Identification of driving manoeuvres using smartphone-based GPS and inertial forces measurement

by

Jarrett Engelbrecht

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in Electrical and Electronic Engineering in the Faculty of Engineering at Stellenbosch University

Department of Electrical and Electronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Supervisors:

Dr. M.J. Booysen Prof. G-J. van Rooyen

April 2015

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 1/12/2014 ....

Abstract

Identification of driving manoeuvres using smartphone-based GPS and inertial forces measurement

J. Engelbrecht

Department of Electrical and Electronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MScEng (E&E)

December 2014

Road accidents are a growing concern for governments and is rising to become one of the leading causes of death in developing countries. Aggressive driving is one of the major causes of road accidents, and it is therefore important to investigate ways to improve people's driving habits. The ubiquitous presence of smartphones provides a new platform on which to implement sensor networks in vehicles, and therefore this thesis focuses on the use of smartphones to monitor a person's driving behaviour. The framework for a smartphone-based system that can detect and classify various driving manoeuvres is researched. As a proof of concept, a system is developed that specifically detects lateral driving manoeuvres and that classifies them as aggressive or not, using a supervised learning classification algorithm. Existing solutions found in research literature are investigated and presented. The best existing solution, a dynamic time warping classification approach, is also implemented and tested. We use an aggressive driving model that is based on the angle of a turn, the lateral force exerted on the vehicle and its speed through the turn. The tests and results of the implemented manoeuvre detection and classification algorithms are presented, and thoroughly discussed. The performance of each classifier is tested using the same data set, and a quantitative comparison are made between them. Ultimately, a lateral driving manoeuvre detection and recognition system was successfully developed, and its potential to be implemented on a smartphone was substantiated. The suitability of supervised learning classifiers for classifying aggressive driving, in comparison to dynamic time warping classification, was successfully demonstrated and used to validate our aggressive driving model. Conceivably, this work can be employed in the future to develop an holistic smartphone-based driver behaviour monitoring system, which can be easily deployed on a large scale to help make the public drive better. This would make our roads safer, reducing the occurrence of road accidents and fatalities.

Uittreksel

Identifikasie van bestuurbewegings met behulp van selfoon-gebaseerde GPS en meting van traagheidskragte

("Identification of driving manoeuvres using smartphone-based GPS and inertial force measurement")

J. Engelbrecht

Departement Elektriese en Elektroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MScIng (E&E)

Desember 2014

Padongelukkige is 'n groeiende bekommernis vir regerings en is een van die hoof oorsake van sterftes in ontwikkelende lande. Aggressiewe bestuur is een van die grootste oorsake van padongelukke, en dit is dus belangrik om ondersoek in te stel oor hoe mense se bestuurgewoontes verbeter kan word. Die alomteenwoordigheid van slimfone bied 'n nuwe platform waarop sensor netwerke geïmplementeer kan word in voertuie. Daarom fokus hierdie tesis op die gebruik van slimfone om 'n persoon se bestuurgedrag te moniteer. Die raamwerk vir 'n slimfoon-gebaseerde stelsel wat verskeie bestuurbewegings kan opspoor en klassifiseer is nagevors. As 'n bewys van die konsep, is 'n stelsel ontwikkel wat spesifiek laterale bestuurbewegings opspoor en dan klassifiseer of dit aggressief is of nie, met behulp van 'n klassifikasie algoritme wat onder toesig geleer is. Bestaande oplossings gevind in navorsingsliteratuur word ondersoek en aangebied. Die beste bestaande oplossing, 'n dinamiese tyd buiging klassifikasie benadering, word ook geïmplementeer en getoets. Ons gebruik 'n aggressiewe bestuurmodel wat gebaseer is op die hoek van 'n draai, die laterale krag wat uitgeofen is op die voertuig en sy spoed deur die draai. Die toetse en die resultate van die geïmplementeer beweging opsporing en klassifisering algoritmes word aangebied, en deeglik bespreek. Die prestasie van elke klassifiseerder is getoets met behulp van dieselfde stel data, en 'n kwantitatiewe vergelyking is tussen beide gemaak. Oplaas is 'n laterale bestuurbeweging bemerking en herkenning stelsel suksesvol ontwikkel en sy potensiaal om geïmplementeer te word op 'n slimfoon is gestaaf. Die geskiktheid van die onder-toesig-geleerde klassifiseerders vir die klassifikasie van aggressiewe bestuur, in vergelyking met dinamiese tyd buiging klassifikasie, was suksesvol gedemonstreer en gebruik om ons aggressiewe bestuurmodel te bewys. Hierdie werk kan in die toekoms gebruik word in 'n holistiese slimfoon-gebaseerde bestuurdergedrag monitering stelsel, wat maklik op groot skaal ontplooi kan word om te help verseker dat die publiek beter bestuur. Dit sal ons paaie veiliger maak, en die voorkoms van padongelukke en sterftes verminder.

Acknowledgements

I would like to express my sincere gratitude to:

My supervisors, for their sound advice and guidance,

MTN, for their financial support,

and my family and friends, for their support and encouragement.

Publications

Work in this thesis has been accepted for publication as follows:

- J. Engelbrecht, M.J. Booysen, and G-J. van Rooyen, “Recognition of driving manoeuvres using smartphone-based inertial and GPS measurement", *1st International Conference on the Use of Mobile Information and Communications Technology in Africa (UMICTA)*, 9–10 Dec 2014, Stellenbosch, South Africa.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations and acronyms

GPS	Global Positioning System
GNSS	Global Navigation Satellite System
GSM	Global System for Mobile communications
WHO	World Health Organization
M2M	Machine-to-machine
OBU	On-Board Unit
CAN	Controller Area Network
ECU	Electronic Control Unit
OBD	On-Board Diagnostics
GLOSA	Green Light Optimal Speed Advisory
LDWS	Lane Departure Warning System
NHTSA	National Highway Traffic Safety Administration
DAS	Driver-Assistance System
ADAS	Advanced Driver-Assistance System
ITS	Intelligent Transportation Systems
USB	Universal Serial Bus
SMA	Simple Moving Average
LPF	Low Pass Filter
BPF	Band Pass Filter
HBF	High Pass Filter
FIR	Finite Impulse Response
DTW	Dynamic Time Warping
ML	Maximum Likelihood
LDA	Linear Discriminant Analysis
DLDA	Diagonal Linear Discriminant Analysis
IMU	Inertial Measurement Unit
DFT	Discrete Fourier Transform
SD	Secure Digital or Standard Deviation
OS	Operating System
TP	True Positive
TN	True Negative
FP	False Positive

FN	False Negative
Hz	Hertz
mHz	milli-Hertz
km/h	Kilometers per hour
m/s^2	Meters per second per second
rad/s	Radians per second

List of symbols used

g	Acceleration of gravity (9.81 m/s^2)
$a_{x,y,z}$	Acceleration in the x, y, or z direction (m/s^2)
$\omega_{x,y,z}$	Rotation rate in the x, y, or z direction (rad/s)
$\mathbf{a}$	Acceleration signal vector
$\boldsymbol{\omega}$	Rotation rate signal vector
λ	Acceleration threshold
θ_x	Yaw
θ_y	Pitch
θ_z	Roll
θ_{pre}	Pre-rotation
θ_{tilt}	Tilt-rotation
θ_{post}	Post-rotation
$E_{\mathbf{a}}$	Acceleration signal energy
h	Bend severity
v	Speed
$\mathbf{x}_k$	State variable
$\mathbf{z}_k$	Measurement variable
$\mathbf{w}_k$	Process noise
$\mathbf{v}_k$	Measurement noise
β	Kaiser window shape parameter
f_0	Fundamental frequency
$\mathcal{F}$	Feature space
c	Local cost measure
C	Cost matrix
p	Warping path
p^*	Optimal warping path
$\boldsymbol{\theta}$	Parameter set of statistical model

Chapter 1

Introduction

Most modern smartphones have a variety of embedded sensors — typically an accelerometer, gyroscope, light, proximity and magnetic sensors, as well as a microphone, camera and Global Positioning System (GPS). This variety of sensors make many sensing applications possible. An example of such an application is gesture recognition, which is used to answer a call when bringing the phone to one's ear, or paging through a document by the wave of a hand [1, 2]. In a similar way, different activities such as walking, running, cycling and driving can be detected and classified using the inertial sensors of a phone that is carried in a user's pocket [3]. In this project we specifically investigate how lateral driving manoeuvres can be identified by exploiting a smartphone's embedded sensors.

1.1 Road safety

Worldwide, more than a million deaths are caused by road accidents per year [4]. The World Health Organization (WHO) predicts that road fatalities will rise to become the fifth leading cause of death by 2030 [4]. Research done in the United States shows that, in more than 50% of fatal road accidents, unsafe driving behaviours were involved [5]. Road accidents are caused by a variety of factors, but aggressive driving behaviour is one of the major causes [6]. Investigating ways to make drivers aware of their dangerous habits and to teach them not to drive aggressively, is imperative to reducing the occurrence of road accidents.

1.2 Proposed solutions

In the last decade, various entities have been developing solutions to monitor a vehicle and its driver's behaviour [7–9]. However, these solutions are expensive and intended for fleet management, and there is little incentive for individuals to buy them. However, the increasingly ubiquitous presence of smartphones – with their variety of sensors – presents the possibility to easily implement vehicle monitoring systems on a large scale.

Driver behaviour monitoring is an attractive application for smartphones in vehicles. Drivers can be monitored to make them aware of their potentially dangerous driving behaviour. Anonymous participatory sensing could also enable identifying areas where accidents are more likely to occur [10]. The authorities could also be notified to investigate extreme cases of aggressive driving.

Smartphones' connectivity also allows for the implementation of other vehicle monitoring features, such as traffic monitoring, traffic re-routing and accident reporting. Accident detection is possible using only the sensors in a modern smartphone, as shown by White et al. [11]. The swift automatic reporting of road accidents to authorities can prevent fatalities by minimising the response time of emergency services. Additionally, using a machine-to-machine (M2M) communication platform would allow the redirection of other drivers away from an accident. Notifying drivers that they are approaching an accident scene could also increase their alertness and warn them to slow down, thereby preventing further accidents.

1.2.1 Insurance incentive

Medical aid and vehicle insurance companies can benefit by obtaining driving behaviour data of their clients. It would enable an insurance company to offer competitive premiums and better services such as swift emergency roadside assistance. For this purpose, companies are starting to develop smartphone-based driver behaviour monitoring and accident detection systems, such as Discovery Insure, who already released an Android driver behaviour monitoring application to the public. The advantage of their application is its ability to operate unobtrusively and incessantly without the need of user interaction. It is also an inexpensive and widely deployable system, that could help reduce the occurrence of road accidents, by making drivers aware of their bad driving habits.

