<td>—</td>
<td>—</td>
<td>82%</td>
<td>Deep Neural Networks exhibit better scalability</td>
</tr>
<tr>
<td>Yu et al. (2017) [24]</td>
<td>Abnormal Driving</td>
<td>Smartphone</td>
<td>Magnetometer and GPS</td>
<td>—</td>
<td>SVM and NN</td>
<td>Mean, Max., Min Std. event time</td>
<td>96.36%</td>
<td>6 months of driving data in real conditions</td>
</tr>
</tbody>
</table>

1.3.1 DriveSafe: an App for Alerting Inattentive Drivers and Scoring Driving Behaviors

DriveSafe is an app created by Bergasa et al. [22] that detects inattentive driving, such as drowsiness and distraction, giving feedback under real-world conditions. It makes use of computer vision and pattern recognition techniques with the data from smartphone sensors.

For Drowsiness and Distraction, detection has used the camera, microphone, GPS and accelerometer from a fixed smartphone. The camera was directed to the road. With these sensors, they were able to detect in a robust way lane drifting, weaving and violation of predefined acceleration and turning events thresholds. Both of the two inattentive driving characteristics were scored in events per kilometre.

Unlike a "black-box" method, the driver had access to the scoring process and computer vision techniques, as seen in the GUI of Figure 1.2, which can be distracting, even with the incorporated option to turn off the visual distraction. The overall accuracy of the app was up to 82%, and it presented results similar to the existing AXA Drive app [25].
1.3.2 Mobile Phone Based Drunk Driving Detection

Drinking Under the Influence (DUI) or more commonly, drunk driving, is one of the main causes of crashes and consequent fatalities in the road [7]. Since it is impossible to patrol and check every driver that might be drunk, Dai et al. proposed in [19] an app that can detect drunk drivers and alert them in real time. For implementation of the algorithm, it is made only use of the accelerometer of a smartphone in order to detect certain drunk driving cues like weaving or wide radius turns. The implementation showed promising results, however, the possible sliding of the free smartphone inside the car deteriorated the results.

1.3.3 Driving Scoring for Fleet Management and Insurance Applications using Fuzzy Logic

Driving profile of each driver is an important aspect that can be used in multiple commercial applications. In fleet management, information on the vehicle degradation and fuel consumption by different drivers can be used to promote a more sustainable company. In the car insurance market, the profile information could be used in a Usage Based Insurance (UBI) or Pay-As-You-Drive (PAYD) schemes.

A way of obtaining data makes use of Telematics boxes (e.g., Ingenie [26], Fairplay [27]). These boxes contain a GPS and other sensors, such as an accelerometer, as well as a SIM card to transmit data. However they operate in a "black-box" methodology, that is considered intrusive by some of the drivers. Another approach is the use of smartphone apps like Aviva RateMyDrive [28] and StateFarm DriverFeedback [29] that can monitor and give a score to the driver, helping adjust the insurance policy.

Inside this area, Castignani et al. suggest the use of Fuzzy-Logic, which combines input data with chosen inference rules in order to give a reliable score that suggests each driver profile [21], [30], [31]. In his later work, an app was created with the name SenseFleet. The scheme used is presented in figure 1.3. A sensor fusion is made with accelerometer and GPS in order to acquire data. After filtering the data, events like Overspeed time and a number of moderate or aggressive acceleration and steering rate are collected. The fuzzy sets help score in four categories, Calm, Average, Moderate and Aggressive.
1.3.4 Driving Style Recognition with Dynamic Time Wrapping

In his paper Johnson et al. propose the use of Dynamic Time Warping (DTW) in the detection of aggressive behaviour, making use of the MIROAD system [20]. This system makes use of the camera, accelerometer, gyroscope and GPS, all present in most of the smartphones. It works in an active mode, that analyses in real-time 5 minute segments, discarding them in case no aggressive maneuvers are detected. The passive mode, stores all data in five minute segments to be analysed in the future. Events are detected if inputs are over a chosen threshold, considered aggressive behaviour. DTW algorithm is used to find the closest match between the 120 templates that describe aggressive and non-aggressive maneuvers. The method had problems differentiating certain types of curves (like left turn and U-turn), however, it was able to detect up to 97% of aggressive behaviour.

1.3.5 Hidden Markov Method for detection of Driving Behavior

As described by Mitrović [17], through the learning from previous driver’s experiences it is possible to differentiate different driving experiences as well as find similar models for routes in different geographic locations. These similarities cannot be identified without making use of learning techniques, this being the main advantage of the use of models like Hidden Markov Models (HMM).

Kuge et al. tested HMM in the recognition of lane changing maneuvers considering 3 different possible states: Lane Change Emergency, Lane Change Normal and Lane Keeping [15]. His results were promising in the use of maneuver recognition to model the human behaviour as well as to detect a maneuver pattern before the maneuver is completed which suggests prediction capabilities.

Torkkola et al. inspired in speech recognition created the drivemes in analogy to phonemes [16]. Driveme is a state or number of states that make a HMM. Each HMM model relates to a different
maneuver. They detected that some maneuvers had *drivemes* in common, which reduced the number of states to less than one half.

Mitrović also used HMM for recognition of different driving events: Left Curve, Right Curve, Left and Right Intersection with or without roundabout [17]. He pondered the number of states considering that a small number would take out the flexibility of identification but a too big of a size would generate many false positives. After a compromise, he was able to identify 98.3% of driving events.

Takano *et al.* [18] approach to the problem included 4 distinct phases: Segmentation, Symbolization, Recognition and Generation of driving patterns. Previous works from different authors would follow similar work-flow, however, in that paper, the framework was explicitly presented in a pyramid, Figure 1.4. *Proto-symbol* was the name given to the HMM in contrast with the state definition by Torkkola *et al.*, and based on the hierarchy between them they were able to predict with considerable precision the steering angle, however still faulty. Based on the tested sensors they determined the vehicle speed and acceleration, two of the most critical features for recognition.

![Figure 1.4: Overview of a framework based on symbolization, recognition and generation of driving pattern [18]._generation works as a prediction of the following proto-symbol, generating the pattern.](image)

### 1.3.6 SVM in driver identification

Support Vector Machine (SVM) is a machine learning that, more recently, was used to study the influence of traffic in the prediction of maneuvers inside roundabouts [32] as well as real time identification of driving behaviour [24]. Different maneuvers considered dangerous were identified making use of accelerometer and gyroscope data. 16 features like standard deviation, mean, maximum and minimum were extracted from the low-pass filtered raw data of different drivers. The classifier obtained showed promising results and it had an accuracy very similar to the neural networks studied in the same paper.
1.3.7 Deep Learning in driver identification

Actual machine learning algorithms are dependent on the features selected by the data scientist, which are hard to select and most of the times do not scale very well. For this and other reasons, Deep Learning has become up-and-coming in the last years. Dong et al. [23] used information from a large number of different trips from unique drivers to train 5 types of neural networks including non-deep learning networks for comparison. He then compared the performance of each one, as well as the stability of the algorithms when the experiment was made between 50 to 1000 drivers. The results were confirmed that the deep learning methods have a better scaling, however, the best results are related to the computationally more expensive neural network. In the same paper is made a final reflection on the anonymity of the data, being defended that the presented method, that does not use any location or time data is less invasive.

1.4 Objectives and Innovation

The main objective of this thesis is to develop a complete embedded system, that collects data, as well as the analysis of such data in order to visualize driver's behaviours characteristics. It includes the design and development of an embedded system with a microcontroller, accelerometer and GPS for data collection as well as the data analysis that makes possible the retrieval of relevant information from the driver.

The first part of the work is focused on the consolidation of electronics theoretic knowledge acquired during the master degree and its application in a corporate environment. With no previous knowledge about the design and production of printed circuit boards, an embedded system is going to be developed. All components will be selected according to the needs of the sensors and production constraints will be taken into account for the production of the board. The microcontroller in the system will also need to be programmed in order for the data to be correctly collected.

The second part is an introduction to an area of interest that did not make part of the courses taken in the degree. New concepts and algorithms related to machine learning will be introduced and their practical application and validation understood. Also, for a more interactive presentation of the data, a complete web platform will be created. This last step involves languages and frameworks also not experienced during the degree.

Even considering the seen works do not correspond to the totality of developed work in this area, this thesis proposes to innovate by trying to prove that a whole workflow, from data collection to results visualization, can be created to extract information from drivers in computationally efficient and most flexible way.
1.5 **Context of this work in Aeronautic Industry**

The sensors and models used in this thesis have a number of potential applications in the scope of the aeronautical industry.

The hands-on process of an embedded system development, with all development phases, from initial concept, to design and manufacturing with cost estimates are an important skill for any avionics engineer. Even more with accelerometers and GPS sensors that are widely used in aircraft’s and all kinds of Unmanned Aerial Vehicle (UAV) to help with the control and guidance.

The Machine Learning Algorithms and Data Processing are starting to be widely used in the aviation and space industry. They can be used for the improvement of fleet management in airlines, engine optimization, development of spacecraft design assistants or even to make UAVs fly themselves [33]. It is an up-and-coming topic in the industry and will certainly continue to gain more relevance in the next years.

1.6 **Thesis Structure**

This thesis was separated into 9 Sequential Chapters. This means that later chapters will make use of information or hardware explained in previous ones:

- Chapter 2: The Embedded System main components and bus protocols used are explored and explained;
- Chapter 3: The different procedures on the development of a Printed Circuit Board are explained taking into account the needs of the embedded system;
- Chapter 4: The embedded system is programmed and tested in different positions inside a car. The software is firstly evaluated in an Arduino with the same sensors;
- Chapter 5: Data is corrected, completed and smoothed. The performance of three low-pass filters is compared on real acceleration data;
- Chapter 6: Data is labelled on different maneuvers/events and transformed to include relevant features;
- Chapter 7: Four models are built using the K-NN, SVM and Random Forest algorithms. The parameters of each model are tuned taking attention to the variance and bias, and their performance compared in terms of accuracy and F1-score;
- Chapter 8: Categories for each maneuver, taking acceleration as a metric are presented. The Maneuver Index concept is described and used to help characterize different drivers;
- Chapter 9: Conclusion and Future Work.
Embedded System Concepts

Contents

2.1 Microcontroller .................................................... 13
2.2 Accelerometer ....................................................... 15
2.3 GPS Module .......................................................... 17
2.4 Low-Level Communication Protocols ............................. 19
Embedded System is an electronic system integrated into an electrical or mechanical device, dedicated to performing a specific amount of tasks. Taking into account the information of the Chapter 1, the main components of the system are going to be an Accelerometer, to measure acceleration, a GPS module, to gather position and speed of the vehicle, an \(\mu\)SD socket and card, to save the data and a microcontroller, to collect data from the sensors and save it on the card, following a specific algorithm. The Figure 2.1 illustrates the system and the protocols needed for the implementation of a simplified scheme.

![Figure 2.1: Simplified scheme of the desired embedded system components and which protocols are needed for the implementation.](image)

### 2.1 Microcontroller

Microcontrollers are very small electronic devices, with memory and a processor, integrated into a unique chip that can be connected to sensors/actuators and programmed to serve an application. Due to its characteristics and price, they are widely common in embedded systems from vehicles to house appliances.

A popular microcontroller, used in the Arduino Platform [34], is the Atmega328P (fig.2.2) by Atmel. It has the following main characteristics [35]:

- **8-bit AVR CPU**, one of the most commonly used for 8-bit systems, able to run at 16MHz;
- **I\(^2\)C, SPI and UART** available to communicate with accelerometer, \(\mu\)SD card and GPS module.
- **Plastic Duo-In-Line Package** (PDIP), easier to solder and replace manually, compared to smaller packages such as the Quad Flat No Leads (QFN) package;
The most important blocks of the microcontroller are shown in the figure 2.3.

- **Central Processing Unit (CPU)**, acts like the "brain" of the microcontroller, connecting the blocks between each other through the databus. It receives, decodes and ensures that the instructions are correctly executed;

- **Flash Program Memory**, for the ATmega328P, a 32Kbytes reprogrammable non-volatile memory that stores 16bits wide instructions;

- **Static Random Access Memory (SRAM)**, offers space for registers, some dedicated and others accessible, as well as space to store variables. Unlike the Flash memory and EEPROM this
• **Electrically Erasable Programmable Read-Only Memory (EEPROM)**, is used to hold data that is not intended to be deleted when the system gets without power. For example, this can include configurations or variables.

• **General Purpose Input Output (GPIO) Ports**, there are 3 groups, Port B and Port D with 8 pins, mostly digital pins and Port C with 6 pins, all analogue pins.

• **System Clock**, the microcontroller has an internal RC oscillator, however, an external 16MHz Crystal is used instead. The crystal increases the data transfer speed and improves the phase noise stability as well as the jitter, deviation on the delay of the data flow.

• **Watchdog Timer**, this timer can be enabled through the fuse register, near the RC oscillator. Once enabled it starts a countdown that can be restarted by a programmed instruction. When the countdown ends the microcontroller is rebooted.

• **Brown out Detection**, this feature is also activated through the fuse. In case the DC power of the microcontroller gets lower than the programmed minimum tension, the device is rebooted.

Due to the architecture and technical details presented about the microcontroller, it is considered to be the most appropriate device for this thesis’s application. It is affordable, easy to implement in a PCB, and provides the flexibility needed for this project. The **Watchdog Timer** and the **Brown out Detection** are not critical for the microcontroller to complete its task. However, they ensure a robust and more stable system, since, by rebooting when the CPU stops running instructions (because of a bug for example) or non-optimal conditions occur, it makes sure no registers in the EEPROM are corrupted, as well as no invalid data is collected.

### 2.2 Accelerometer

The Accelerometer is a very common Integrated Circuit (IC) for measuring acceleration. They have many applications, from inertial navigation in airplanes to sudden velocity changes detection to inflate airbags in a car [36]. It is even used to detect a laptop in free fall, diverting the needle from scratching the hard drive [37]. The most used type of accelerometer in embedded systems is the semiconductor accelerometer. It has high accuracy and stability, low power consumption and it is nanofabricated, which means its dimensions are smaller than a mechanical accelerometer. Taking a closer look at the figure 2.4 it is possible to explain the operation of this type of sensor.

A movable mass changes two capacitance values by moving closer or away from the fixed electrodes. Comparing the capacitances, the motion orientation can be found. In an accelerometer like the one in
Figure 2.4: Scheme of the inner workings of a semiconductor accelerometer [38].

Figure 2.5: SEM photo of an accelerometer by Analog Devices Inc. [39].

The accelerometer selected for the developed embedded system is the MMA8451Q [40]. It has, among others, the following characteristics:

- **QFN package**, even being a SMD difficult to solder by hand, it is still possible. Also this accelerometer comes with a great amount of other features, such as adjustable bandwidth, and it is cheaper than other easier to solder packages;

- **3-axis**, that gives freedom in the range of applications with the data. It is possible to analyze the driver behaviour as well as the roughness of the pavement;

- **Output data rates** from 1.56Hz to 800Hz;

- **Dynamic Ranges** of $\pm 2g$, $\pm 4g$ or $\pm 8g$ ($1g = 9.8m/s^2$);
• Sensitivity from 4096 counts/g to 1024 counts/g depending on the scale;

• \( I^2C \) communication protocol.

Figure 2.6: MMA8452Q digital accelerometer (QFN) [41].

Figure 2.7: Gtop.013, GPS module.

2.3 GPS Module

The Global Navigation Satellite System (GNSS) consists of a constellation of satellites in Medium Earth Orbit (MEO) that globally transmit signals and provide information about a receiver’s position. The most widely used, is the GPS [42], however other options like the European Galileo [43] or the Russian Global’naya Navigatsionnaya Sputnikovaya Sistema (GLONASS) [44] could be used.

The Navigation System with Time and Ranging Positioning System (NAVSTAR-GPS), or just GPS, is owned by the US military and has more than 30 satellites in MEO. The fundamental frequency of the GPS is \( 10.23 \times 10^6 \) Hz, but the signals are transmitted in two carriers, L1 (\( 154 \times 10.23 \times 10^6 \)) and L2 (\( 120 \times 10.23 \times 10^6 \)), being the second carrier used for self-calibration [45]. The signals are:

• Coarse/Acquisition Code (C/A), is a code sequence on the L1 carrier that repeats every 1ms. Consists in 1023 bits that identify the satellite and contain information about the satellite clock;

• Precision Code, it is an encrypted signal, repeated every 7 days. It also contains information about the satellite clock, but with a higher precision;

• Navigation Message, is a sequence with 1500 bits that takes 30 seconds to transmit. It contains information about satellite orbit and other system parameters.

The encrypted signal is a deliberate decision made by the US military to compromise the GPS service in other countries. In other words, the position precision is reduced due to encryption. The GPS service can then be distinguished in two levels of operation:
• **Precise Positioning Service (PPS)**, used only by the military or authorized entities;

• **Standard Positioning Service (SPS)**, available to anyone with a GPS receiver, however it is less accurate (up to 50 meters error).

The deterioration of the signal precision, by encryption as well as other external factors, makes the precision of the system not good enough for some applications, such as aeronautical guidance. For these reasons the Satellite-Based Augmentation Systems (SBAS) exists, being the one used in Europe called European Geostationary Navigation Overlay Service (EGNOS). It complements the GPS constellation signals with the signal of multiple ground stations. It increases the accuracy and integrity of the system as a result.

To conclude, this is the explanation of how the position of the GPS module is computed. The distance, \( R_i \), from a satellite to a receptor is given by:

\[
R_i = \sqrt{(x_{si} - x_u)^2 + (y_{si} - y_u)^2 + (z_{si} - z_u)^2}
\]  

(2.1)

where:

- \( s_i \) = Satellite \( i \)
- \( u \) = Receptor

In order to obtain the values of the 3 coordinates is necessary to apply a triangulation with 3 different satellites. Only this way, there would be 3 equations for 3 unknown variables. However, the obtained value would still contain an error greater than expected. This is due to the error of the clock in the GPS module, in comparison with the atomic clock in the satellites, \( \Delta t_u \). It is then necessary a fourth satellite signal to solve the Equations 2.2 of the satellites pseudo-range [45]. The pseudo-range is used rather than just range because of the unknown receiver clock offset.

\[
\begin{align*}
PR_1 &= \sqrt{(x_{s1} - x_u)^2 + (y_{s1} - y_u)^2 + (z_{s1} - z_u)^2} + c\Delta t_u \\
PR_2 &= \sqrt{(x_{s2} - x_u)^2 + (y_{s2} - y_u)^2 + (z_{s2} - z_u)^2} + c\Delta t_u \\
PR_3 &= \sqrt{(x_{s3} - x_u)^2 + (y_{s3} - y_u)^2 + (z_{s3} - z_u)^2} + c\Delta t_u \\
PR_4 &= \sqrt{(x_{s4} - x_u)^2 + (y_{s4} - y_u)^2 + (z_{s4} - z_u)^2} + c\Delta t_u
\end{align*}
\]  

(2.2)

where:

- \( PR_i \) = Pseudo-Distance of the receptor to a satellite \( i \)
- \( s_i \) = Satellite \( i \)
- \( u \) = Receptor
- \( c \) = Speed of light \((3 \times 10^8 m/s)^2\)
- \( \Delta t_u \) = Error on the clock of the module’s receptor in comparison the the satellites

The module selected for the embedded system was the GTPA013, shown Figure 2.7 [46]. This GPS module has a built-in antenna, it is Surface Mount Device (SMD) and has an update rate up to 10 Hz.
2.3.1 NMEA 0183 Standard

National Marine Electronics Association (NMEA) 0183 is the Interface Standard, created by the National Marine Electronics Association, used in most GPS modules as well as other marine electronics [47]. It defines the electrical signal requirements and the sentence formats for serial (UART) bus. The standard was designed to have one device sending to multiple listeners and the data in American Standard Code for Information Interchange (ASCII) [48] form, including position, speed, time, course and other information.

When communicating with the GPS module, different messages are received, such as \texttt{GPRMC} that corresponds to the minimum recommended data. These messages are expected by the microcontroller and give information about, for example, a number of satellites in view and, if a fix is obtained, the Coordinates and Speed of the Module.

2.4 Low-Level Communication Protocols

The Parallel interface transmits multiple bits at the same time, having a dedicated connection for each one. It was commonly used to connect peripherals like printers. However, if the communication between all the peripherals is made with 8 bits, the number of connection pins on a small microcontroller would make the implementation too complex and would not be cost-effective. For this reason, serial interfaces, where only a bit is sent at a time, are used.

The serial communications can be asynchronous, as illustrated in Figure 2.8(a), like UART or synchronous, like SPI and \textit{I}^2\textit{C}. The synchronous communications have the data paired with an extra clock line, as in Figure 2.8(b).

![Illustration of the two types of serial communication.][49]

The Baud Rate is an important parameter, it defines the speed of the serial line in bits-per-second,
bps. A usual standard value for UART protocol is 9600bps. Another important rule in serial communications is the data framing, as shown in Figure 2.9. There is a start bit and one or two stop bits, that show where the message begins and ends. The data message can be transmitted in frames of 5 to 9 bits, depending on the message. Some protocols include also a parity bit.

![Figure 2.9: Data Framing with voltages, start and stop bit of usual TTL serial.](image)

Also in Figure 2.9 is possible to analyze a Transistor-Transistor Logic (TTL) serial signal. In TTL the supply voltage is 5V to GND. In the projected embedded system the Complementary Semi-Oxide-Semiconductor (CMOS) voltage, 3.3V is also used. The conversion among these voltage levels is explained in the Section 3.2.

### 2.4.1 UART Protocol

To establish communication between the GPS module and the microcontroller it is used UART protocol. In this protocol, the information is transmitted through the serial lines are called RX and TX for receiving and transmitting, but taking into account the device. This means the RX line in one device must be connected to the TX line of the other device. The transmission can be either:

- **Simplex**, using only one line, where only one of the devices sends and the other receives the data;

- **Full-duplex**, when the devices can send and receive at the same time;

- **Half-duplex**, when the devices can send and receive, but only one action at the time.

The main disadvantage of this protocol is that both devices need to be synchronized. In case one of the clocks is not as precise, some messages might be lost. In the synchronous protocol, this problem is overcome since there’s a mutual clock line that keeps the synchronization between devices, as presented in Figure 2.8(b).
2.4.2 SPI Protocol

The SPI is a synchronous serial communication protocol. It is used in this embedded system to communicate with the µSD card and also to program the microcontroller. SPI protocol works as a Master-Slave, Full-Duplex Interface, where one device, the microcontroller, is the master that controls the clock and all other devices are the slaves, as illustrated in Figure 2.10.

![Figure 2.10: SPI configuration with 1 master and 3 slaves.]

There are 3 lines that are connected to all the devices:

- **Synchronization Clock (SCK)**, line of the synchronization clock, that is controlled by the master;
- **Master Out/Slave In (MOSI)**, Master Out/Slave In, transfers data from the master to the selected slave;
- **Master In/Slave Out (MISO)**, Master In/Slave Out, transfers data from the selected slave to the master.

The other lines are the Slave-Select/Chip Select (SS) that select the slave with which the master is communicating. The logic of the SS is called *active low*, which means that when a slave is communicating its value is pulled to *Low*, ’0’, and in all the other slaves is kept *High*, ’1’. SPI supports high clock rates (up to 10MHz), however, if such bit transmission speed is not required, a solution with less wire/pin connections can be used, called *I²C*.