1.3 Thesis statement and hypotheses

Aggressive and normal (non-aggressive) lateral driving manoeuvres can be successfully detected and classified by a smartphone, using data from its embedded accelerometer, gyroscope and GPS. Supervised learning classification algorithms can be utilised for aggressive driving manoeuvre recognition and outperform existing solutions.

Hypothesis 1:

Smartphone sensors can be sampled and processed fast enough to detect and classify driving manoeuvres.

Hypothesis 2:

Rotation rate measurements from a gyroscope are sufficient to detect the start and end of a lateral driving manoeuvre, and to classify the severity of a road bend when driving through it.

Hypothesis 3.

Acceleration and rotation rate measurements can be successfully filtered to remove unwanted noise and offsets.

Hypothesis 4.

Aggressive lateral driving manoeuvres can be identified from acceleration, speed and bend severity information.

Hypothesis 5.

Supervised learning classification algorithms can be successfully applied to aggressive driving recognition, and they can perform better than the existing dynamic time warping approach as found in literature.

1.4 Research objectives

Objective 1:

Develop a system that can detect lateral driving manoeuvres using data solely from a smartphone's embedded accelerometer, gyroscope and GPS. The system must record the relevant sensor and GPS data to be used for classification of a manoeuvre. The system must be able to operate as an application on a smartphone in real-time.

Objective 2:

Develop a system that can evaluate detected lateral driving manoeuvres against a determined model of aggressive driving. Investigate the suitability of various supervised learning classification algorithms for classifying aggressive manoeuvres.

Objective 3:

Implement a dynamic time warping approach to classifying driving manoeuvres, as described in existing literature, and compare its performance to that of the best suited supervised learning classifier.

1.5 Scope of work

This thesis focuses on the use of smartphones to monitor a person's driving behaviour. The framework for a smartphone-based system that can detect and classify various driving manoeuvres is developed. As a proof of concept, a system is developed that specifically detects lateral driving manoeuvres and that classifies them as aggressive or not, using a supervised learning classification algorithm. The aggressive driving model is based on the angle of a turn, the lateral force exerted on the vehicle and its speed through the turn. Existing solutions found in research literature are investigated and presented. The best existing solution is also implemented, and then compared to the most effective supervised learning classification algorithm that was tested.

1.6 Thesis structure

Chapter 2 presents a comprehensive literature survey of the current state of smartphone-based vehicle monitoring systems. There are four main categories: Road condition monitoring, vehicle telematics, driver behaviour monitoring and collaborative driving. Particular focus is placed on a few examples of smartphone-based driver behaviour monitoring systems.

Chapter 3 gives an overview of the aggressive driving model that is investigated and the design of the experimental system to test it.

Chapter 4 describes the system design in further detail. The data acquisition system and collection process is discussed, as well as two aggressive driving recognition approaches.

Chapter 5 provides details of the tests and results of the different approaches and algorithms described in Chapter 4. The performance of each approach is tested using the same sampled data set and a quantitative comparison is made between them.

Chapter 6 presents a summary of the work and the conclusions drawn from it. The research hypotheses set out in Chapter 1 are validated with the test results. The contributions of the work and possible future work are discussed.

Chapter 2

Literature Survey

In this chapter a brief overview is first given of the current literature on smartphone sensing in vehicles. A number of available papers specifically describing vehicle monitoring systems that are entirely implemented on a smartphone are then analysed, reviewed and compared. Lastly, the challenges facing the progress and adoption of vehicle monitoring systems are discussed.

2.1 Smartphone sensing in vehicles

Machine-to-machine (M2M) communications describes a system where multiple electronic devices communicate autonomously to enable the sharing of information [12]. Among the millions of M2M devices that will be deployed world-wide in the coming years, smartphones will be the most mobile, versatile and powerful devices that can be used as sensors and M2M gateways [13]. Therefore, much research has been done on smartphone sensing applications in recent years [13–21].

In this section, a brief overview is given of the current literature on smartphone sensing in vehicles. The literature is organised in four categories: Road condition monitoring, vehicle telematics, driver behaviour monitoring and collaborative driving. The features and goals of the recent projects are listed in Table 2.1 to provide context of the systems discussed in the rest of this thesis.

2.1.1 Collaborative driving

M2M platforms have been developed where smartphones are used as sensor gateways in vehicles to support traffic management applications such as detecting congestion and rerouting traffic [22–24]. The platform described by Ali et al. serves the same purpose, but information is manually entered by users on their smartphones [25].

SignalGuru [14,27] is a service that uses smartphones to opportunistically detect traffic signals with their cameras and collaboratively share and learn traffic signal schedules. This enables Green Light Optimal Speed Advisory (GLOSA) applications which provide drivers with the schedule of forward traffic signals, allowing them to avoid coming to a complete halt and thereby lowering their fuel consumption. Jam Eyes [26] is an application that uses a smartphone's camera and wifi to detect vehicles around it in a traffic jam, in order to collaboratively calculate the length of a traffic queue.

Table 2.1: Literature relevant to smartphone sensing in vehicles.

Ref.	Year	Technology	Category	Goal
[13]	2007	mobile phone	collaborative driving	mobile phones as sensor gateways
[22]	2012	smartphone, external sensors	vehicle telematics	engine parameters collection from external sensors
[23]	2013	smartphone, external sensors, vehicle ECU	vehicle telematics	opportunistic transfer of external sensors and CAN bus data
[24]	2011	smartphone	collaborative driving	intelligent driver guidance tool
[25]	2012	smartphone	collaborative driving	road incident and traffic crowd-sourcing
[26]	2012	smartphone	collaborative driving	traffic queue length detection
[14] [27]	2011	smartphone	collaborative driving	traffic signal detection and learning
[28]	2009	smartphone	driver behaviour monitoring	lane departure warning system
[10]	2011	smartphone	driver behaviour monitoring	aggressive driving detection
[29]	2012	smartphone	driver behaviour monitoring	driving style characterization
[30]	2010	smartphone	driver behaviour monitoring	drunk driving detection
[31]	2012	smartphone	driver behaviour monitoring, road condition monitoring	advanced driver-assistance system
[32]	2008	smartphone	collaborative driving, driver behaviour monitoring, road condition monitoring	road and traffic condition monitoring
[33]	2012	smartphone	collaborative driving, road condition monitoring	pothole detection and notification
[34]	2011	smartphone	collaborative driving, road condition monitoring	pothole detection and notification
[35]	2013	smartphone	vehicle telematics	hybrid electric vehicle diagnostics
[36]	2011	smartphone, vehicle ECU	vehicle telematics	accident detection and notification
[11] [37]	2010	smartphone	vehicle telematics	accident detection and notification
[38]	2011	smartphone	driver behaviour monitoring	eco-driving assistant
[39]	2012	smartphone	driver behaviour monitoring	eco-driving assistant

2.1.2 Vehicle telematics

Zaldivar et al. [36] proposed using smartphones as an alternative on-board unit (OBU) in vehicles to access information in the vehicle's electronic control unit (ECU) wirelessly. An ECU is typically accessed through an industry-standard on-board diagnostic connector, known as OBD-II. By connecting an OBD-II-to-bluetooth adapter to the vehicle's controller area network (CAN) bus, a smartphone can gain access to the bus via bluetooth. Automatic accident detection is accomplished by using data obtained from the CAN bus together with the smartphone's GPS and accelerometer. Similarly, Yang et al. [35] developed a smartphone-based diagnostic system for hybrid electric vehicles that also accesses a vehicle's CAN bus through OBD-II.

The WreckWatch [11] accident detection system developed by White et al. differs from the one developed by Zaldivar et al. in that it detects accidents using only the accelerometer values from a smartphone, and not the values from a vehicle's electronic control unit (ECU).

2.1.3 Driver behaviour monitoring

Johnson and Trivedi [10] developed one of the first complete driver behaviour monitoring systems on a smartphone. Their system can detect and classify a number of aggressive and non-aggressive driving manoeuvres when placed in a vehicle, by only using the internal accelerometer, gyroscope, magnetometer and GPS of a smartphone. Other driver behaviour monitoring systems similar to the work in [10] has also been developed. Eren et al. [29] used a similar approach based on the same algorithms, but expanded their system by adding a driving style classification feature. Dai et al. [30] developed a system that specifically detects drunk driving. Fazeen et al. [31] developed a driver behaviour monitoring system that advises a driver on dangerous vehicle manoeuvres.

SmartLDWS [28] is a vision-based lane departure warning system developed for a smartphone. It employs a novel lane detection algorithm that provides satisfactory performance with the poor cameras typically found on older smartphones — while also being scalable to available computing power.

Eco-driving applications for smartphones aim to increase a driver's fuel efficiency by evaluating their driving and providing constructive feedback. Artemisa [38] is one such application which uses a smartphone's accelerometer to model a person's driving style and provides eco-driving tips to correct bad driving habits.

2.1.4 Road condition monitoring

Eriksson et al. [40] developed one of the first road condition monitoring systems that detects and maps road anomalies, such as potholes, using an accelerometer and GPS. Mednis et al. [34] and Ghose et al. [33] later both developed road condition monitoring applications for a smartphone which sends sensor data to a remote server and alerts drivers of potholes in the road.

Mohan et al. [32] developed a road and traffic monitoring system, named Nericell, which also employs smartphone sensors to detect certain conditions. In addition to the driver behaviour monitoring feature, Fazeen et al. [31] also added a road condition characterisation and mapping feature to their system that uses a smartphone's GPS and accelerometer.

Table 2.2: Summary of techniques and sensors used by smartphone-based vehicle monitoring systems.