2.4.3 I²C Protocol

*I²C* or Two Wire Interface (TWI) (as it is referred in the Atmega datasheet) is a synchronous serial interface developed by Philips that allows multi-master communication at frequencies from 100kHz to 400kHz. In the developed embedded system it will be used for communication between the microcontroller and accelerometer. To achieve that, it needs only 2 signal wires, *SCL* and *SDA*. *SCL* is the
clock signal used for synchronization and \textit{SDA} is the data signal \textsuperscript{50}. This serial bus has the purpose of establishing short distance, low velocity communications and follows the scheme of the Figure 2.11. The two pull up resistors are used to set both signal lines to high when the devices are not forcing any value, since the \textit{I}²\textit{C} protocol pins are open-drain.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{i2c_config.jpg}
\caption{\textit{I}²\textit{C} configuration with 2 masters and 2 slaves.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{i2c_signal_protocol.jpg}
\caption{\textit{I}²\textit{C} signal protocol for 8bits devices \textsuperscript{51}.}
\end{figure}

The Figure 2.12 helps demonstrate the principle of \textit{I}²\textit{C} protocol. When the master desires to start a communication it pulls the \textit{SDA} signal to low, alerting all the other devices. The first sequence of 8 bits consists in a 7-bit address and the last bit, \textit{R/W}, informing if data is being requested or is going to be sent. The receiving device must pull the \textit{SDA} line low if the message is successfully received, the \textit{ACK} bit. Data can then be transferred between master and slave until the \textit{SDA} signal is pulled to high.
## PCB Design

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Autodesk® EAGLE™8</td>
<td>25</td>
</tr>
<tr>
<td>3.2 Schematic</td>
<td>26</td>
</tr>
<tr>
<td>3.3 Board</td>
<td>28</td>
</tr>
<tr>
<td>3.4 Manufacturing</td>
<td>30</td>
</tr>
</tbody>
</table>
The Printed Circuit Board (PCB) consists of a sequence of layers, where pads and lines are printed to electrically connect the different components together [52]. It assures that the power and signals are correctly sent to the devices soldered to the board. In an Embedded System, the various electronic components are held and connected by the PCB. It is then one of the most important parts of such systems, assuring a proper performance. There are multiple layouts, that can be used in the production of a PCB, but, in this thesis, a common double-sided configuration was chosen.

![Figure 3.1: Layout of a PCB with 2 copper layers.](image)

The figure 3.1 represents the layout of a double sided PCB:

- **Substrate**, usually made of FR4, a composite material made of fiber glass and epoxy resin. It is flame resistant, electrically isolates the two copper layers and gives rigidity and thickness to the board.

- **Copper**, usually with 35µm of thickness, it is the conductor layer and where routes are drawn.

- **Soldermask**, applied over the copper layers. It helps to avoid corrosion, increases the isolation of the board and makes the soldering process easier.

- **Silkscreen**, has no electrical influence on the board but makes it readable. It highlights where components should go and identifies the serial number and version of the PCB.

### 3.1 Autodesk® EAGLE™8

The software used for the design of the PCB was the Autodesk® EAGLE™8.0.2, that has two main cross-dependent processes, the schematic and board design.

- **Schematic**, Figure 3.2 on the left: The different components that will be part of the embedded system are selected and connected to each other. In the end, the whole circuit appears represented with components labelled and in the correct package.
• **Board**, Figure 3.2 on the right: The different components chosen during the Schematic are placed over the board in the desired configuration. In this window, the different layers can be accessed and routes, vias and holes can be created. All other considerations, such as the Ground (GND) connection between layers, and information in the silkscreen, are represented.

After the necessary iterations between the two processes above, the board Gerber Files, explained in Section 3.4.1, can be created and sent to production. The software also offers tools that make possible the creation of specific packages and symbols every time the wanted component is not available in the software libraries (see appendix B.2).

### 3.2 Schematic

The Schematic is a model of the circuit. On it, all components are represented by symbols, connected to each other, the same way they would be in the embedded system. This representation allows the designer to build and troubleshoot the whole system in a more intuitive way. It is easier to understand, for example, if a circuit is in parallel or series, or to calculate the current and voltage across each component.

The embedded system must have a centralized microcontroller, that receives data from the GPS module, through UART, and from the accelerometer through I²C. The information must then be organized, by a specific algorithm described in section 4.1, and sent to the μSD card through SPI. A more detailed explanation of the decisions on the schematic can be found in the Appendix A.

A good practice, during the schematic phase, is to specify each component's manufacturer part number, in order to easily find the datasheet.Datasheets are extremely important for a proper imple-
mentation of the different components on the board. They give the designers a usual implementation guide, suggestions for noise reduction, which pins can be left floating and many other electrical characteristics. A usual suggestion to reduce noise in all components is the use of decoupling capacitors as close as possible to the $V_{cc}$ pins or any other pin that supplies energy to a different block of the component.

### 3.2.1 Power Supply

The system will be used in a car, where a common tool is the lighter adapter that translates the car's 12V to a 5V USB connector. An also common cable is the USB to micro B cable, that serves most of the actual smartphones. Due to this fact, it was chosen to use a female micro B connector to power the board, since it would easily serve any user, namely drivers.

Even if the microcontroller is powered by 5V, the GPS module, accelerometer and $\mu$SD card need to be powered with 3.3V. A voltage regulator was added to overcome this situation, which is explained in Appendix A.3.

### 3.2.2 Level Shifthing

Two different voltages, 5V and 3.3V, are used in the embedded system. It is important to translate them to enable a safe and correct communication between the microcontroller and sensors.

On the datasheets are specified the high-level and low-level input and output voltages. Looking at Table 3.1, can be observed that the minimum $HIGH$ value for output in the accelerometer is lower than the minimum value for input in the microcontroller. This means there would be an ambiguity and the signal could be either interpreted as $HIGH$ or $LOW$, which is not desirable. On the other hand, a 5V output signal from the microcontroller can permanently damage the accelerometer (since it is higher than 3.3V expected by the IC).

<table>
<thead>
<tr>
<th>MMA8451Q (3.3V)</th>
<th>Atmega328P (5V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_{OH(min)}$</td>
<td>2.97 V</td>
</tr>
<tr>
<td>$V_{OL(max)}$</td>
<td>0.99 V</td>
</tr>
<tr>
<td>$V_{IH(min)}$</td>
<td>2.48 V</td>
</tr>
<tr>
<td>$V_{IL(max)}$</td>
<td>0.99 V</td>
</tr>
</tbody>
</table>

---

$V_{OH(min)} =$ Minimum Output Voltage that the IC will use to transfer a high ('1') value.

$V_{OL(max)} =$ Maximum Output Voltage for the IC will use to transfer a low ('0') value.

$V_{IH(min)} =$ Minimum Input Voltage for the IC clearly identify a high ('1') value.

$V_{IL(max)} =$ Maximum Input Voltage for the IC clearly identify a low ('0') value.
For each communication channel with the microcontroller, it was necessary to implement a different Level Shifting method, depending on the protocol used. For a detailed explanation of each method consult Appendix A.5.

3.2.3 LED’s

The LED’s chosen to implement were:

• **PWR**, to show if the microcontroller and the board are receiving energy;

• **RX** and **TX**, to show the serial communication status;

• **LOAD**, because it is a standard debugging LED in Arduino boards and will blink when a program being uploaded to the microcontroller.

3.3 Board

![Figure 3.3: Embedded System Board Top Layer.](image)

The board of the embedded system with all connections and components placement is presented in Figure 3.3, top layer, and Figure 3.4, bottom layer. The general dimensions are 2700x2100 mils (6.858x5.334cm), same height as a credit card with slightly smaller width. The holes have same size and distance between each other as the Arduino Uno [53].
3.3.1 Components Position

Components are displayed in the board taking into account the relative connections they have with each other. Special care is taken with decoupling capacitors, that should be as close as possible to the respective IC pin. The crystal should also be as close as possible to the microcontroller.

For the majority of the components, it was possible to find the package in the libraries of the software, however for the case of the micro B socket, was necessary to design the package with help of the tools in Eagle, shown on the Appendix B.2.

The routing of the components was made with 10mil width routes, taking special attention into making them:

- **As short as possible**, because every route creates a small resistance that increases with length. Due to this fact, long routes will slightly degrade the signal;

- **Rectilinear**, less expensive than curvilinear;

- **45deg lines**, compared to 90 deg they have less chance of signal reflection at high frequencies, which can create phase cancellation. Even not being a problem for this particular device, it still is a best practice to avoid 90deg lines.

A special routing case occurs in the U.FL connector. This RF connector, for a high-frequency signal, connects the external antenna to the GPS module. Due to the signal frequency, a direct or curve route with specific impedance had to be calculated, see Appendix B.1 for more information.
3.4 Manufacturing

3.4.1 Gerber Files and Excellon

In order to ask a manufacturer to print the designed circuit board, it is important to convert the Eagle board file to another format. By doing so the identity of the product is protected and only the necessary information is sent. The most widely used format in the industry is called Gerber files and was developed by Ucamco [54]. It is an image description format that helps make the communication between the CAD and CAM professionals in a secure and accurate way. Gerber distinguishes itself from other non-image data formats by having information about layer order and function and differentiating objects like SMD and via pads. These characteristics make it easier for the manufacturers to understand the intentions of the designers.

For the drilling, another format is usually used, called Excellon [55]. It contains information about the metric system, tool number, drill sizes and coordinates of each drill.

3.4.2 PCB Quotation

After generating these files the board design can be safely sent to a manufacturer. However, an important middle step is the quotation. The quotation is requested by a manufacturer after providing the Gerber Files and specifying additional information such as:

- **Number of copper layers**, in the case of this PCB 2 layers;
- **Electrical Test**, tests for short circuits or unusually high resistance values in a net;
- **Quantity**, the more units are requested, the cheaper each board becomes. For prototyping, like the board in this thesis, usually 2 to 3 units are requested, in case, for example, one of the boards gets damaged during the soldering process;
- **Delivery days**, without critical urgency a 6 days window is enough. The smaller the window given to the manufacturer the more expensive the board will become.

It is given particular interest in ordering specifications inside the manufacturer standard of fabrication since it lowers the price:

- **Copper Thickness**, usually 35$\mu$m since it is a standard;
- **Isolation material**, FR4 is usually standard material;
- **PCB thickness**, standard is 1.6mm however, a quotation was asked also for 0.8mm because of the impedance controlled routing, check Appendix B.1;
• **Solder Mask Color**, Green is the standard;

For the designed PCB it was requested two identical quotations from two different manufacturers. One of the manufacturers asked the double the amount of the other, revealing the importance of evaluating the values of more than one manufacturer.

The value difference between the use of 1.6mm thickness and 0.8mm represented a 30% reduction in the cost. The impedance variance was not high enough to totally compromise the antenna functionality so the 1.6mm option was the selected for the PCB.

### 3.4.3 Components Selection

The different components were mainly searched on the Digi-Key website\(^1\). The main advantage of this website is that it contains a wide variety of filters in the search engine.

All components chosen for this system are currently active (as an opposite of obsolete), Lead-Free and RoHS Compliant, being the last two due to environmental concerns. Restriction of Certain Hazardous Substances (RoHS) is a directive with the objective of restricting the presence of dangerous substances, that are often used in electronic components, like Lead (Pb) or Mercury (Hg) [56].

Another concern on the selection of components is the package. The size, form and how it is connected in the board. There are two ways of connecting components to a PCB:

• **SMD**, the component is mounted and soldered in the same layer of the PCB;

• **Through-Hole Technology (THT)**, the component is mounted in one side/layer and soldered in the other side/layer of the PCB.

### 3.4.4 Soldering

After the arrival of all the components and PCB, was necessary to solder them all together in order to complete the projected embedded system.

Soldering is the process of fixing a component to its respective pads in the board, using melted metal as the joint material, as demonstrated in Figure 3.5, where through-hole components are being soldered to the designed PCB.

As expected all components were fairly easy to solder, with exception of the accelerometer’s QFN package due to its size, pads position (under the IC) and provided soldering specifications [40]. In the end, it was achieved the embedded system in Figure 3.6

\(^1\)https://www.digikey.com/
Figure 3.5: Thesis author soldering Through-Hole pins on the developed board.

Figure 3.6: Final appearance of the projected embedded system.
4 System Set-Up

Contents

4.1 Arduino Setup .................................................. 35
4.2 Embedded System Setup ....................................... 36
4.3 Installation of the Device in a car ............................. 38
The embedded system developed has no serial interface through USB connector. It would be harder to debug and test code in that board compared to an Arduino board. For that reason an Arduino Uno Board [34] was used to test the intended software/firmware before uploading it to the developed system.

For this prototype it was used an Arduino Uno Rev3, a GPS Logger Shield from SparkFun [57] and an Accelerometer MMA8452Q from SparkFun [58].

4.1 Arduino Setup

For the Arduino implementation, instead of using the UART, is going to be used Software Serial. On the Arduino the native UART of the microcontroller is being used by the USB interface, which makes it not always available. On the developed system, this limitation does not exist, so the software serial library is not used. The libraries used in the Arduino code, that are also usable on the developed system, were the following ones:

- TinyGPS [59], a widely used library for interface with GPS modules;
- AltSoftSerial [60], for the software serial;
- SdFat [61], for the interface with the \( \mu SD \) card;
- SparkFun_MMA8452Q [41], for the interface with the accelerometer;

The software was built following the diagram of Figure 4.1. During the Setup, the communications with the GPS module and the accelerometer are initialized and it is checked if there is a \( \mu SD \) card in the board. The Loop sequence will not start unless a card is inserted. Every Loop sequence, the microcontroller will check if there are available bytes to be read in each sensor. If there are, the data values are updated, so that next time they are saved in the card, they correspond to the most recent values detected. The chosen data rates for the 2 sensors were 10Hz for the accelerometer and 1Hz for the GPS. This means that, considering a vehicle moving at 50km/h to 120km/h, the GPS would log data every 14 to 33 meters and the accelerometer every 1.4 to 3.3 covered meters. For each Loop sequence, if the difference between the last time data was saved in the card, \( \text{LastLogAcc} \) or \( \text{LastLogGPS} \), and the momentary time, \( \text{TimeNow} \), is equal or bigger than the wanted rate, values are saved on the file.

The labels in each GPS and accelerometer file are respectively:

- Latitude, Longitude, Date, Time and Speed;
- X axis, Y axis, Z axis, Date and Time.

The speed is saved in \( m/s \) and the accelerations in 3 axis are in g's (\( 1g = 9.8m/s^2 \)). The maximum g value set in the sensor is 2g, sufficient for non-accident conditions.
Figure 4.1: Diagram of the flow of the software/firmware implemented in the Atmega328P. Setup and Loop are the same functions present in every Arduino Program. The .csv files are all stored in the µSD card.

4.2 Embedded System Setup

To transfer the program to the developed board, a programmer was used, in this case, an Arduino with the ArduinoISP program uploaded. The connection with the developed system was made through the ICSP connector, SPI protocol with extra GND and 5V connections to power the board. The result of this connections is shown in the Figure 4.2.

The Atmega328P, as explained in its datasheet [35], comes with its fuses in certain default values. Fuses are a fourth type of memory where each Fuse byte corresponds to 8 fuses that define functionalities of the microcontroller. In the Atmega328P there are 3 fuses:

- **Fuse Low Byte**, sets up the clock source and start-up time;

- **Fuse High Byte**, turns on the Watchdog timer and can disable the serial programation of the microcontroller, among other functions;

- **Extended Fuse Byte**, activates the Brown-Out Detection and its trigger level.
In order for this system to work properly, it’s crucial to change the value of the Low-Fuse, that is responsible for setting clock source. By default, the clock source is the internal 8MHz oscillator. The clock start up time was maintained at $+65\text{ms}$ but the clock source was set to the external crystal (Low-Fuse = 0xF7).

### 4.2.1 Code Adaptation

The embedded system developed has no need to use a software serial library for the GPS data, since the only hardware UART connection is with the GPS module.

A 10Hz frequency showed only momentary spikes of acceleration that could either be due to noise or actual driver’s behaviour, so to avoid this ambiguity, higher frequencies were tested. At 40Hz was easier to distinguish the noise from the actual maneuvers/behaviours, where acceleration changes were more progressive (instead of a sudden spike). A slightly higher frequency, 50Hz, did not bring more useful information, however the noise was higher, when compared to the 40Hz samples. For these reasons, the accelerometer frequency was set at 40Hz which means an acceleration entry every .35 to 0.83 covered meters (50 to 120km/h).

The accelerometer used in the developed system, $MMA8451Q$ has slight, but crucial, differences to the $MMA8452Q$, used on the SparkFun shield in previous implementation. The differences are highlighted in the Table 4.1:

<table>
<thead>
<tr>
<th>Difference</th>
<th>$MMA8451Q$</th>
<th>$MMA8452Q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Adapting the $SparkFun_{MMA8452Q}$ library to account for the differences in address, ID and resolution, the code uploaded to the board can remain the same.
Table 4.1: Highlight of code related differences between MMA8452Q and MMA8451Q accelerometers.

<table>
<thead>
<tr>
<th></th>
<th>MMA8452Q</th>
<th>MMA8451Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device ID</td>
<td>0x2A</td>
<td>0x1A</td>
</tr>
<tr>
<td>Bit Resolution</td>
<td>12-bit/8-bit</td>
<td>14-bit/8-bit</td>
</tr>
<tr>
<td>Address</td>
<td>0x1D (SA0 = 1)</td>
<td>0x1C (SA0 = 0)</td>
</tr>
</tbody>
</table>

4.3 Installation of the Device in a car

The installation of the device in the vehicle is a critical part in the performance of the embedded system. The position of the board establishes the direction and orientation of the accelerometer’s axis. Also it should be allocated in such a way that unnecessary vibrations and movements, like sliding, are avoided.

The first proposed solution was to use a common cellphone holder arm that grabs onto the front window of the car, as shown in Figure 4.3(a). This solution makes the adjustment of the board orientation easier but does not provide an optimal solution to avoid vibrations. The long arm of the holder is susceptible to considerable vibrations when the car passes through paved roads.

![Figure 4.3](image_url)

(a) Physical Allocation nº1  
(b) Physical Allocation nº2

Figure 4.3: Demonstration of the alternatives for the allocation of the device in a car. The allocation nº1 has the prototype system, composed by the arduino and the GPS logger shield.

The second solution tested was the use of Velcro to fix the board to the car’s dash panel, as shown in Figure 4.3(b). This solution proved simple to implement and overcame previous allocation problems. It was then provided velcro to the different drivers, everytime they would take the device for data collection.
# Initial Data Analysis

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Python Language</td>
<td>41</td>
</tr>
<tr>
<td>5.2 Preliminary Observation</td>
<td>41</td>
</tr>
<tr>
<td>5.3 Correction and Completion</td>
<td>42</td>
</tr>
<tr>
<td>5.4 Smoothing</td>
<td>43</td>
</tr>
<tr>
<td>5.5 Comparison between Butterworth and Savitzky-Golay</td>
<td>47</td>
</tr>
</tbody>
</table>
In this Chapter, the first data samples obtained by the embedded system are observed and corrected. Also, since the acceleration is too noisy to identify patterns, three smoothing techniques were tested.

5.1 Python Language

For the data analysis and machine learning application, one of the most used languages is Python [62]. Python is a high-level programming language created by Guido Van Rossum that offers a clear syntax and a wide variety of open-source modules and packages, which provide time-saving tools for data handling and analysis.

![Python Logo](image)

*Figure 5.1: Python Logo.*

Some of the packages used in this thesis were:

- **NumPy**, for mathematical expressions [63];
- **Pandas**, for data handling and analysis [64];
- **Scipy.signal**, for data smoothing [65];
- **Matplotlib**, for data visualization [66];
- **Scikit-Learn**, for machine learning [67].

5.2 Preliminary Observation

The embedded system, during data acquisition, created in two files, one for the accelerometer data and another one for the GPS module data. The columns of this data as expected are:

- **GPS**: ['Lat', 'Lon', 'Date', 'Time', 'Speed']
- **Acc**: ['X', 'Y', 'Z', 'Date', 'Time']

The GPS module data starts being saved to the card only after getting a $3D – Fix$ or, in other words, when the module has detected enough satellites to obtain its position. This way is made sure all the columns have valid values from the start. The same does not happen with the accelerometer that will
display NaN in the Date and Time columns while there is no 3D-Fix, but is still recording the axis’s values in case some analysis only requires acceleration data.

Every time the system reboots it will append the header, after the last entry, in both files. Every new header will then correspond to a new path taken by the driver. These paths will be labeled as Path and will have a numeric value associated. The columns then become:

1. GPS: ['Lat', 'Lon', 'Date', 'Time', 'Speed', 'Path']
2. Acc: ['X', 'Y', 'Z', 'Date', 'Time', 'Path']

From the following data head, it can be observed that a whole row, that should have a time value 17:03:18, is missing. Since the programmed frequency for the GPS values was 1Hz, it means there was no new information to be saved on that specific second.

<table>
<thead>
<tr>
<th></th>
<th>Lat(degrees)</th>
<th>Lon(degrees)</th>
<th>Date</th>
<th>Time</th>
<th>Speed(m/s)</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>41.172025</td>
<td>-8.679923</td>
<td>2017-05-15</td>
<td>17:03:15</td>
<td>0.05</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>41.172028</td>
<td>-8.679920</td>
<td>2017-05-15</td>
<td>17:03:16</td>
<td>0.07</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>41.172035</td>
<td>-8.679913</td>
<td>2017-05-15</td>
<td>17:03:17</td>
<td>0.10</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>41.172040</td>
<td>-8.679912</td>
<td>2017-05-15</td>
<td>17:03:19</td>
<td>0.14</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>41.172043</td>
<td>-8.679908</td>
<td>2017-05-15</td>
<td>17:03:20</td>
<td>0.07</td>
<td>0.0</td>
</tr>
</tbody>
</table>

This can be due to many external factors, such as entrance in a tunnel or even an atmospheric interference that makes it harder for the antenna in the GPS module to receive information from a sufficient number of satellites. A sensor fusion algorithm could be used to mitigate this situation, however, this utility is beyond the scope of this thesis. So a correction/completion approach is going to be taken.

5.3 Correction and Completion

Even considering the data is correct most of the time, it is important to make sure that any unexpected data output by the GPS or accelerometer is corrected since it will influence the quality of the data.

For the GPS data, missing timestamps will be added using the mean of the Latitude, Longitude and Speed of the timestamps exactly before and after.

Even if the accelerometer data does not have the same problems as the GPS, because the values have to be relatable to the position and velocity, it is important to keep the timestamp correct on the accelerometer file. In case a time stamp is missed by the GPS module, the accelerometer would continue logging but with the last recorded timestamp. To correct this situation the wrong time stamp was replaced by the expected value, taking into account the previous Time value.
5.4 Smoothing

The data collected by the accelerometer at 40Hz is noisy, as shown in Figure 5.2, where each axis corresponds to the illustrated in Figure 5.3. Some of the main factors are the vibration created by the car's motor and pavement irregularities. In order to mitigate the noise, smoothing and filtering techniques will be tested. The objective of smoothing is to achieve a signal as free of noise as possible while preserving and making patterns stand-out, in other words, to reduce the Signal-to-Noise ratio (SNR), described by Equation 5.1, without accentuated attenuation or distortions on the signal.

\[ SNR = \frac{P_{\text{signal}}}{P_{\text{noise}}} \quad (5.1) \]
5.4.1 Butterworth Filter

One widely used low-pass digital filter is the Butterworth filter [68]. It has a frequency response as flat as mathematically possible until the cutoff frequency, followed by a transition band, that is as small as higher the filter order is.

![Figure 5.4: Frequency Domain of the X axis acceleration data.](image)

![Figure 5.5: Comparison between raw accelerometer data and data filtered by a 9th order Butterworth filter with cuttoff frequency of 1Hz.](image)

One of the most important parameters of the low-pass filter is then the cutoff frequency. This frequency should be as low as possible in order to filter most of the noise. However, it should be high enough, so that it preserves all the variations in the signal without distortion.

From all the events examined in the collected data, sudden brakes present a higher frequency. For this reason, this event was examined on a frequency domain, as presented in Figure 5.4. The event
has a duration of approximately 1 second and taking a look at the zoomed frequency spectrum most of the relevant frequencies are under 1Hz, suggesting that a cutoff frequency of 1Hz would be an acceptable value. The 9th order Butterworth filter, since there is no considerable computational cost between different filter orders, with a cutoff frequency \( f_{\text{cutoff}} = 1 \text{Hz} \) was applied to the data through a forward-backward filtering, since all data is available and this technique allows for no phase distortion (non-casual filter), as observable in Figure 5.5.