Reference	**Detection technique**	**Sensors used**
Mohan [32]	pattern matching, orientation calibration	accelerometer, microphone, GPS
Dai [30]	pattern matching, orientation calibration	accelerometer, gyroscope
Johnson [10]	endpoint detection, DTW	accelerometer, gyroscope, magnetometer, GPS
Eren [29]	endpoint detection, DTW, Bayesian classifier	accelerometer, gyroscope, magnetometer
Fazeen [31]	pattern recognition	accelerometer, GPS
White [11]	pattern matching	accelerometer, microphone, GPS

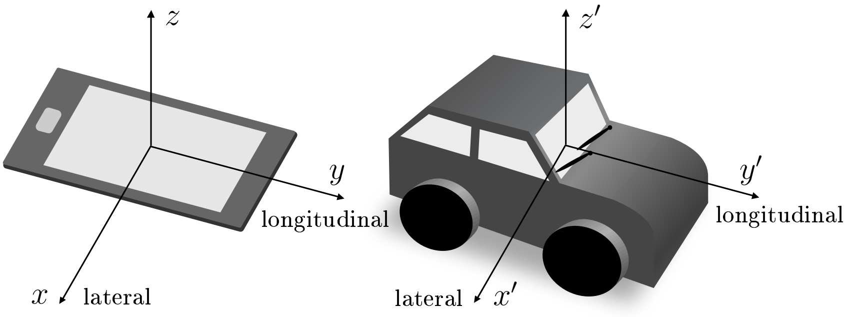


Figure 2.1: Smartphone and vehicle coordinate system used in analysis.

2.2 Entirely smartphone-based vehicle monitoring systems

The vehicle monitoring systems catalogued in Section 2 that solely rely on the embedded sensors of a smartphone are further analysed in this section. Only systems that use a smartphone's embedded sensors in an unobtrusive and energy-efficient manner are considered. For instance, systems using image processing techniques on a smartphone's camera are not considered, as it is processing and power intensive, as well as requiring the smartphone to be mounted on the dashboard. The relevant published papers that are analysed are listed in Table 2.2.

In the rest of the thesis, readings from an accelerometer's three axes (x, y, z) are denoted as a_x, a_y and a_z. Readings from a gyroscope's three axes are denoted as ω_x, ω_y and ω_z. A vehicle's axes are denoted as x', y' and z'. Accelerometer readings are expressed proportional to the acceleration from gravity, g (9.8 m/s^2), and gyroscope readings in terms of rotation rate (rad/s). Acceleration vectors are denoted as $\mathbf{a}_x$, $\mathbf{a}_{x'}$, etc; representing signals sampled in time. As shown in Figure 2.1, the axes of a smartphone are defined as x pointing towards the right and y to the top from the phone's front, while z points out orthogonal to the screen. A vehicle's axes are defined as x' pointing towards the right and y' to the front of the vehicle, while z' points up towards the roof.

2.2.1 Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones — Mohan et al.

Nericell [32] is a smartphone-based system designed to detect certain conditions pertaining to vehicles such as braking, bumps in the road, hooting and stop-and-go traffic. It uses a smartphone's accelerometer, microphone, GSM communications and GPS for this purpose. Nericell aggregates sensed data from multiple participating smartphones on a centralised server. Mohan et al. envisages the system being used to annotate existing traffic maps with information such as the condition of road surfaces and the level of chaotic traffic.

Nericell strives to use the sensors in a power-efficient manner. Only the accelerometer is sampled continuously with the GSM radio kept active, which is needed for communication anyway. The system relies on input from these devices to turn on the microphone or GPS only when they are needed in order to conserve energy. Data is filtered and processed locally on each smartphone before sending it to a server for aggregation.

2.2.1.1 Virtual reorientation procedure

The accelerometer is the key sensor of the system, as it is used for braking, pothole and bump detection. However, the orientation of its axes in relation to the vehicle in which it is, must be known. Therefore, an algorithm was developed to virtually reorientate the smartphone's accelerometer to the vehicle's frame of reference.

An accelerometer measures the acceleration associated with the weight experienced by any mass — therefore, if the accelerometer is aligned correctly, it will measure $a_z = -1g$. The framework used for the reorientation of the accelerometer is based on Euler angles, which simplifies the calculations considerably. The orientation of the accelerometer can be described as a pre-rotation about z', a tilt about y' and a post-rotation about z', denoted as θ_{pre}, θ_{tilt} and θ_{post} respectively. Only the tilt operation changes the angle of z with respect to z', and since $|a_{z'}| = 1$ when assuming the vehicle is on flat ground,

$$\theta_{\text{tilt}} = \cos^{-1}\left(\frac{a_z}{a_{z'}}\right) = \cos^{-1}(a_z) \tag{2.2.1}$$

Also, since $|a_{z'}| = 1$, pre-rotation followed by tilt would result in non-zero a_x and a_y. Therefore

$$\theta_{\text{pre}} = \tan^{-1}\left(\frac{a_y}{a_x}\right) \tag{2.2.2}$$

To estimate θ_{tilt} and θ_{pre} using (2.2.1) and (2.2.2), periods when the vehicle is stationary or in steady motion have to be identified. However, a simpler method that proved to work well was to use the median values of a_x, a_y and a_z over a 10 second window. As long as no high-speed sharp turns are performed during the window, the values are notably stable, even on a bumpy road. Lastly, the post-rotation about z' has no influence on the forces experienced due to gravity, therefore another force is needed in order to estimate the angle of rotation.

The acceleration and deceleration of a vehicle in a straight line provides a force in the x' direction. Deceleration (braking) tends to induce a stronger force than acceleration and is therefore used for the estimation procedure. The GPS trace is used to monitor the vehicle for a sharp deceleration event in a more or less straight line. As explained in the paper [32], a maximisation procedure yields an equation for θ_{post} dependent on θ_{tilt} and

Table 2.3: Cross-correlation between two well-oriented accelerometers (f, g), a disoriented accelerometer (h) and a virtually reoriented (h') accelerometer [32].

$f \star g$	$\theta_{\text{pre}}/\theta_{\text{tilt}}/\theta_{\text{post}}$	$h \star f$	$h' \star f$	$h \star g$	$h' \star g$
0.90	7°/38°/106°	0.30	0.88	0.20	0.91
0.75	174°/34°/-107°	0.43	0.72	0.54	0.87
0.94	174°/34°/-107°	0.59	0.84	0.67	0.90
0.74	4°/42°/12°	0.65	0.72	0.63	0.68
0.76	3°/44°/-1°	0.62	0.71	0.69	0.79
0.78	-80°/42°/121°	0.65	0.73	0.64	0.73

θ_{pre}. Therefore, to estimate θ_{post}, θ_{tilt} and θ_{pre} must first be estimated using (2.2.1) and (2.2.2). The GPS trace is then evaluated for a braking event and during the transient surge the mean of a_x, a_y and a_z are recorded. This is done to account for the time delay in the speed estimate received from the GPS.

The effectiveness of the virtual reorientation procedure was validated through testing. Values from a separate, well-oriented accelerometer, was recorded simultaneously with accelerometer values from a smartphone running Nericell, during numerous drives. The cross-correlation between the data was calculated to quantify the results, which are shown in Table 2.3. The cross-correlation is far from perfect, but this is mostly due to sensor noise and the fact that no two accelerometers will give the exact same values. In general, the cross-correlation between a well-oriented accelerometer and a disoriented smartphone accelerometer, improves substantially when the latter has been virtually reoriented.

2.2.1.2 Driving event detection

After a smartphone's accelerometer has been successfully reoriented, it can be used to detect certain events from which road and traffic conditions can be deduced. The first problem pertains to detecting braking events. A high occurrence of braking on a stretch of road could indicate poor road conditions or heavy traffic. Braking causes a surge in the y' direction of the accelerometer due to the force acting on it. Detecting a braking event is fairly easy, since the surge typically spans more than a second. The mean of $a_{y'}$ is calculated over a sliding window of N seconds, and if it exceeds the threshold λ, a braking event is assumed. The GPS could also be used to detect braking, as with the orientation procedure, but it uses more power and it is susceptible to the GPS localisation error.

Detecting a bump or pothole in a road is the second problem for which the accelerometer is used. This is more challenging to accomplish, since the accelerometer's signal can vary considerably when driving over a bump, depending on the size of the bump and the vehicle's speed. The duration of such events are typically also very short (in the order of milliseconds). Nericell uses two methods for bump detection — one for low speeds and one for high speeds. When a vehicle's wheel hits the bottom of a pothole, a sharp force is induced which causes a distinct spike in the curve of $a_{z'}$. Therefore, a surge in $a_{z'}$ greater than a threshold is recognised as a bump. However, at low speeds (<25 km/h) the spike is not distinct enough from noise. The sustained crossing of a threshold for at least 20 ms is indicative of a bump at low speeds, while at high speeds slight unevenness also causes sustained peaks and dips. Monitoring for a peak in $a_{z'}$ is therefore appropriate for high speeds (*z-peak*); while evaluating for a short sustained rise or dip in $a_{z'}$ is appropriate for low speeds (*z-sus*). Lastly, the coarse estimate of speed obtained from GSM-localisation is sufficient to govern which method to apply.

Nericell also uses a smartphone's microphone for horn detection. Although audio recording and processing consumes a considerable amount of power, it is only triggered when frequent braking is detected. To protect user privacy, only anonymous information obtained after processing is sent to the server. The goal of horn detection is to conjecture chaotic traffic conditions in some places, such as at unregulated intersections. The discrete Fourier transform is performed on short audio samples to be able to detect significant energy spikes in the frequency domain. The detection algorithm is based on empirical observations: if at least two spikes (harmonics) are detected, with one in the 2.5 to 4 kHz range, the audio sample is classified as containing the sound of a horn.

Braking detection was tested during a 35 km drive with varied traffic conditions. The detection performance is measured in terms of false positives (FP) and false negatives (FN). For the ground truth, 45 braking events were identified from the GPS trace with a threshold of $\lambda = 0.1g$ that must be exceeded for at least 4 seconds. A 4 second window and threshold values of $\lambda = 0.11g$ and $\lambda = 0.12g$ were used for the accelerometer braking detection algorithm. For $\lambda = 0.11g$, a FN and FP rate of 4.4% and 31.1% was obtained respectively. While for $\lambda = 0.12g$, a FN and FP rate of 11.1% and 17.7% was obtained respectively.

Bump detection was tested by manually annotating bumps on a route to use as the ground truth. To lessen subjectivity, a route was repeated a few times and a consensus was reached between two or three people. On a 30 km route of mixed road conditions, a total of 101 bumps or potholes were noted. At low speed, FN and FP rates of 37% and 14%, respectively, were obtained by *z-sus*. At high speed, FN and FP rates of 41% and 8%, respectively, were obtained by *z-peak*.

The horn detector was tested simultaneously on four phones at a chaotic intersection. The ground truth was establish by listening to an audio recording and manually noting the time when a horn is heard. With a high enough threshold to avoid false positives, FN rates of 0% to 50% was obtained between the different phones while placed inside and outside of an enclosed vehicle.

The virtual reorientation procedure is based on the assumption that the vehicle is on flat ground when it is performed. If the vehicle is on a steep incline when the reorientation procedure is started, the accuracy of the system could be considerable reduced.