### 5.4.2 Savitzky-Golay Algorithm

Another smoothing technique that can be used is the method developed by Savitzky and Golay. It proposes the smoothing through the convolution of polynomials that fit successive data windows [69]. Each polynomial is achieved by the least squares method and applied in an equally sized window. The two variables that greatly influence the performance of the algorithm are the polynomial degree and the window size.

![Comparison between raw accelerometer data and Savitzky-Golay filtered data.](image)

**Figure 5.6**: Comparison between raw accelerometer data and Savitzky-Golay filtered data.

A high polynomial degree reduces the distortion of the signal as it increases, however, there is less noise gets filtered. Usual polynomial degrees are low, \( \text{degree} < 5 \), and when it is set to 1, the algorithm works as a Moving-Average.

The window size increases the SNR, however as a consequence, it also increases the distortion of the signal. The response shown in Figure 5.6 was achieved with a window size 47 and a \( \text{degree} = 2 \). The first value was based on the frequency of the device (40Hz) and the second one was obtained after experimentation in different samples of data, considering that the higher the degree value, less noise is filtered.
5.4.3 Custom Douglas–Peucker algorithm

The Douglas-Peucker algorithm objective is to describe a certain curve by a similar one with fewer points [70]. The algorithm recursively finds the point that is further away from a line between the first and last point of a curve segment. The value that limits the maximum allowable distance between each line and point is $\varepsilon > 0$.

The limitation of the Douglas-Peucker algorithm is that it discards all the simplified data, which means that the size of the data changes. Since this is not the desired result, the algorithm was customized, so that every simplified value is approximated to a point in each new curve segment, as represented on the Figure 5.7.

The performance of two different $\varepsilon$ values was tested, as shown in Figure 5.8, and the results are far
from what was desired. Due to the stochastic nature of the noise, small values of $\varepsilon$ cannot characterize the original signal, and for a higher $\varepsilon$ values, the signal gets distorted to the point where much of the data information is lost. Therefore, this algorithm is definitely an undesired option.

5.5 Comparison between Butterworth and Savitzky-Golay

Both Butterworth and Savitzky-Golay alternatives present similar results in the smoothing of the data. However, a detailed window, as in Figure 5.9, shows that the Savitzky-Golay algorithm has slightly more noise when compared to the Butterworth low pass filter.

To better compare these signals, and taking the magnitude bode diagrams from Figure 5.10, the noise energy of both filtered signals was calculated. The total energy of the signal can be obtained from the Parseval's Theorem [71] by the Equation 5.2, where $|F(\omega)|^2$ is the energy spectral density, shows the energy distribution in the frequency domain, and $\omega = 40\pi = 20Hz * 2\pi$ is half the sample frequency.

$$E_{signal} = \frac{1}{\pi} \int_{0}^{40\pi} |F(\omega)|^2 d\omega$$

(5.2)

If from the calculated signal energy is subtracted the energy passed by the filter until the cutoff frequency ($f_{cutoff} = 1Hz = 2\pi$), described by Equation 5.3, it is obtained the noise energy in the signal, by Equation 5.4.

$$E_{low pass} = \frac{1}{\pi} \int_{0}^{2\pi} |F(\omega)|^2 d\omega$$

(5.3)
Figure 5.10: Comparison of the magnitude Bode diagram of the 9th order Butterworth filter with cutoff frequency of 1Hz and Savitzky-Golay algorithm 47/2. The green line represents the point where $f = 1$ Hz.

For both filters, the results are presented in the Table 5.1. Referring to the signal-noise ratio Equation 5.1, and knowing that energy is directly proportional to power, it is possible to infer that the Low-Pass Butterworth filter generates a signal with better SNR when compared to the Savitzky-Golay, becoming the filter used for the rest of this thesis, 9th order Low-Pass Butterworth filter with $f_{cutoff} = 1$ Hz.

Table 5.1: Noise energy comparison in both low-pass techniques with the same data sample.

<table>
<thead>
<tr>
<th>Algorithm/Filter</th>
<th>Savitzky-Golay</th>
<th>Low-Pass Butterworth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Energy</td>
<td>9.52 J</td>
<td>1.93 J</td>
</tr>
</tbody>
</table>
# Dataset Characteristics and Labeling

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Driver Maneuver Labeling</td>
<td>51</td>
</tr>
<tr>
<td>6.2 Dataset Characteristics</td>
<td>54</td>
</tr>
<tr>
<td>6.3 Data Transformation and Features</td>
<td>54</td>
</tr>
</tbody>
</table>
So far, the data has been collected by the different drivers in their own cars, and it was already corrected, completed and smoothed. In this chapter, the data will be Labeled, with matter to the type of maneuver. Unlike some of the studies analyzed in Section 1.3, the data used in this thesis comes from real-world driving conditions, meaning, it was not in a controlled circuit or simulator. Each driver took a different route, encountering situations of low-speed traffic, high speed in highways and also pavements with relative bad conditions. In the end data from 4 drivers, and about 7 hours or 250km of data were considered to contain enough variety to build a flexible model.

6.1 Driver Maneuver Labeling

The focus of the labelling is on the driver. The maneuvers do not necessarily classify a certain route taken by the driver, but rather, classify what forces were applied to the vehicle. The most obvious case of this distinction is in high traffic situations, where, due to low speed, the forces applied to the car are not enough to identify a curve but rather identify multiple braking and acceleration occurrences. To have a more informed decision of what maneuvers should be considered, it was analyzed the Boletim Estatístico de Acidentes de Viação (BEAV) [72], kindly recommended by the Autoridade Nacional Segurança Rodoviária (ANSR) [73]. For the particular case of this thesis, the attention was focused on two points, the E3 (Actions and Maneuvers before an Accident) and in E4 (Complementary information to the actions and maneuvers).

![Figure 6.1: Number of drivers that lost their lives in 2016 compared by type of maneuver](image)

6.1.1 E3 Actions and Maneuvers before an Accident

In the Annual Report of 2016 [74], the most fatal maneuvers are highlighted in Figure 6.1. By far, most fatalities are during normal driving, which can include curves, followed by left maneuvers such
as turn and overtake of another vehicle. This statistic reinforces the idea that maneuvers should be distinguished between right and left. Stopped or parked refers to when the car is completely stopped with engine turned off and not to situations with traffic lights or high traffic, as explained in the documentation for the filing of the BEAV.

### 6.1.2 E4 Complementary information to the actions and maneuvers

Observing the information on Figure 6.2 is possible to conclude that "Excessive Velocity for the existing conditions" is the most common cause of driver death. So, instead of using Overspeed as a metric for dangerous behaviour, as Castiganni et al. [21] did, it is going to be used the "excessive velocity" as a metric. The main reason behind this decision is that the calculation of the overspeed without using Google Maps APIs Premium Plan [75], can become a complex problem, that was considered out of the scope of this thesis.

![2016: Driver deaths according to complementary information (\%)](image)

**Figure 6.2:** Number of drivers that lost their lives in 2016 compared by complementary information [74](adapted).

The view presented in this thesis suggests that excessive velocity for existing conditions in a maneuver can be detected through high acceleration values. Acceleration of the vehicle represents the force on its mass, as well as the derivative of the velocity in function of time, as described in Equation 6.1. For longitudinal accelerations, sudden braking and accelerations will reflect themselves as high absolute longitudinal acceleration values, since the velocity changes significantly in a slow amount of time.

In curves, taking Figure 6.3, it is possible to infer that the lateral acceleration depends on the velocity of the vehicle, being directly proportional to it considering constant radius, as seen in Equation 6.2. The higher the velocity, the higher the acceleration will be. It can be said that a high absolute lateral acceleration either in small or large radius maneuver will flag if the speed excessive for the maneuver. For example, on a highway, even if a car is going up to 100km/h the lateral acceleration on a soft curve would be no higher than 0.2g's, as that in a small curve in a city an speed around 70km/h could flag
lateral accelerations up to 0.5g’s.

\[
\vec{a} = \frac{\vec{F}}{m} = \frac{dv}{dt} \tag{6.1}
\]

\[
a_y = \omega^2 r = \frac{v^2}{r} \tag{6.2}
\]

Figure 6.3: Relationships between longitudinal and lateral acceleration with speed during a curve [22].

6.1.3 Maneuvers and Labels

For the identification and manual labelling of the maneuvers and events were mainly used the X and Y axis data from the accelerometer, with help of the position and speed given by the GPS data. The accelerometer, as shown in papers such as Yu et al. [24], usually gives enough information to identify an occurrence, however with uncertain decisions, such as between Left Turn and Left Curve, it can only be clearly determined with the vehicle position. In the case of this thesis, every time, through the acceleration data, an event was identified, it would be correlated to the GPS entries with the same time and date and, with help of the Google Maps [76], it would be possible to identify and label the maneuver with higher certainty. It is then important to notice that only maneuvers and events that are possible to be supervised, with the GPS and accelerometer data, can be considered. For example, a camera for road tracking would have been necessary to confirm maneuvers such as Lane Change and Overtaking, therefore they can not be considered for this thesis.

The selected maneuvers and labels are represented on the Table 6.1. Based on the data collected, low acceleration is considered to be any value under \(0.1 \text{g} \simeq 1 \text{m/s}^2\). It is important to notice that the “Normal” Label, does not necessarily mean the driver is not going through a Curve or Roundabout, it only means that there is not any high acceleration value detected. The Figure 6.4 illustrates examples of identified and labelled maneuvers. The shaded region demonstrates the whole time considered part of the maneuver. Each occurrence includes the beginning and recovery of the event, in other words, when the acceleration (either X or Y axis) starts growing (usually from 0m/s²) until it returns to a “stable”
position (usually around 0 m/s²) or begins another maneuver.

Figure 6.4: Example of classification of with 3 different maneuvers. Shaded regions show the time considered part of the maneuver.

### 6.2 Dataset Characteristics

The cars used in this dataset were different but all common mid-price category B1 vehicles. The data comes from 4 different drivers between 25 and 35 years old, 1 female, 3 males. All of them spent the same time driving, making a total of about 27300 seconds or 7 hours and half of the data.

Table 6.2 represents the number of maneuvers, with their mean time. Based on the mean time, is possible to estimate, that half of the dataset corresponds to normal low acceleration driving.

### 6.3 Data Transformation and Features

Data transformation is the process of statically describing the data in order to accentuate its characteristics. One way of achieving this is by taking a window of raw data and describing it as a single data entry through features, let's call this new entry in the transformed data, transformed data point. This whole process will help the machine learning algorithms identify patterns more effectively when compared to a stream of raw data.

Recalling the observations at the Section 5.4, the faster maneuver takes around 1 second, so the data could be divided in 1-second windows, called short frames, that would correspond to a transformed
Table 6.1: Maneuvers and correspondent numeric labels.

<table>
<thead>
<tr>
<th>Maneuver/Event</th>
<th>Observations</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Driving with low longitudinal and lateral accelerations.</td>
<td>0</td>
</tr>
<tr>
<td>Left Curve</td>
<td>Includes possible Lane Changes and Overtakes.</td>
<td>1</td>
</tr>
<tr>
<td>Right Curve</td>
<td>Includes possible Lane Changes and Overtakes.</td>
<td>2</td>
</tr>
<tr>
<td>Left Turn</td>
<td>When the driver has to change direction in a road intersection. Usually there are high longitudinal accelerations, since the car has to stop or slow down before the maneuver.</td>
<td>3</td>
</tr>
<tr>
<td>Right Turn</td>
<td>When the driver has to change direction in a road intersection. Usually there are high longitudinal accelerations, since the car has to stop or slow down before the maneuver.</td>
<td>4</td>
</tr>
<tr>
<td>Sudden Brake</td>
<td>Everytime the driver has to reduce the velocity in a relatively short amount of time. Common before intersections and high traffic situations.</td>
<td>5</td>
</tr>
<tr>
<td>Sudden Acceleration</td>
<td>Everytime the driver has to increases the velocity in a short amount in a short amount of time. Common after intersections and high traffic situations.</td>
<td>6</td>
</tr>
<tr>
<td>Roundabout</td>
<td>Special maneuver composed by two right turns and one left turn.</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 6.2: Total number of maneuvers of each type in the dataset with respective mean time.

<table>
<thead>
<tr>
<th>Maneuver</th>
<th>Number of Occurences</th>
<th>Mean Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left Curve</td>
<td>342</td>
<td>10.56</td>
</tr>
<tr>
<td>Right Curve</td>
<td>275</td>
<td>12.02</td>
</tr>
<tr>
<td>Left Turn</td>
<td>52</td>
<td>9.37</td>
</tr>
<tr>
<td>Right Turn</td>
<td>63</td>
<td>11.06</td>
</tr>
<tr>
<td>Brake Event</td>
<td>209</td>
<td>6.34</td>
</tr>
<tr>
<td>Acc Event</td>
<td>286</td>
<td>6.44</td>
</tr>
<tr>
<td>Roundabout</td>
<td>59</td>
<td>19.55</td>
</tr>
<tr>
<td>Total</td>
<td>1203</td>
<td>10.76</td>
</tr>
</tbody>
</table>

data point. Each frame would contain features of both acceleration axis, X and Y, and Speed. The Z axis information was not considered since it is more relevant in pavement classifications. The Mean, Maximum, Minimum and Standard Deviation was considered the relevant features to characterize the data, as they are widely used on the observed papers of Section 1.3. The transformed data point will also contain a label of the maneuver that is more common on the short frame (1-second window).

In Section 6.2 was identified that most maneuvers take in average 11 seconds, so it would be useful to add this information during the data transformation. After experimenting some approaches was decided to create a second frame, called long frame, that would contain the data from the same second as the short frame as well as 5 seconds before and after. As illustrated in Figure 6.5, instead of transforming the data only with the short frame, each transformed data point will be characterized by the set of features from the short frame plus the features from the long frame.
Figure 6.5: Data Transformation. After transformation, each frame of data will contain 21 features. 9 from a 1s window and the other 12 from an 11s window.

The features in each frame are presented in the Table 6.3. The Speed was acquired through the GPS module, that collects data at 1Hz, for this reason, it does not make sense to consider a standard deviation, maximum or minimum value. The only speed data entry each second was opted to be included in the frame labelled as a "Mean" for convenience.

Table 6.3: Features Selected to characterize each time frame. Acc. or Acceleration includes the X and Y axis.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Maximum</th>
<th>Minimum</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Frame (1s)</td>
<td>Acc. and Speed</td>
<td>Acceleration</td>
<td>Acceleration</td>
<td>Acceleration</td>
</tr>
<tr>
<td>Long Frame (11s)</td>
<td>Acc. and Speed</td>
<td>Acc. and Speed</td>
<td>Acc. and Speed</td>
<td>Acc. and Speed</td>
</tr>
</tbody>
</table>
Machine Learning Algorithms for Maneuver Classification

Contents

7.1 Bias-Variance Trade-Off ................................................. 59
7.2 Validation and Learning Curves ........................................ 60
7.3 Support Vector Machines - Linear ..................................... 60
7.4 Support Vector Machines - RBF Kernel ................................ 62
7.5 K-Nearest Neighbours .................................................... 63
7.6 Random Forest ........................................................... 65
7.7 Confusion Matrix .......................................................... 67
7.8 Visualization of the Classification ...................................... 68
Based on the algorithms surveyed in the state of the art and listed in Table 1.1, in this thesis will be given focus to the Support Vector Machines algorithm when building the classification model, since it is one of the most widely used algorithms currently, and deep learning would need a larger dataset.

In the beginning of this chapter, some relevant machine learning concepts will be introduced and then four models are going to be compared. The SVM performance will be compared between linear and RBF kernel and then two other non-linear machine learning algorithms, K-NN and Random Forest will be tested.

7.1 Bias-Variance Trade-Off

Overfitting means that a model performing well in the training data will perform poorly with unseen data. In other words, it is a model with high variance. To avoid overfitting, the complexity of the model can be reduced, for example by feature selection.

Underfitting means that a model also performs poorly on unseen data, however due to a high bias. Bias is the difference between the expected output value by the model and the true value. If it is equal to 0, then the model predicts the true value every time. The Figure 7.1 shows how the model evolves between these two states as a function of capacity, that is described as the model’s ability to fit a greater range of functions [77]. The trade-off is obtained by tuning the complexity of each model through the algorithm parameters.

7.1.1 Stratified K-Fold

To make a decision about validation and accuracy of the following Machine Learning (ML) algorithms, the dataset is going to be separated into training and test sets. This ensures that the models are tested in unseen data. To ensure consistency in the estimate of performance and reliable information about the bias and variance of the model, a k-fold cross-validation is going to be used. This method splits
the training set in \( k \) different folds, without replacement. Trains the algorithm with \( k-1 \) folds and tests in 1 fold, obtaining \( k \) accuracy values. The final performance will be a mean of all the models obtained. The difference to the stratified k-Cross is that on the latter one, the class proportions are preserved, obtaining better results when, like the case of this thesis, the class proportions are unequal [79]. The standard value \( k=10 \) is used in this thesis.

### 7.2 Validation and Learning Curves

Validation and Learning curves both help diagnose and address common issues on models, such as overfitting and underfitting. Making use of the stratified k-fold two curves are plotted, one corresponding to the training accuracy (values obtained with the training folds) and another the validation accuracy (values obtained with the test folds). In the case of the Learning Curve, they are plotted in function of the training samples, as shown in Figure 7.4 and Validation Curve in function of some parameter to be tuned, as shown in Figure 7.3. If there is a large distance between the two curves, it means the model is overfitting, since the accuracy of the training samples is higher than the value obtained with the test samples. In another hand, if both curves are close but both with an accuracy value lower than desired, it can be concluded that the model is underfitted. The shaded region above the curves illustrates the variance of the values along the different folds.

![Validation Curve: Linear SVM C value](image)

**Figure 7.3:** Validation Curve showing the Linear Support Vectors Machine model accuracy in function of value \( C \), related to the slack variable. kernel='linear', decision_function_shape='one-vs.-one'

### 7.3 Support Vector Machines - Linear

It is suspected that the data is not linearly separable, because of conflicts, for example, between classes such as Roundabout, Left Curve and Left Turn that have similar characteristics, also there is
high chance that misclassifications are present in the dataset. Still, before taking a non-linear approach, a linear model will be tested, since linear models are computationally less expensive. One widely used algorithm that can be used to test this hypothesis is the SVM. Support Vector Machine (SVM) is an influential and intuitive algorithm for classification that focus on maximizing the margin when selecting which line, like the one described in Equation 7.1, better separates the classes [80], as illustrated in Figure 7.10. This approach enables a lower generalization error, as small margins tend to overfit. The algorithm will be implemented, again, with the Sklearn Library [81].

$$y(X) = w^T x + b$$ (7.1)

Considering a positive and negative hyperplanes parallel to the decision boundary defined in Equations 7.2 and 7.3, it is possible to, after subtraction and normalization by the vector $w$ length, $\|w\|$, obtain the Equation 7.4, that can be interpreted as the margin.

$$w_0 + w^T x_{pos} = 1 \quad (7.2)$$
$$w_0 + w^T x_{neg} = -1 \quad (7.3)$$
$$\frac{w^T (x_{pos} - x_{neg})}{\|w\|} = \frac{2}{\|w\|} \quad (7.4)$$

The objective should be to maximize $\frac{2}{\|w\|}$, or minimize the reciprocal $\frac{1}{2} \|w\|^2$. However, since the data has most likely misclassifications, the slack variable, $\xi$, will be introduced, in order to allow optimization in the presence of the labelling errors. The parameter $C$ will control the allowable penalization as shown in Equation 7.5. The larger the parameter $C$ value is, the more misclassifications are allowed in each class.

$$\frac{1}{2} \|w\|^2 + C(\sum \xi) \quad (7.5)$$

Before evaluating the results of the parameter tuning and because this is a multiclass classification, it is important to explain the decision function shape, for which there are two options, one-vs.-rest and one-vs.-one. On the first one, it is created a classifier for each class, and each class is compared to the rest of the classes as a whole. The second approach creates a classifier for each pair of classes. The chosen method is then computationally more expensive, because for $N$ classes, $N \times \left(\frac{N - 1}{2}\right)$ classifiers are created instead of $N$, however it performs better on unbalanced datasets.

The $C$ parameter can be tuned, taking into account both accuracy and bias-variance trade-off through the Validation Curve shown in Figure 7.3. It is possible to verify that there are no significant changes along the curves, no overfitting and probably. Figure 7.4 confirms that a linear model can effectively
classify the maneuvers. A non-linear algorithm, SVM with RBF kernel will be tested to verify if the accuracy is improved.

### 7.4 Support Vector Machines - RBF Kernel

One of the great advantages in the use of SVM is that it can be adapted to non-linear problems. It has the ability to transform the data in a higher dimensional form through a mapping function \( \phi(\cdot) \), for training and classification. This transformation is computationally simplified by the kernel function, described on Equation 7.6, which avoids the expensive dot products between training \( \phi(x^{(i)}) \), transformation of \( x^{(i)} \) and unseen data \( \phi(x) \), transformation of \( x \).

\[
k(x, x^{(i)}) = \phi(x).\phi(x^{(i)})
\]

The most widely used kernel is the Radial Basis Function (RBF) kernel, in Equation 7.7, where the parameter \( \gamma = \frac{1}{2\sigma^2} \) and \( \sigma \) is a free parameter for tuning. A more intuitive way of understanding the kernel is as a similarity function, where, if the compared samples are identical the value will be 1 and if completely different, will be close to 0. This means that a higher \( \gamma \) force dissimilar samples to a 0, resulting in a tighter decision boundary.

\[
k(x, x^{(i)}) = \exp(-\gamma \|x - x^{(i)}\|^2)
\]

Predictions can the be made with the function eq.7.8, where \( \alpha \) is a coefficients vector.
The first parameter to be observed the accuracy influence was the $C$, as presented in Figure 7.6. Observing the curves, it can be noticed that the validation accuracy tends to around 0.90, after which it starts decreasing. However being close to such accuracy values comes with the cost of a high variance. There is a high chance of misclassifications in the dataset so the parameter $C$ should have a reasonably large value, to allow misclassification on the classes, while avoiding a high variance. For the validation curve of the $\gamma$ parameter was used a $C=4$, as shown in Figure 7.6. It can be clearly be observed that for high $\gamma$ values the boundary get’s tighter, resulting on an exponentially higher overfit and that a value around $\gamma=0.05$ seems to guarantee a good bias-variance trade-off. A finer tune of both parameters proceeded with grid search and multiple validation curves. It was considered values with a good bias-variance trade-off, $C=4$ and $\gamma=0.04$, as seen on the learning curve of Figure 7.7.

The non-linear model seems to have a more interesting performance compared to the linear model, also built with SVM. Then it seems more interesting to further test other non-linear algorithms such as K-NN and Random Forest.

### 7.5 K-Nearest Neighbours

K-Nearest Neighbours is one of the simplest supervised machine learning algorithms and it can be used either for classification and regression. This algorithm does not build a model during the training, but rather saves the different instances of the training data. The classification of a new element is made...
by the amount of nearest neighbour values of a certain class. The class with more representatives from the neighbours is then chosen for the considered element. Instead of using a naive neighbour search, so-called "brute force", that does not perform well for datasets of this size so a tree-based data structure approach was used, in this case, KD-tree. KD-Tree is a binary tree that creates partitions on each axis, avoiding high dimensional distances computations. A new data point is compared along each partition from the root until a leaf node. It will not necessarily find the closest value, however, it is a good approximation for low-dimensional searches.

In this thesis, the weight of each neighbour is going to be proportional to the inverse of the distance. Also the implementation is going to be made with the Sklearn library [82].