2.2.2 Mobile Phone-Based Drunk Driving Detection — Dai et al.

Dai et al. [30] developed a system which can detect drunk driving by solely using a smartphone's accelerometer. As far as is known, they were the first to develop a system that uses smartphone sensors for driver behaviour recognition. Their motivation for designing such a system is the fact that most of the time drunk driving goes unnoticed by the authorities, which puts many people's safety at risk.

They summarised drunk driving related behaviours from a study done by the United States National Highway Traffic Safety Administration (NHTSA). There are two categories of behavioural cues which correspond to a high probability of drunk driving. The first category is related to lane positioning problems such as drifting and swerving. The second category is related to speed control problems such as sudden acceleration or erratic braking. Both these categories of cues can be detected by using an accelerometer to map these cues into the lateral and longitudinal acceleration of a vehicle.

The system is designed with four software components: a monitoring daemon module, calibration module, pattern matching module and an alert module. The calibration module determines the orientation of the smartphone within a moving vehicle. This enables the system to function irrespective of where and how the smartphone is placed in a vehicle. The monitoring daemon continuously examines accelerometer samples in order to start the calibration module when vehicle movement is detected. The initial acceleration of a vehicle induces a continuous longitudinal force on the accelerometer in either the forward or backward direction. This acceleration is denoted as vector $\mathbf{a}_l$. It was determined empirically that $\mathbf{a}_l$ must exceed $0.265g$ for several seconds before the calibration module is started.

When the calibration module is started, the orientation sensor (gyroscope) of the smartphone is first used to obtain its yaw, pitch and roll — denoted as θ_x, θ_y and θ_z, respectively. The horizontal acceleration components of the smartphone's x and y-axis, denoted as $\mathbf{a}_{xh}$ and $\mathbf{a}_{yh}$, are then obtained from

$$\begin{aligned} \mathbf{a}_{xh} &= \mathbf{a}_x \cos(\theta_z) \\ \mathbf{a}_{yh} &= \mathbf{a}_y \cos(\theta_y) \end{aligned} \tag{2.2.3}$$

Next, the magnitude of $\mathbf{a}_l$ is obtained by

$$|\mathbf{a}_l| = \sqrt{|\mathbf{a}_{xh}|^2 + |\mathbf{a}_{yh}|^2} \tag{2.2.4}$$

The angle between vector $\mathbf{a}_{xh}$ and $\mathbf{a}_l$ is denoted as ϕ, while the angle between vector $\mathbf{a}_{yh}$ and $\mathbf{a}_l$ is denoted as ψ. These angles are obtained from

$$\begin{aligned} \phi &= \arccos(\mathbf{a}_{xh}/|\mathbf{a}_l|) \\ \psi &= \arccos(\mathbf{a}_{yh}/|\mathbf{a}_l|) \end{aligned} \tag{2.2.5}$$

Lastly, the lateral acceleration vector $\mathbf{a}_{x'}$ and longitudinal acceleration vector $\mathbf{a}_{y'}$ of the vehicle is obtained from

$$\begin{aligned} \mathbf{a}_{x'} &= \mathbf{a}_{xh} \sin\phi + \mathbf{a}_{yh} \sin\psi \\ \mathbf{a}_{y'} &= \mathbf{a}_{xh} \cos\phi + \mathbf{a}_{yh} \cos\psi \end{aligned} \tag{2.2.6}$$

The pattern matching module is only activated once the calibration procedure is done. It evaluates the difference between the maximum and minimum value of lateral acceleration ($\mathbf{a}_{x'}$) within a pattern checking time window. If the difference exceeds a set threshold, the module reports that an abnormal curvilinear movement has occurred. The module also checks whether the longitudinal acceleration ($\mathbf{a}_{y'}$) exceeds fixed positive or negative thresholds at any given time, indicating a speed control problem. Multiple rounds of pattern matching are performed on both the detection algorithms, which is necessary in order to detect drunk driving with a high degree of certainty.

Tests were conducted to obtain a total of 72 sets of data for drunk driving related behaviours and 22 sets of data for regular driving. The time window for lateral acceleration was set to 5 seconds, as most acceleration patterns happen within this period. The threshold values were set to achieve a low number of FN and a sensible FP probability. The approach achieved a FN rate of 0% for both lateral and longitudinal driving events, and a FP rate of 0.5% and 2.4% for lateral and longitudinal driving respectively. Testing also showed that the system has tolerable power consumption. Initially, starting from a fully charged battery, the smartphone's battery level was at 78% after 7 hours of operation — compared to 92% when the smartphone and system were idle for 7 hours.

A limitation of the system is that it cannot determine the speed of the vehicle, which would improve the ability of the system to identify dangerous driving patterns. The smartphone's GPS can provide the speed of the vehicle, but at the expense of increased power consumption. The system could also be improved by using the GPS to match the movement of the vehicle to road directions.

2.2.3 Driving Style Recognition Using a Smartphone as a Sensor Platform — Johnson and Trivedi

The driver behaviour monitoring system developed by Johnson and Trivedi [10], named MIROAD, solely relies on the internal accelerometer, gyroscope, magnetometer and GPS of a smartphone. They were the first to develop a more complex pattern recognition approach. They also argue that anonymous participatory sensing would allow the number of aggressive drivers in an area to be established. This gives foresight into where accidents may possibly occur.

A smartphone running MIROAD can detect and classify a number of aggressive and non-aggressive driving manoeuvres. However, unlike in Section 2.2.1 and 2.2.2, the smartphone has to be mounted on the dashboard of the vehicle. The system assumes the smartphone's axes are orientated with x, y and z in the direction of the sky (z'), left side ($-x'$) and back ($-y'$) of the vehicle respectively. Output from the accelerometer, gyroscope and magnetometer of the smartphone is used for manoeuvre recognition. With the magnetometer, corrections are made with respect to magnetic north. The Euler rotation, also used in Section 2.2.1, can therefore be determined more accurately from a reference attitude.

The manoeuvres associated with aggressive driving are hard left and right turns, swerving, and sudden braking and acceleration patterns. For the detection of longitudinal movements, the rotation rate ω'_x and the a'_y acceleration are used. For lateral movements, the a'_x acceleration and rotation rate ω'_z, as well as the Euler angle about y' is used. The accelerometer and gyroscope are continuously sampled at a rate of 25 Hz. In order to detect manoeuvres, the start and end of driving events are determined by using the end-point detection algorithm. For lateral manoeuvres, a simple moving average (SMA) of ω'_z is continuously calculated for a short window of k samples. From the current sample i, we have

$$\text{SMA}(i) = \frac{\omega'_z(i)^2 + \omega'_z(i-1)^2 + ... + \omega'_z(i-k-1)^2}{k} \tag{2.2.7}$$

The beginning of an event is detected if the SMA goes above a set upper threshold. The succeeding gyroscope values are concatenated until the SMA falls below a bottom threshold, signifying the end of the event. The event is dismissed if it exceeds 375 samples, or 15 seconds.

When a valid driving event has been detected, the signals recorded during the event are compared to a set of template events using the Dynamic Time Warping (DTW) algorithm [41]. DTW finds an optimal alignment between two time-dependant sequences with different lengths. Consider a matrix of the Euclidean distance between each point of two sequences, as seen in Table 2.4. Both sequences start at the bottom left corner. An optimal warping path constitutes the minimum sum of distances, or cost, while adhering

Table 2.4: DTW cost matrix showing the optimal warping path.

Template	Minimum-distance									***0***
0	0	-1	-2	-3	-4	-4	-2	-1	-1	***0***
1	1	0	-1	-2	-3	-3	-1	***0***	***0***	1
2	2	1	0	-1	-2	-2	***0***	1	1	2
4	4	3	2	1	0	***0***	2	3	3	4
3	3	2	1	***0***	***-1***	-1	1	2	2	3
3	3	2	***1***	0	-1	-1	1	2	2	3
1	1	***0***	-1	-2	-3	-3	-1	0	0	1
0	***0***	-1	-2	-3	-4	-4	-2	-1	-1	0
Measured	**0**	**1**	**2**	**3**	**4**	**4**	**2**	**1**	**1**	**0**

to monotonicity, boundary and step size conditions. The template event with the lowest warping path cost is the closest match to the detected event.

Video footage is continuously recorded by the rear camera of the smartphone facing towards the front of the vehicle, but no image processing is performed. MIROAD is able to play back video and sensor data synchronously to provide a recreation of an incident. To limit memory usage, all data is recorded in five minute intervals and if it is not flagged for any detected events, it is deleted. The system audibly alerts drivers of aggressive driving events through a software speech synthesizer. Alerts are also sent with the vehicle's location to external systems via the GSM internet connection.

MIROAD was tested in three different vehicles, with three different drivers. They collectively accumulated 201 driving events on highways, urban and rural roads, of which about 50 were considered possibly aggressive. While the detection rates vary for different events, in total, 97% of all the aggressive driving events were correctly identified by die DTW algorithm.

A limitation of the system is that the smartphone has to be kept in a fixed position within the vehicle. The system also consumes more power than the simpler systems with a minimal increase in detection accuracy.

2.2.4 Estimating Driver Behavior by a Smartphone — Eren et al.

The system developed by Eren et al. [29] characterises a person's driving as either safe or risky. Sudden manoeuvres, turns, lane departures, braking and acceleration are seen as risky events. These events are detected by only using the accelerometer, gyroscope and magnetometer of a smartphone.

The system detects driving events similarly to the technique described in Section 2.2.3. Moving average filters are applied to the raw sensor data to eliminate noise. Likewise, the endpoint detection algorithm is used to identify events, and the dynamic time warping algorithm is also used to compare input data to template events. However, another layer is added to the system that Johnson and Trivedi's system lacks — labelling of the driver behaviour, as can be seen in Figure 2.2.