The value of \( K \) influences greatly the results since a small value can make the classification bound-
Figure 7.8: Learning Curve showing the $K$-NN model performance improvement as the size training samples grows. Nº of neighbours $K = 8$, NN algorithm = "kd tree", weights= "uniform".

aries clearer but the use of a large value will help suppress the noise in the samples. To tune the value of neighbours was used a validation curve. The performance of the model had a good bias-variance trade-off around $K=8$. The learning curve of Figure 7.8 shows a continuous growth of accuracy as the number of training samples grows. This fact makes it reasonable to accept that increasing the size of the dataset would result in better results.

Figure 7.9: Simplified scheme of the Random Forest Algorithm [83].

Figure 7.10: The training samples circled are the support vectors and the distance between two classes of support vectors is the margin. [81] (adapted)

7.6 Random Forest

Random Forest is a classifier based on multiple decision trees [84]. It is also called an ensemble model as it combines weaker learners in order to create a stronger and more robust model, less prone to
overfit. Each decision tree will make a decision upon one class and the most popular will be considered for the output, as illustrated in Figure 7.9. In this thesis the algorithm is implemented with the Sklearn library [85].

For each decision tree, \( N \) samples are randomly selected from the training data, with replacement, so that each tree is exposed to a different part of the data. From the total of \( M \) features, \( m \) features are selected at random, these are the ones considered when searching for the best split to generate two children nodes. The value \( m \) should be adjusted, however, as stated by Breiman [84], the algorithm will achieve the same results with one or two features. A parameter that can be used to reduce the variance is the minimum number of samples per child node in the trees, whose Validation Curve as shown in Figure 7.11. This number is responsible for limiting the depth of the trees, the smaller it gets, the more
specific will be each tree and the larger the more general they will become. The values estimated through the validation curves were fine-tuned through grid search and then the model Learning Curve of Figure 7.12 obtained.

### 7.7 Confusion Matrix

Until now the models were evaluated considering the accuracy, however there are other equally important metrics. Taking into account the confusion matrix in Figure 7.13, it is possible to more clearly describe what is the accuracy, by the Equation 7.13.

<table>
<thead>
<tr>
<th>Predicted class</th>
<th>( P )</th>
<th>( N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( P )</td>
<td>True Positives (TP)</td>
<td>False Negatives (FN)</td>
</tr>
<tr>
<td>( N )</td>
<td>False Positives (FP)</td>
<td>True Negatives (TN)</td>
</tr>
</tbody>
</table>

\[
FPR = \frac{FP}{FP + TN} \quad (7.9)
\]

\[
TPR = REC = \frac{TP}{FN + TP} \quad (7.10)
\]

\[
PRE = \frac{TP}{FP + TP} \quad (7.11)
\]

\[
F1 = \frac{2 \times PRE \times REC}{PRE + REC} \quad (7.12)
\]

\[
ACC = \frac{TP + TN}{FP + FN + TP + TN} = 1 - \text{error} \quad (7.13)
\]

Other metrics, such as False Positive Rate, True Positive Rate or Recall, and Precision, described by Equations 7.9, 7.10 and 7.11 respectively, are also important, since they give a better understanding of how useful the model is, particularly on unbalanced datasets. Since the problem is multiclass a different calculation of the precision has to be made, called macro averaging method, shown on Equation 7.14. This method is a simple mean of the precision of all \( k \) classes that makes possible to weight all classes equally. In the equation, \( PRE_k \) corresponds to the precision on the classification of the event with label \( k \) against the all the other \( k - 1 \) classes as a whole. Through this averaging method all the classes precision is weighted equally.

\[
PRE_{macro} = \frac{PRE_1 + ... + PRE_k}{k} \quad (7.14)
\]

The F-1 Score makes a harmonic mean between the precision and recall, giving a conservative average value between both. Like the accuracy, the closer to 1 the better the model is. The Table 7.1 compares the four models built above with respect to accuracy, precision, recall and F1 score.
The accuracy values obtained from the test set show that the four models have a similar performance, however, when compared the Precision and Recall the SVM with 'RBF' kernel shows a clear superiority, with 10% more in the F1-Score than the other two (K-NN and Random Forest). It is interesting to notice that the linear SVM model obtained results very close to the other two non-linear machine learning algorithms. For future work would be interesting to explore other linear models.

Table 7.1: Comparison of the 4 models, through Accuracy, Precision, Recall and F1-Score obtained with the test set.

<table>
<thead>
<tr>
<th>Model Algorithm</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-NN</td>
<td>87.994</td>
<td>83.779</td>
<td>59.844</td>
<td>67.901</td>
</tr>
<tr>
<td>Random Forest</td>
<td>87.665</td>
<td>78.142</td>
<td>61.829</td>
<td>68.390</td>
</tr>
<tr>
<td>SVM (Linear)</td>
<td>86.072</td>
<td>77.721</td>
<td>59.479</td>
<td>66.675</td>
</tr>
<tr>
<td>SVM (RBF Kernel)</td>
<td>89.348</td>
<td>83.754</td>
<td>69.053</td>
<td>75.166</td>
</tr>
</tbody>
</table>

7.8 Visualization of the Classification

As a sanity check that the classifier is working correctly, the model was graphically tested. On the Figure 7.14, a strip of the route taken by the driver is presented with the different maneuvers identified by colours. The identification is mostly accurate, as expected by the accuracy values.

![Figure 7.14: Strip from classified route taken by a unseen driver. Developed with Google Maps API [76].](image-url)
8

Acceleration Categories and Characterization of Drivers

Contents

8.1 Categories ......................................................... 71
8.2 Driver’s Characterization: Radar Graph ..................... 71
8.3 Driver’s Characterization: Web-Based Presentation ...... 72
So far, a data from different drivers have been collected and a maneuver classifier created. In this Chapter is going to be shown how the built model can help characterize different driver. First, three different categories for the maneuvers/events, based on maximum acceleration values, will be introduced. Followed by an explanation of the driver characteristics information shown on a built online site.

8.1 Categories

In this section are described the metrics that categorize the intensity of the maneuvers. With the model created, it is possible to identify maneuvers from a driver during a certain route or an agglomerate of routes, however, this information is more characterizing of the route than the driver itself. To better understand each driver the maneuvers are going to be divided in three categories, according to the accelerations seen during the labeling process and presented on Table 8.1. These categories were also inspired by Bergassa et al. [22], as well as the tests in extreme conditions at [86], where vehicles are evaluated by having an acceleration of 0.7g at constant radius turn and 0.8g during a brake since such events are “among the more severe driving conditions for durability” of the vehicle.

<table>
<thead>
<tr>
<th>Acceleration Label</th>
<th>Low</th>
<th>Mid</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration Range</td>
<td>$acc_{max} &lt; 0.2g$</td>
<td>$0.2g &lt; acc_{max} &lt; 0.5g$</td>
<td>$acc_{max} &gt; 0.5g$</td>
</tr>
</tbody>
</table>

8.2 Driver’s Characterization: Radar Graph

In an attempt to better translate the information of each driver a score for each maneuver was created. Let’s call it “Maneuver Index” and it is defined by the Equation 8.1. The Maneuver Index for each maneuver $m$, $MI_m$, is calculated through a weighted mean of the number of occurrences of the maneuver in each of the three categories described in Section 8.1, $low_m$, medium$_m$, high$_m$, divided by total number of kilometers made by the driver, $Total\_KM$. This way instead of only showing how common a maneuver is on a certain driver, the index has a tendency to be higher or drivers with higher accelerations detected. The weights $w$ can be tuned depending on the situation, but for this case the weights will be $w = [0.1, 0.3, 0.6]$, making high acceleration instances more penalizing than low acceleration.

$$MI_m = \frac{low_m \times w_0 + medium_m \times w_1 + high_m \times w_2}{Total\_KM}$$ (8.1)
To obtain an overall Score of the driver a mean can be done:

\[ \text{Score}_{\text{overall}} = \frac{\sum_{i}^{N} M_{i}}{N} \]  

(8.2)

Analyzing the Radar Graphs of Figure 8.1 is possible to understand that these metrics help characterize driver behaviour and compare them. On the Graph 8.1(a) the difference is obvious between drivers. The blue driver, most probably, encountered more traffic, forcing him to have more sudden accelerations and brake events than the red driver.

From the graphs can be observed that, as the data increases, there is a tendency for the \( \text{Score}_{\text{overall}} \) to become lower. So the comparison between drivers can only be easily made when the number of \( \text{Total}_{KM} \) is similar. On the Graph 8.1(b), the Green driver has a higher acceleration maneuver index and lower Roundabout and Left Curve maneuver index when compared to the Yellow Driver. This can be interpreted as a preference of the Yellow driver to not reduce the speed before such maneuvers, as the other one prefers to accelerate after the maneuvers, braking "softly" before turning.

![Radar Graphs showing maneuver index values for each maneuver to different drivers.](image)

(a) Comparison of two unique drivers with about 20km of data and a \( \text{Score}_{\text{overall}} \approx 0.3 \)  
(b) Comparison of two unique drivers with more than 80km of data and a \( \text{Score}_{\text{overall}} \approx 0.1 \)

Figure 8.1: Radar Graphs showing the maneuver index values for each maneuver to different drivers. The names are fake to maintain the anonymity of the drivers in the study. The graphs were created using Chart.js [87].

### 8.3 Driver’s Characterization: Web-Based Presentation

All the data collected and analyzed with the chosen classifier is displayed on a webpage developed with Django [88], that is a high-level Python web framework and deployed in Heroku [89], a
platform that allows to build and run apps on a cloud. The webpage is called BeDriver and its link: https://bedriver.herokuapp.com/. The idea behind the website is the creation of a more interactive presentation of the results obtained by each driver. To achieve this there are two main divisions, the "Drivers" and the "Map".

8.3.1 BeDriver: Drivers

On the Drivers division is presented a list of all the drivers whose data was collected during the thesis, the identities are fake however all the rest of the data is real. By clicking on the "Details" button is possible to closely investigate each driver, through the information presented, shown in Figures 8.2 and 8.3. All the data were obtained with the model built on this thesis and the Maneuver Indexes are calculated on the webpage. On the profile page of each driver is possible to change or update the maneuver classification results and all the graphs will actualize automatically according to the changes.

![BeDriver](https://yourimageurl.com)

**Figure 8.2:** Driver Profile for the Yellow Driver. Under the name a quick description of the driver is presented. The table shows for each maneuver the number of occurrences in each category and the maneuver index.

![Doughnut Graph](https://yourimageurl.com)

**Figure 8.3:** Doughnut Graph on the left illustrates and compares the number of occurrences of each maneuver. The radar graph displays the maneuver index for each maneuver.
8.3.2 BeDriver: Map

On the Map, it is possible to upload a file generated by the classifier. The Google Maps API will read the file and output the route colour coded, according to each maneuver, as shown in Figure 8.4. For future work would be important to add a label, so that the user could understand which colour corresponds to which maneuver.

Figure 8.4: Map division of the BeDriver webpage, demonstrating a quick strip of a colour coded classified route. Black = Normal, Red = Brake, Orange = Roundabout.
Conclusion

Contents

9.1 Conclusion ................................. 77
9.1 Conclusion

The work developed in this thesis arises in the context of road safety. According to the WHO, this is a global problem that causes millions of deaths around the globe, as well as billions of costs. The main reasons behind such numbers are the driving behaviours and vehicle condition.

Since CEiiA is developing a car sharing service, connected to the MOBI.ME platform, the development of a driver characterization solution was suggested to optimize its current platform. Such solution can be used to signal dangerous driving behaviours, as well as to estimate CO2 emissions and vehicle maintenance costs, that can justify the application of different fares to each driver.

For the collection of data from unique drivers, it was developed an embedded system. This device should be as economical as possible but stay flexible and with easy installation in every user’s vehicle.

A complete PCB was designed with the EAGLE 8.02 software. It was selected an accelerometer, to obtain longitudinal and lateral acceleration at 40Hz, and a GPS module, to obtain velocity and position at 1Hz. A microcontroller was used to receive the data from the sensors and save them, following a specific algorithm, to the µSD card on the board. During the schematic phase of the project, it was implemented level shifting strategies to the different serial protocols and a voltage regulator. All the components were ordered and quotations requested for the PCB. The whole process of design and production of the system was completed with the soldering of the components on the board.

While the embedded board was in production, to perform data acquisition it was used an Arduino platform, with a shield for the sensors. Such solution was also used as a debugging platform for the software/firmware used in the developed embedded system. For end-user applications, considering that the product will be scaled for commercial applications, the developed board is a more economical solution and has a smaller size for installations on the vehicles, when compared to the Arduino implementation.

The collected data, a total of seven hours, from four unique drivers, was analysed using Python language. Three smoothing techniques, Butterworth filter, Savitzky-Golay and adapted Douglas-Peucker were tested in the noisy acceleration data. The option selected was a 9th order low-pass Butterworth filter with $f_{\text{cutoff}} = 1\text{Hz}$, since it has the best noise rejection characteristic.

The maneuvers, acquired from four different drivers, were manually labelled, taking into account the acceleration and position of the vehicle. Labeling is one of the most critical parts of this work. It establishes the patterns that the model should learn, so the finer it is, the better the model would be. To understand if a linear machine learning algorithm could be used, was necessary to check if the data is linearly separable.

In this thesis, the performance of models built with Linear SVM and non-linear machine learning algorithms K-NN, Random Forest and SVM with the RBF kernel, were compared for the first time, in the context of maneuver detection. The three models parameters were tuned with the aid of validation curves, taking special attention to the bias-variance trade-off. The model created with SVM with the RBF
kernel has the highest accuracy (89%) and, clearly the best F1-score from the three models (75%).

The obtained results have a lower accuracy than what was reported on some of the HMM applications. However, a direct comparison of these numbers cannot be made, since in our work the data was obtained in real driving conditions, instead of the simulations used in the other works. This work then proves that it is not only possible to classify different driving maneuvers, but also to achieve it in real conditions, with many unknown variables that cannot be modelled in simulations, most of them of a stochastic nature.

The data from the different drivers, classified by the model is used on a website, built with Django, that makes use of the maneuver index to characterize multiple drivers. The classified events are sorted into three acceleration categories, low, mid and high, corresponding to the maximum acceleration value detected during the maneuver. The scores obtained by each driver can then give a clear idea of the accelerations that the vehicle is exposed.

This thesis successfully shows the complete workflow necessary to the characterization of different drivers. This data is critical to the understanding of the wear of the vehicle and relatable to the emissions of each car (in the case of a fuel engine). In case of an accident, previous data from the driver can be also analyzed and be used as an indicator of aggressive driving. This information could then justify the application of different fares to each driver. These cases show how versatile the information obtained in this thesis can be.

9.1.1 Future work

As future work for this thesis the following items as proposed:

- Incorporation of a GSM module on the embedded system in order to send data directly to a server;
- Development of a sensor fusion solution to improve the quality of the data;
- Collection of more data from different drivers and from different regions, to test the performance of Deep Learning on maneuver classification;
- Creation of a model able to identify each driver of a sharing vehicle, while driving, based on the metrics developed on this paper;
- Incorporation of the whole workflow on the webpage so that anyone could submit their driving data directly.
Bibliography


A.1 Microcontroller

In Figure A.1 the schematic of the Atmega328P and its connections are presented.

The \(0.1\mu F\) capacitors on the \(VCC\) and \(AVCC\) pins, \(C_1\) and \(C_2\) are acting as decoupling capacitors, which is one of the most common applications for this passive components. The power supply signals are usually noisy and can contain high-frequency spikes that might damage sensible ICs or lead to their malfunctioning. These capacitors are connected between the supply voltage and GND, as near as possible to the IC, in order to filter such variances and keep the current as steady as possible. They can also act as a power supply, briefly maintaining the voltage required by the IC when the main power supply voltage drops momentarily.

More than one capacitor can be used so that different frequencies are filtered, which is the case of the GPS module, as shown on Figure A.6.
The connectors add more flexibility to the system. They give access to analogue pins, digital pins and interruption pins from the accelerometer.

The \textit{RESET} pin is pulled up to 5V since bringing it low would result in the reboot of the microcontroller. Such effect can be purposely achieved with the momentary switch \textit{RESET}.

The ICSP Connector, shown in Figure A.2 is a valuable connector that makes it possible to configure and program the Atmega328P.

As explained in Section 2.1, an external crystal is used with two equal value capacitors, illustrated on Figure A.2. Well dimensioned capacitors are important since they assure less time error and noise.

The value of the capacitors depends on the load capacitance present in the datasheet [90]. For the \textit{HC-49US}, a 16MHz crystal with \(\pm 30\) ppm, the load capacitance is \(C_L = 18\mu F\) and the shunt/stay capacitance is \(C_s \leq 7.0\mu F\). Since the capacitors are in parallel, the following formula can give the value
of the capacitors [91]:

\[ C_L = \frac{C_5 \cdot C_6}{C_5 + C_6} + C_s \Rightarrow C_5 = C_6 = 22\text{pF} \quad (A.1) \]

The power source that will supply 5V to the whole embedded system comes from the MicroB USB connector, as shown on Figure A.3. The data pins D1 and D2, since they are not used, are pulled to ground with a resistor to avoid short circuit. The ID pin is left floating since the embedded system will act like a peripheral (“B-device”) of the main device (“A-device”), the car, that supplies the energy.

The other system main components will need 3.3V of supply voltage, so it is important to add a voltage regulator, as illustrated in Figure A.3. In this case, a MIC5219 [92] that outputs a 3.3V from an input of 5V was chosen.

For better visual understanding of the inner state of the microcontroller, four Light Emitting Diode (LED) were chosen and implemented as demonstrated in Figure A.4. The power, PWR, is a green LED that will light up as long as the system is receiving energy. The LOAD is a yellow LED that serves the purpose of easily showing that the board is working or programs are being uploaded properly to the microcontroller. The RXD and TXD will also assure the user that the system is working properly. This two LED use inverse logic because the usual orientation would conflict with the level shifting of the GPS
module.

The high impedance at the input of the amp-op buffer protects electrically the SCK signal in the LED implementation.

A.2 accelerometer

The schematic in Figure A.5 was achieved following the application diagram in figure 4 of the MMA8451Q datasheet [40]. Two resistors were used as pull up in the SCL and SDA, as explained in Section 2.4.3. 4.7kΩ is the value suggested by the manufacturer. SA0 was grounded, forcing the least significant bit of the address of the accelerometer to ‘0’. The interruption pins INT1 and INT2 were connected to the connectors in Figure A.1 through an open jumper, making them accessible if needed. The resistors R9 and R10 are implemented to protect the accelerometer, since by default pins INT1 and INT2 are floating pins.

A.3 GPS Module

For the implementation of the GPS module the datasheet reference design was considered [46]. The decoupling capacitors C14 and C15 with the ferrite bead FB1 reduce the noise and protect the module against harmful frequencies. GPS_FIX LED is used to give a visual information that the module was able to 3D_FIX. When is looking for a fix the LED blinks with 1Hz frequency, and when fixed it stays ON. To allow a quicker boot of the GPS module an external battery, CR1220 is used. The battery keeps the module internal Real Time Clock (RTC) running, which retains previous satellite information, even
when the board has no power supply. This avoids, every time the system is powered up, a cold start, which has a lower sensitivity and can take up to 35s to output 3D-Fixed reading instead of 1 second (with the battery). The Gtop013 module already has an internal antenna, but if it cannot perform well, an external antenna can be plugged into the U.FL connector, X1.

A.4 µSD Socket/Card

The microSD card connector datasheet [93] was used to design the package correctly, following the workflow in Section B.2. The connections for the schematic, however, are quite simple, since it is known that it will communicate through SPI protocol.

Since the connector is a slave, DATAOUT is connected to MISO, DATAIN is connected to MOSI and SCLK is connected to SCK. DAT1 and DAT2 are not needed for SPI protocol and the CD1 and CD2 could be used for detecting the card. But since this four pins not used they were left floating. During the thesis, the µSD card used was from Transcend, with 2GB of memory [94].

A.5 Level Shifting

The microcontroller is 5V powered, however the three other main components, µSD, GPS module and accelerometer work in a CMOS 3.3V power supply.

Due to electrical properties of the three protocols, different forms of level shifting are used:
• SPI, for the microSD a 4-bit voltage translator [95] was used to easily convert the voltage of the four signal lines;

• I\textsuperscript{2}C, the bi-directional protocol has a level shifting specified by the NXP semiconductors, that makes use of MOSFET transistors [96];

• UART, is a simpler protocol, with only two uni-directional signals. For the microcontroller RXD a MOSFET can be used in the same fashion as the NXP semiconductors solution and for the microcontroller TXD a simple voltage divider.

A.6 Complete Schematic of the projected system
Figure A.10: Image of the whole final schematic of the developed embedded system.
B.1 Impedance Controlled Routing

So far, could be considered that a certain voltage is present in the two ends of a route the same time because the frequencies were not substantial when compared to the speed of light \((3 \times 10^8 \text{m/s})\). However when the signal received by the external antenna has a frequency about 1.57GHz (L1 carrier) we need to take special attention to the phase of the signal and impedance of the line. There is always a part of the signal that is reflected along the route, the sharper the corners and bigger the longer the more the integrity of the signal is compromised. So it is important to keep the connection round and as short as possible.

Another way to control this effect is by matching impedances of the GPS module, route and connector. The datasheet of the module [46] refers an impedance of 50Ω, the same as the chosen connector. So it leaves just the route impedance to be adapted, this is called Impedance Controlled Routing [97]. The parameters that change the value of the impedance are shown in the Figure B.1 and Equation B.1.
Figure B.1: Impedance Controlled Routing Parameters \[97\].

\[ Z_0 = \frac{87}{\sqrt{\varepsilon_r + 1.41}} \times \ln(5.98 \times \frac{\text{TraceToPlaneDistance}}{0.8 \times \text{TraceWidth + TraceHeight}}) \] (B.1)

The \( \text{TraceHeight} \) is gonna maintained in the 35\( \mu m \) and a table was obtained making variations with standard \( \text{TraceToPlaneDistance} \) and \( \text{TraceWidth} \). The substrate dielectric, \( \varepsilon_r = 4.6 \), for FR4.

Table B.1: Impedance Controlled Routing Parameters Influence on the impedance value.

<table>
<thead>
<tr>
<th>( Z_0 )</th>
<th>( \text{TraceToPlaneDistance} )</th>
<th>( \text{TraceWidth} )</th>
<th>( \text{TraceHeight} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>112.09</td>
<td>0.8 mm</td>
<td>10 mil</td>
<td>35( \mu m )</td>
</tr>
<tr>
<td>50.96</td>
<td>0.8 mm</td>
<td>56 mil</td>
<td>35( \mu m )</td>
</tr>
<tr>
<td>58.88</td>
<td>1.0 mm</td>
<td>56 mil</td>
<td>35( \mu m )</td>
</tr>
<tr>
<td>65.35</td>
<td>1.2 mm</td>
<td>56 mil</td>
<td>35( \mu m )</td>
</tr>
<tr>
<td>75.56</td>
<td>1.6 mm</td>
<td>56 mil</td>
<td>35( \mu m )</td>
</tr>
</tbody>
</table>

The ideal value will be 56\( \text{mil} \) route width and an FR4 substrate with 0.8\( \text{mm} \), however, is important to keep the values from other options in check for better cost flexibility.

**B.2 Designing a New Component**

In order to create a component first is necessary to check if there are any other similar packages or symbols so that is not necessary to create both. In the case of the Micro USB female connector was opted to create everything from scratch.

As illustrated on the software window on Figure B.2, after creating a new Library, there are 3 relevant items:

- **Device**, Connection between Package and Symbol, with brief description;

- **Package**, figure to use on the board layout, this is the most important part, since the dimensions of the pads and distance between them must be exact, otherwise will be impossible to solder the component on the board. The general dimensions and orientation are also fundamental to properly position the components, for example, the MicroB needs a clear design so that the entrance of the connection faces the outside of the board;
- **Symbol**, figure to use the schematic layout includes names of different pins and respective connections;

![Micro B symbol and package](image)

**Figure B.2:** Micro B symbol and package specially made for the connector with manufacturer part number 10118192 – 0001LF.