A Bayesian classifier is used to label a driver's behaviour as either safe or risky according to a calculated probability. The existence of two classes, r_1 and r_2, related to safe

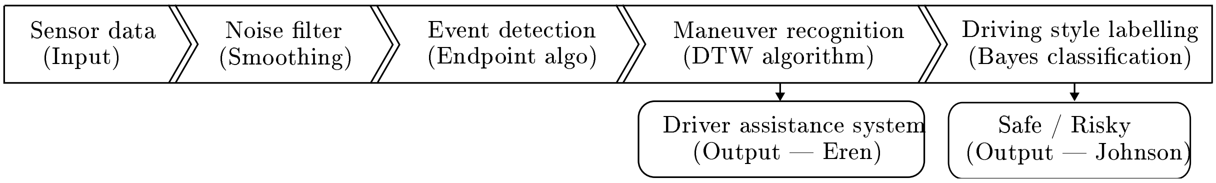


Figure 2.2: Block diagram illustrating the extra layer of Eren et al.'s system over Johnson's and Trivedi's system.

and risky driving is assumed. The calculation is based on the number of occurrences of different driving events (s) over time. The mathematical expression for determining the probabilities is given by

$$P(r_1|s) = \frac{P(r_1)P(s|r_1)}{P(s)} = \frac{P(r_1)P(s|r_1)}{P(r_1)P(s|r_1) + ... + P(r_n)P(s|r_n)} \tag{2.2.8}$$

The classification is made by comparing the calculated probabilities

$$\begin{aligned} P(r_1|s) > P(r_2|s) &\Rightarrow \text{Safe} \\ P(r_1|s) \leq P(r_2|s) &\Rightarrow \text{Risky} \end{aligned} \tag{2.2.9}$$

The driving patterns of a selected group of drivers were analysed. The group consisted of five experienced, five novice and five randomly chosen drivers. Two tests were done with each driver in different weather conditions in order to evaluate the reliability of the system. Two other experienced drivers were also selected to sit in the passenger sides of the vehicles while the tests were performed. They were required to fill in a short survey after each test. The Bayesian classifier correctly identified the driving style as safe or risky, as well as driving event types, for 14 out of the 15 drivers. However, the accuracy of the classifier is based on completed surveys by test participants — therefore the results could have been influenced by subjectivity. Having to place the smartphone in a fixed position within the vehicle is also a limitation of the system, as is the power consumption, similar to the system of Section 2.2.3.

2.2.5 Safe Driving Using Mobile Phones — Fazeen et al.

Fazeen et al. [31] envisages implementing an advanced driver-assistance system (ADAS) on a smartphone. Such a system advises a driver on dangerous situations that emerge from vehicle manoeuvres and environmental factors. The system uses the accelerometer and GPS of a smartphone to enable typical features found in ADAS-equipped vehicles. The aim of the system is to recognise and classify driving behaviour and to map road surface conditions.

Data identified as part of a driving event or road anomaly is stored on the phone and the user has full control over it. Any data that is sent to a server for mapping and machine learning techniques is kept anonymous. Drivers are audibly alerted by the system of dangerous situations.

Road anomalies can be detected because of the vibrations experienced by a vehicle when driving on a rough road. When a vehicle drives over a bump, it ascends onto it, resulting in a sharp rise in the z-axis value of the accelerometer. An increase in the x-axis value is also observed, depending on the shape of the bump, because of the longitudinal force exerted on the vehicle's wheels. The difference between successive accelerometer readings is continuously evaluated. A bump in the road is presumed if the difference exceeds a dynamic threshold, which is dependent on the speed of the vehicle. The height of a bump can be roughly calculated by using dynamics equations. The accuracy of the approximation could be improved if the dynamics of the vehicle's suspension is known.

When a road anomaly is detected, the current GPS coordinates are saved with a corresponding value indicating the condition of the road. The system classifies a section of a road as either smooth, rough, uneven, or as containing a bump or pothole. The data can then be used to map the condition of entire stretches of road.

During testing, an overall accuracy of 85.6% was achieved by the road condition classification system. It was determined empirically that safe acceleration and deceleration never exceeds $\pm 0.3g$, while sudden manoeuvres approaches, but does not exceed $\pm 0.5g$. A gradual lane change, in comparison, exerts an average lateral acceleration of only $\pm 0.1g$. It is therefore possible to differentiate between safe and unsafe driving manoeuvres. It was also found that recognising gear shifts is possible, which would enable the system to advise a driver when to shift gears in order to achieve efficient fuel usage.

The system relies on the GPS for calculation of the vehicle's velocity, consuming power additional to the sampling of the accelerometer. To overcome this limitation, the vehicle's velocity could rather be calculated by integrating the sampled acceleration curve between each gear shift, thereby easing battery usage.

The accuracy of a Nexus One smartphone's accelerometer was tested by comparing it to calculated data from dynamics equations. The system requires the smartphone to be orientated with the top facing to the front of the vehicle and the screen facing upwards. Tests were conducted with the smartphone placed in different locations in the vehicle. The results showed that placement in the center of the vehicle is the best location for monitoring driving behaviour. The required fixed orientation of the smartphone in the vehicle is also a limitation of the system.

2.2.6 WreckWatch: Automatic Traffic Accident Detection and Notification with Smartphones — White et al.

The WreckWatch [11] system detects accidents using only the sensors from a smartphone, as opposed to similar systems which use values from a vehicle's electronic control unit (ECU). White et al. argue that it is unrealistic to expect drivers to connect their smartphones to the ECU for every journey. Another concern is that older vehicles do not have ECUs, therefore, an accident detection system that is not dependent on an ECU is advantageous.

WreckWatch uses a soft real-time (close to real-time) approach sampling the accelerometer, microphone and GPS of a smartphone. An accident is detected by threshold filtering the sensor readings. Data recorded preceding and during an accident is sent via GSM to a centralised server. Important information about an accident can then be relayed to the relevant authorities from a stored database on the server.

False positives are more likely to occur with a system using only smartphone sensor data. Dropping the phone on the floor or making a sudden stop may be detected as an accident. Therefore, context information obtained from filters must be used to prevent false positives. Firstly, the determined acceleration is filtered by ignoring any values below $4g$. Secondly, a user is assumed to be in a vehicle if they are moving faster than 25 km/h. The smartphone's GPS is used to determine the speed of the user. Accelerometer information is only evaluated when the user is travelling faster than 25 km/h. This reduces power consumption and prevents false alerts from occurring if the phone is accidentally dropped while being outside a vehicle.

The WreckWatch system has been developed further to improve low-speed collision detection by adding acoustic data analysis. The microphone of the smartphone is used to listen for high-decibel sounds such as impact noise, car horns or airbag deployment (170 dB peak).

If an accident is detected, emergency responders are automatically notified by the system. Situational awareness is provided to the first responders to enable them to assess the severity of the accident. The GPS coordinates of the accident are immediately sent to the server with other accident characteristics. Thereafter, bystanders and uninjured victims can provide critical information through the WreckWatch application. For instance, pictures of the accident can be taken with the smartphone's camera and shared with the first responders.

A few different tests were performed to evaluate the possibility of false positives occurring. The empirical results demonstrate that dropping the smartphone in a moving vehicle is not likely to cause a false positive. The filter threshold of $4g$ on accelerometer readings are sufficient to prevent it. Furthermore, it was found that threshold filtering can not be used for acoustic detection of airbag deployment. Playing music at full volume or people shouting in the vehicle causes sound signal clipping at 145 dB on the smartphone. Therefore, the system relies on the acoustic signature of a detected event as a secondary indicator of an accident. As with the systems of Sections 2.2.3, 2.2.4 and 2.2.5, the required fixed orientation of the smartphone in the vehicle is also a limitation of the system.

2.2.7 Review and comparison of the summarised papers

In this section, the detection techniques, hardware, software and objectives of the different systems are compared. Although the sensors and detection techniques used by the different systems are similar, as can be seen in Table 2.2, their objectives differ slightly.

2.2.7.1 Driving manoeuvre recognition versus driving behaviour classification

The system of Dai et al. explicitly attempts to determine whether a driver is drunk. This is achieved by detecting and positively identifying a combination of dangerous driving manoeuvres associated with drunk driving. Johnson and Trivedi's system can detect and identify a number of different driving manoeuvres, but does not draw any conclusions from them. Their intent is to use the system to support a holistic driver assistance system (DAS) by providing it with additional information. The system of Eren et al. also detects driving manoeuvres, but incorporates another layer where a person's driving style is labelled as either safe or unsafe with a given probability. Fazeen et al. aims to implement an ADAS entirely on a smartphone. Their system records and analyses various driver behaviours and external road conditions.

It is important to note the difference between driving manoeuvre recognition and driving behaviour classification. One system could detect various manoeuvres, but not necessarily infer anything from them, whereas another system may be able to deduce and classify a driver's behaviour from detected driving manoeuvres. These different systems demonstrate the variety of driving behaviour classifications that can be made. A person's normal driving style can be classified as safe or risky, fuel-efficient or inefficient, skilled

Table 2.5: Summary of hardware and software used by smartphone-based vehicle monitoring systems.

Reference	Hardware	Software
Mohan [32]	HP iPAQ hw6965 PDA, HTC Typhoon smartphone, Sparkfun WiTilt accelerometer	Windows Mobile 5.0 and 2003
Dai [30]	HTC Dream (G1) smartphone	Android 1.6
Johnson [10]	iPhone 4	iOS 4
Eren [29]	iPhone 4	iOS 4
Fazeen [31]	HTC Google Nexus One smartphone	Android 2.1
White [11]	HTC Magic (Google ION) smartphone	Android 1.5

or unskilled — and recommendations can be given accordingly to improve their driving. On the other hand, a person's driving behaviour can sometimes differ from normal due to certain circumstances. A person could be driving under the influence of alcohol, drugs or other sensory impairments. In such situations a driver could be warned of their dangerous behaviour or the relevant authorities could even be notified of the driver's behaviour and location.

2.2.7.2 Accuracy versus simplicity

It is impractical to quantitatively compare the performance and power consumption of the different systems. All of the systems were implemented on different smartphones that have varied sensors and computing power. The test studies were performed in various countries with different road and traffic conditions. Their methods of establishing the ground truth for tests were not necessarily the same and could vary due to subjectivity.

Figure 2.3 shows a qualitative comparison of the accuracy versus simplicity of the different systems. A system that achieves high detection accuracy with a simple algorithm is considered superior. The assumption is that a simpler system uses less resources and therefore consumes less power, although it can not be explicitly proven here. The experimental and empirical test results of the systems as given in each paper were used to compare detection accuracy, although the testing procedures differed as mentioned. The perceived simplicity of each system is based on what each system is trying to detect, what sensors it uses and how its algorithms function.

WreckWatch of White et al. is empirically proven to be virtually 100% accurate and is the simplest system, because it only detects accidents and nothing else. The road condition monitoring feature of Mohan et al. is more accurate than that of Fazeen et al, and its implementation is simpler.

The drunk driving detection of Dai et al. is the most accurate, achieving a false negative rate of virtually 0%. Dai et al. implemented a simple yet effective pattern matching approach that requires very little computation. Essentially, only the difference in subsequent values on the relative longitudinal and latitudinal axes are calculated. If the difference exceeds a certain threshold, an aggressive driving manoeuvre is assumed. The algorithm used by Nericell of Mohan et al. works in a similar manner. Both systems consume less than 12% of the phone's battery during a normal use cycle.

In contrast, Johnson and Trivedi, as well as Eren et al., implemented a more complex pattern recognition approach derived from speech recognition techniques. Their systems perform well, achieving a true positive rate of 97% and 93%, respectively. Although

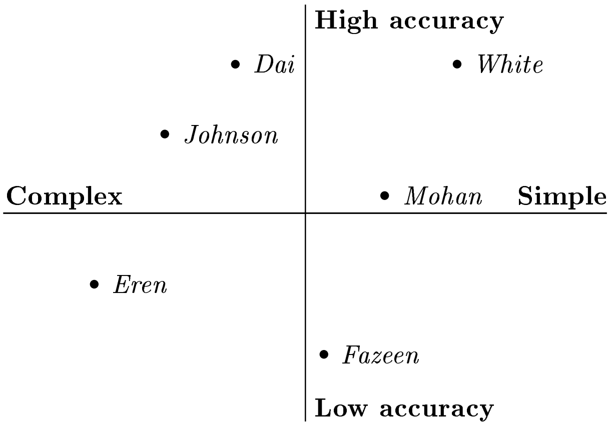


Figure 2.3: Qualitative comparison of accuracy versus simplicity of the different systems.

it can not be explicitly proven here, the simpler approaches are likely to consume less power while achieving similar performance to the more complex approaches. Arguably, Dai et al. accomplished the same functionality as Eren et al. with a simpler algorithm, as both systems can infer a certain aspect of a driver's behaviour from detected driving manoeuvres.

In Table 2.5 the hardware and software on which each system was developed can be seen. The systems of Mohan et al. and Dai et al. were developed on hardware and software that are now considered obsolete, yet their systems were simple and accurate. This suggests the performance of these systems do not necessarily benefit from the improvement of embedded sensors used in smartphones. The computing power and efficiency of modern smartphones has increased dramatically over the last decade, which provides headroom for more complex solutions. Therefore there is still merit in implementing a more complex approach as used by Johnson and Trivedi, if the accuracy could be improved to such an extent as to have very few false negatives (FN) or false positives (FP).

2.2.7.3 Contributions and best practices

In terms of contributions made, Dai et al. and Mohan et al. were the only authors to implement a procedure to calibrate the system to any arbitrary orientation of the smartphone. All of the other systems assume that the smartphone is placed in a fixed position within a vehicle. Automatic virtual reorientation of a smartphone's axes to a vehicle's axes is considered a best practice for any smartphone-based vehicle monitoring system.

2.3 Challenges facing vehicle monitoring systems

In this section, a number of remaining challenges facing the future progress of vehicle monitoring systems are discussed.

2.3.1 Algorithm performance

It is difficult to compare the accuracy and performance of the different implemented algorithms. The algorithms must each be tested using the same hardware and ground truth before the distinction between them can be fully appreciated. A study to better compare the algorithms would help to establish best practices on which to further build on.

2.3.2 Data aggregation

The collection and aggregation of data is a key component of intelligent transportation systems (ITS). The utility of vehicle monitoring systems could be substantially improved with effective participatory sensing. Sharing data with aggregation servers would allow providing additional services, such as warning users of accidents and heavy traffic. An unified framework for participatory vehicle sensing must be developed.

2.3.3 Wide-scale deployment

A key challenge facing vehicle monitoring systems is to accomplishing wide-scale deployment and use. Few individuals are willing to buy a dedicated system. It is for that reason that implementing such systems on smartphones is an attractive solution. There are no associated hardware costs, and smartphones are prevalent, even in developing countries. In fact, [42] expects ITS to be mobile-phone based in the developing world.

2.3.4 Automation

The operation of the smartphone application must be fully automatic and unobtrusive. It is necessary that during driving, user interaction with the smartphone is minimised — in order to keep the driver's focus on the road.

2.3.5 Safe feedback

Alerting drivers of notifications and warnings received from aggregation servers must not take their focus off the road. It must also not require any distracting interaction with the smartphone. Careful consideration must be given to the interface between a smartphone and driver.

2.3.6 Power consumption

The power consumption of such applications running on a smartphone must be minimised so that it will not affect a user's normal smartphone usage habits. The power consumption could be decreased by developing simplified algorithms that use less computing power.

2.3.7 Multi-platform

For a smartphone-based vehicle monitoring system to become widely used, it must work on a wide variety of smartphones. Another challenge is therefore to develop a multi- and cross-platform solution, and obtaining consistent performance between different smartphone hardware and software. The amount of noise in sensor readings can vary between different smartphones, for example. Typically, access to sensors are limited by the firmware and operating system of a smartphone. Therefore, techniques to standardise sensor readings on different smartphones through software must be investigated.

2.4 Conclusion

The use of smartphones for road safety applications was investigated in this Chapter. Firstly, a brief overview was given of smartphone sensing in vehicles. Although the concept may have existed for a while, the first paper demonstrating it was published in 2007. This field of research is therefore still relatively young and open for new ideas. Secondly, a few representative papers describing a variety of vehicle monitoring systems that are entirely smartphone-based were each separately summarised, and subsequently reviewed and compared. It is clearly demonstrated that it is possible to implement a complete vehicle monitoring system, and even a driver assistance system (DAS), on a smartphone, with acceptable performance. However, there are still a number of challenges facing the future progress of such systems. A key issue is accomplishing wide-scale deployment and use. In many cases these systems require many users before they become useful, and since there is little incentive for individuals to use them, adoption will be slow. We envisage that smartphone-based vehicle monitoring and driver assistance systems will be a crucial part of ITS in developing countries in future. If the majority of drivers on the road can be alerted of dangerous behaviour, situations and road conditions, there would consequently be fewer road accidents, and hence fewer road fatalities. This will also help lower traffic congestion by encouraging drivers to take alternative routes. The techniques used for smartphone-based vehicle monitoring could also be applied to other smartphone sensing applications in the future.

Chapter 3

Experimental Design

The existing literature reveals that machine learning techniques has thus far not been extensively used for smartphone-based driver behaviour monitoring. The most sophisticated system proposed thus far uses dynamic time warping, a well documented pattern recognition algorithm [43] [41] [44]. The goal of the investigation is to compare the performance of the DTW approach and supervised machine learning techniques in recognising aggressive driving.

The overall design of the experimental system is presented in this chapter. Firstly, a predictive aggressive driving model is presented. Secondly, an overview is given of the manoeuvre detection algorithm, as well as the two classification approaches used to recognise aggressive driving manoeuvres. Block diagrams of the system's design are provided.

3.1 Aggressive driving model

Aggressive driving is considered to be the conscious behaviour of a driver performing any manoeuvre in such a manner that increases the risk of a road accident. Such deliberate driving behaviour often involves exceeding the speed limit.

Speeding is one of the largest contributing factors to road accidents in South Africa, because of the combination of many un-roadworthy vehicles, poorly trained drivers and the substandard condition of many roads [45]. Further compounding the problem is the typical budgetary constraints of traffic law enforcement in many municipalities, making it difficult to effectively enforce speed limits. Evidently, because of this, making drivers aware of the danger of speeding has always been a top priority of road safety initiatives, such as the Arrive Alive campaign in South Africa [45]. Drivers are, unfortunately, not always aware that they are driving too fast for the shape of the road they are on. To make drivers aware of unsafe manoeuvres, we propose an aggressive driving recognition system that can run on a driver's own smartphone.

Given the discussion above, the aggressive driving model we use is based on the angle of a turn, the lateral force exerted on the vehicle and its speed through the turn. The gyroscope, accelerometer and GPS of a smartphone is used accordingly to obtain the required information.

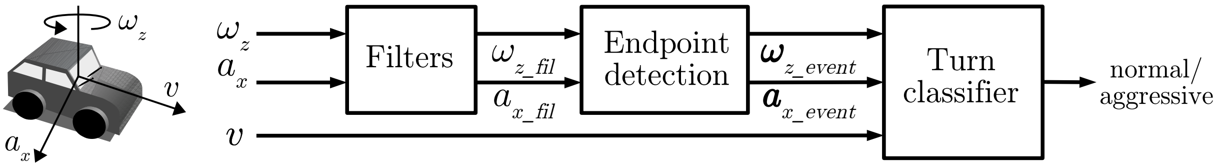


Figure 3.1: Overall system block diagram.

3.2 System overview

Figure 3.1 presents a block diagram of the overall system. The system is designed to detect lateral manoeuvres, or more specifically turns, and classify them as taken normally or aggressively. The diagram shows the three inputs of the system and their relation to the vehicle, namely the vehicle's lateral force of acceleration, a_x, rotation rate around its vertical axis, ω_z, and its forward velocity, v. Using only these three inputs, the system must be able to detect and classify a turn based on previously provided hand-annotated training data. Taking in filtered data, the endpoint detection block outputs signal vectors to the turn classifier. Two different turn classifiers are implemented and tested in the system.

3.2.1 Detection algorithm

Output from the accelerometer and gyroscope of the smartphone is used for manoeuvre recognition. For the detection of lateral manoeuvres, the a_x acceleration and rotation rate ω_z are used. The accelerometer and gyroscope are continuously sampled at a rate of 20 Hz, in line with [10].

Figure 3.2 shows a block diagram of the data processing before classification occurs. The accelerometer output is band-pass filtered to remove sensor noise and the gravitational force vector. The filter is designed with a very low cut-off frequency, as the gravitational force vector is essentially a DC offset in the signal. The vector's direction only fluctuates at times when the vehicle's roll and pitch changes while driving. The gyroscope output is only low-pass filtered to remove high frequency noise.

In order to detect manoeuvres, the start and end of driving events are determined by using the endpoint detection algorithm. For lateral manoeuvres, a simple moving average

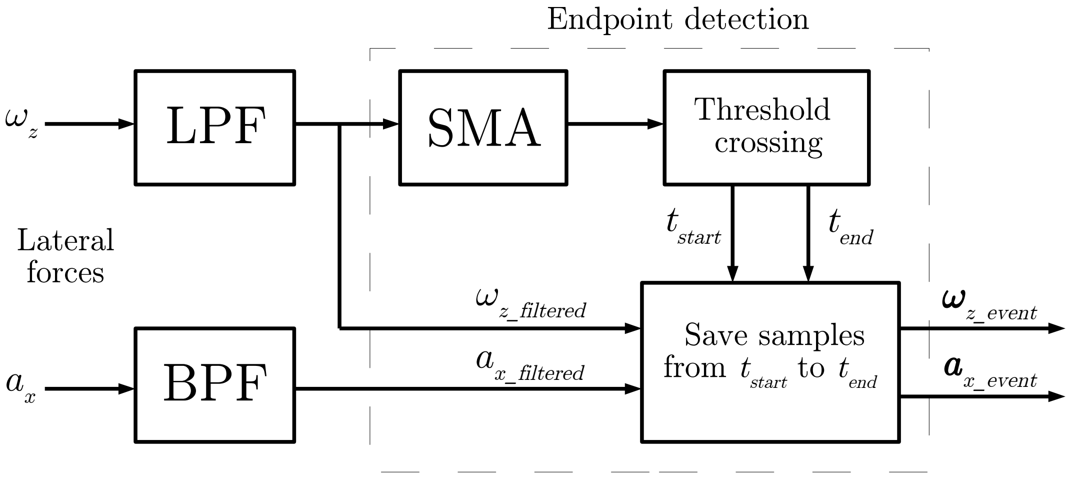


Figure 3.2: Subdiagram of the detection algorithm before classification occurs.

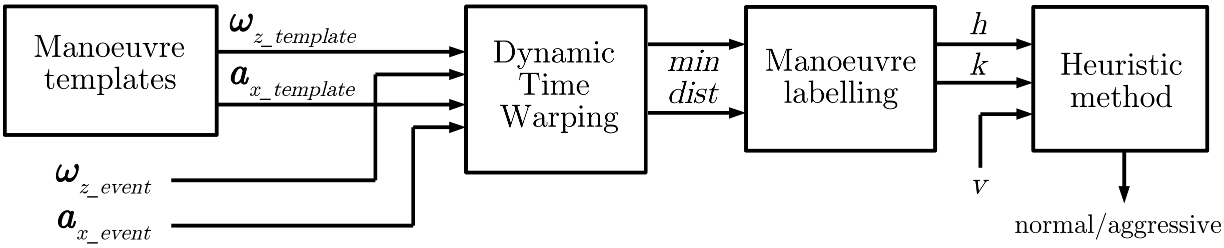


Figure 3.3: Subdiagram of the dynamic time warping classification approach.

(SMA) of ω_z is continuously calculated over a window of 40 samples. It was empirically found to smooth the signal enough to prevent short pulses in the signal from registering as start or endpoints. The beginning of a lateral event is detected if the SMA goes above a set threshold. The previous 40 and succeeding samples of ω_z are concatenated until the SMA falls again below the same threshold, signifying the end of the event. The samples of a_x are also saved during the same time window. An event is dismissed if it is less than 2.5 or more than 15 seconds long. This is to keep the algorithm from waiting too long for potentially erroneous or noisy data. The length boundaries were established empirically to detect most valid events.

3.2.2 Classification methods

As mentioned, two different classification methods are implemented in order to compare their performance. The first method uses dynamic time warping to compare detected events to driving manoeuvre templates, and then uses the results in a simple heuristic to classify the manoeuvres. The aim is to reproduce, to some extent, the dynamic time warping approaches found in existing literature. The second method uses supervised learning to train a maximum likelihood classifier to label driving manoeuvres. The maximum likelihood classifier was chosen, as it was found to be the most suitable supervised learning classifier for the recognition of aggressive driving.

3.2.2.1 Dynamic time warping approach

This approach is based on the work of Johnson and Trivedi [10], and Eren et al. [29]. The basis of the approach has already been described in Chapter 2, Section 2.2.3:

> “When a valid driving event has been detected, the signals recorded during the event are compared to a set of templates using the DTW algorithm. DTW finds an optimal alignment between two time-dependent sequences with different lengths. ...An optimal warping path constitutes the minimum sum of distances, while adhering to monotonicity, boundary and step size conditions. The template with the lowest minimum-distance warp path to the detected event is the closest match.”

The acceleration and rotation rate templates are discrete Gaussian-shaped signals with fixed lengths that were created from collected driving data. The ω_z templates indicate the angle of a turn. It allows the system to classify a left or right bend from 1 to 3, based on the closest matching ω_z template — with 1 indicating an easy bend, 2 a medium bend and 3 a sharp bend. Similarly there are six a_x templates with increasing amplitudes.

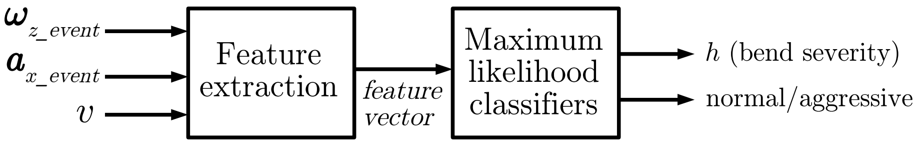


Figure 3.4: Subdiagram of the maximum likelihood classifier.

A heuristic method is used to label any recognised turn as taken normally or aggressively, based on the vehicle's speed (obtained from the GPS) and matching a_x and ω_z template. From experimental results it was evident that two conditions need to be satisfied to classify a turn as aggressive:

$$\begin{array}{ll} 1. & v > 50(3-h) \\ 2. & k > 4 \quad \text{or} \quad k > (h+2) \end{array}$$

where v is the vehicle's speed in km/h, h is the labelled bend severity and k is the a_x template number ranging from 1–6. The first condition is that the vehicle must be above the set speed limit for the bend severity; and secondly, a significant lateral force must be exerted on the vehicle, also depending on the bend severity.

Figure 3.3 shows a block diagram of the dynamic time warping classification approach as described. The ω_z and a_x signals of a detected event is used as input, as well as the vehicle's speed at the start of the event.

3.2.2.2 Maximum likelihood classifier

The maximum likelihood classifier is a non-linear statistical classifier. A training data set was prepared from collected driving data with which two classifiers were trained. The first classifier is trained to label the severity of a bend. The second classifier is trained to label a turn as taken normally or aggressively.

Figure 3.4 shows a block diagram of the maximum likelihood classification method. Feature selection is first performed on the ω_z and a_x signals of a detected event. The vehicle's speed is also given as an additional input. A representational feature vector is then given as input to the two classifiers for bend severity and driving style classification. The feature vector consists of the coefficients of a polynomial curve fitting performed on the ω_z signal, as well as it's fundamental frequency. The peak-to-peak amplitude, energy and fundamental frequency of the a_x signal are also selected as features.

Chapter 4

Detail Design

The detail system design is presented in this chapter. The hardware and software used to collect driving data with which the system was developed and tested is discussed. The data collection process is also described. Lastly, the software implementation of the detection algorithm and the two classification methods are discussed in further detail.

4.1 Data acquisition hardware and software

Two independent data acquisition systems were used simultaneously for data collection — an Android smartphone and a dedicated Arduino-based system.

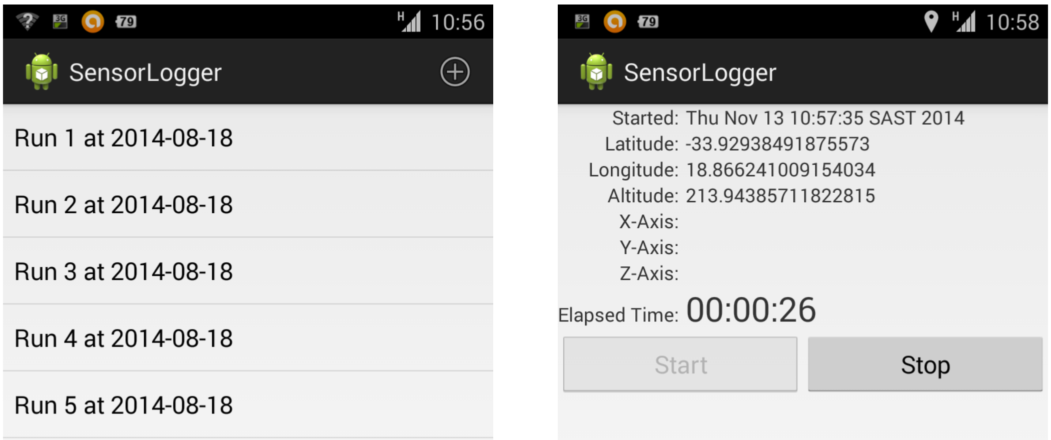
4.1.1 Smartphone and Android application

A Samsung Galaxy S3 smartphone was used for data collection. A simple data logger Android application was developed that samples the accelerometer and gyroscope at a minimum of 20 Hz. Although a higher sampling rate is possible, it increases processor usage, which can lead to the smartphone's operating system becoming unresponsive. Besides that, 20 Hz is considered fast enough for the proposed system, as manoeuvre classification would not benefit from any signal components above 10 Hz. The application saves the sensor samples and GPS data to an SQLite database. The Android OS does not allow direct access to a smartphone's sensors. Instead, a software interrupt provides sensor samples at a variable rate. The system must therefore be able to handle the possibly varying sample rates of the smartphone's sensors.

Tutorials from [46] were used as a basis from which to develop the Android application. Figure 4.1 shows two screenshots of the SensorLogger application.

4.1.2 Dedicated system

In order to validate the smartphone's data, an Arduino Uno board was used to also log data from a dedicated GPS and inertial measurement unit (IMU) to an SD card. The Pololu MinIMU-9 was used [47]. It is a small board with an L3GD20 3-axis gyroscope and an LSM303DLHC 3-axis accelerometer and 3-axis magnetometer, connected to an I^{2}C interface and voltage level shifting circuit. An ITEAD GPS shield was used as a dedicated GPS [48]. The shield features a GlobalSat EB-365 GPS module and a micro-SD card slot. The Arduino microcontroller communicates with the GPS module and SD card through UART and SPI respectively. The IMU was also sampled at 20 Hz and the



Run 1 at 2014-08-18

Run 2 at 2014-08-18

Run 3 at 2014-08-18

Run 4 at 2014-08-18

Run 5 at 2014-08-18

Started: Thu Nov 13 10:57:35 SAST 2014
Latitude: -33.92938491875573
Longitude: 18.866241009154034
Altitude: 213.94385711822815
X-Axis:
Y-Axis:
Z-Axis:
Elapsed Time: 00:00:26

(a) List view

(b) Run view

Figure 4.1: Screenshots of the developed SensorLogger application's two views.

data saved to the flash with the GPS timestamp, in order to compare the data to that of the smartphone.

Additionally, during preliminary tests, data from a u-blox 5 GNSS receiver was also recorded directly to a laptop via USB using u-blox's u-center software. It can record 3-D instantaneous velocity and heading in the earth's reference system at 4 Hz. Differentiating the velocity vector gives a good estimate of the vehicle's acceleration without the earth's gravitational acceleration. This is needed to test methods of removing the gravitational acceleration vector from the smartphone's accelerometer data. The u-blox's 3-D velocity is also used to validate the smartphone's speed estimate from its GPS, which receives updates once a second.

4.2 Data collection

Six people were invited to participate in the experiment as drivers. Each of them drove a pre-determined route while subjective labelling of their turns were performed by hand.

4.2.1 Driving route

All the participants drove the same route. A route of 15 km was chosen that has varying bends and up- and downhill parts. The route necessitates drivers to manage their speed as straighter sections are followed by several sharp bends. A map of the route is shown in Figure 4.2. The route follows a winding road to a point, and then back with the same road in the opposite direction. Therefore, each left bend has an opposing right bend and vice versa. All the distinct bends were annotated by hand on the map with a severity of 1, 2 or 3, and given a unique identifier. The route has 55 identified bends — 28 right and 27 left bends.

4.2.2 Preliminary data

Preliminary data was recorded with the smartphone, dedicated IMU and u-blox GNSS receiver in a fixed position and orientation in the vehicle. The smartphone's and IMU's axes were aligned to the vehicle's axes. A standard front-wheel drive sedan was used. The preliminary data was needed to test the accuracy of the smartphone's sensors and the

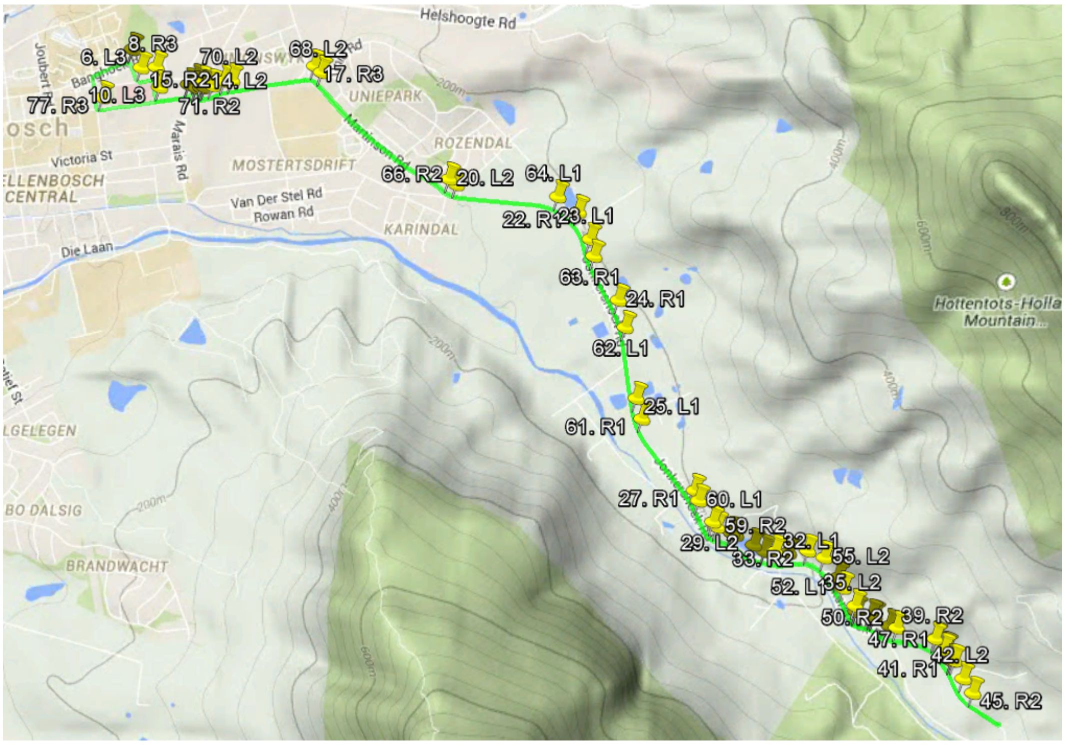


Figure 4.2: The route driven by each participant, showing all the hand-annotated bends.

reliability of the Android application. The u-blox's data was specifically used to validate the accuracy of a high-pass filter in removing the gravitational acceleration vector from accelerometer data. It was also used for sanity testing during the system development.

4.2.3 Labelled training and test set

The six participants each drove the route once or twice for the training and test data set. They each drove in their own vehicle to ensure familiarity and a normal driving style. A form with all the identified bends in sequential order was used for each run. The researcher labelled each bend as either taken normally or aggressively by the driver. Although such labelling is subjective by nature, it was done as consistently as possible. Only the smartphone was used to record data and it was kept in the same orientation and position in each vehicle, so that the smartphone's axes were aligned with the vehicle's axes.

All the collected data from the Arduino system and u-blox were parsed and stored in separate SQLite databases with the same structure as the smartphone's database. This and all further system development were done in Python 2.7.6.

4.3 Detection algorithm

The detection algorithm, as described here, detects lateral vehicle manoeuvres, but is designed to specifically detect vehicle turning, rather than lane changes or overtaking. The algorithm can be easily adjusted, though, to detect other vehicle manoeuvres. Various manoeuvres could then be detected by running multiple instances of the algorithm simultaneously.

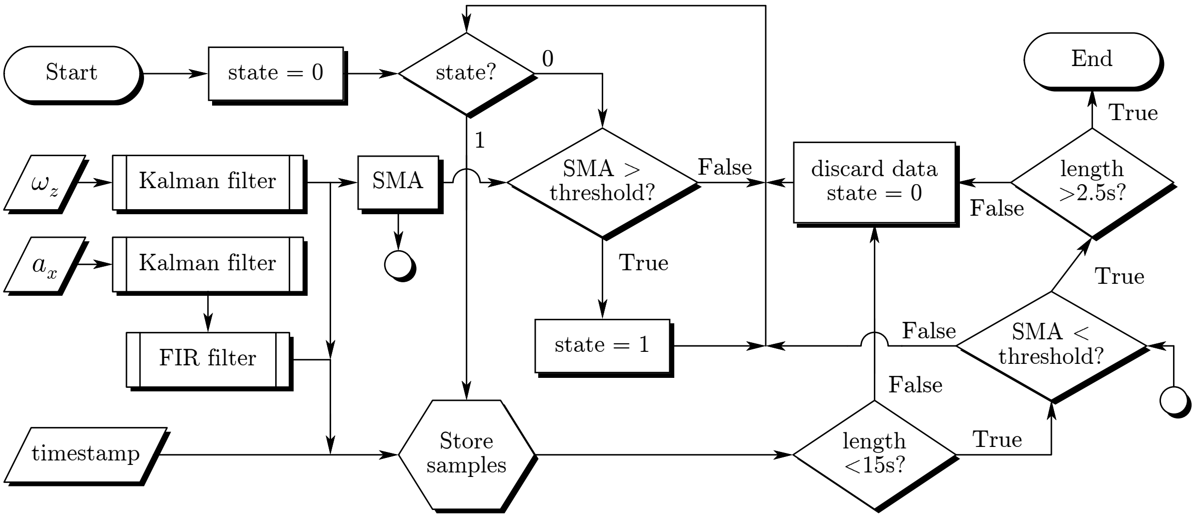


Figure 4.3: Flow diagram of the detection algorithm.

Figure 4.3 shows a flow diagram of the entire detection algorithm. It specifically shows the process of detecting and storing the data of a single event. The raw sensor data is first filtered before it is used by the algorithm. The details of the filters are discussed in the following subsections. To determine the start and endpoint of an event, a simple moving average (SMA) of ω_z is continuously calculated for a short window of 40 samples. From the current sample i, we have

$$\mathrm{SMA}(i) = \frac{\omega_z(i)^2 + \omega_z(i-1)^2 + ... + \omega_z(i-39)^2}{40} \tag{4.3.1}$$

The start of an event is detected if the SMA goes above a set threshold, and the end is detected when the SMA falls again below the same threshold. This process is known as endpoint detection. The algorithm goes through three states, where (i) it checks for the start of an event; (ii) it stores filtered sensor samples while monitoring for the end of the event; and (iii) it sends the recorded data of the event to the classifier. While in the second state, if the length of the event exceeds 15 seconds, the data is discarded and the algorithm returns to the first state. However, if the end of the event is found before 15 seconds passes, the algorithm enters the third state. In the third state, if the length of the event is found to be less than 2.5 seconds long, the data is discarded and the algorithm returns to the first state. In order to detect and distinguish between left and right turns, we run two instances of the algorithm with the sensor inputs inverted. In Figure 4.3, it is not explicitly shown that the data of a detected event is sent to the classifiers, but only that the data is stored.

4.3.1 Kalman filter

The first process in the system is to filter out noise from the smartphone's gyroscope and accelerometer data. A Kalman filter is used for this purpose [49]. It is an algorithm that recursively calculates an optimal estimate of a measurement from noisy input samples. It is simple to implement in software and it processes samples as they arrive, therefore it is feasible for use in a real-time system. Since only ω_z and a_x samples are used, only they are filtered.

Time update (Predict)

(1) Project the state ahead

$$\hat{x}_{k+1}^{-} = \hat{x}_k$$

(2) Project the covariance ahead

$$P_{k+1}^{-} = P_k + Q$$

Measurement update (Correct)

(1) Compute the Kalman gain

$$K_k = P_k^{-}(P_k^{-} + R)^{-1}$$

(2) Update estimate with measurement

$$\hat{x}_k = \hat{x}_k^{-} + K_k(z_k - \hat{x}_k^{-})$$

(3) Update the error covariance

$$P_k = (1 - K_k)P_k^{-}$$

Figure 4.4: The Kalman filter operates by cycling through the time update and measurement update equations recursively [50].

In general, the equations for the state $\mathbf{x}_k$ and measurement $\mathbf{z}_k$ of a discrete-time process are as follows:

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k \tag{4.3.2}$$

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{4.3.3}$$

The random variables $\mathbf{w}_k$ and $\mathbf{v}_k$ are the process and measurement noise, respectively. They are assumed to be independent and Gaussian. The equations for the Kalman filter are derived from the above equations. The matrix $\mathbf{A}_k$ relates the current state to the next state, while $\mathbf{B}_k$ relates the optional control signal $\mathbf{u}_k$ to the state. The matrix $\mathbf{H}_k$ relates the state to the measurement. These matrices can change with each measurement. For our model, however, the equations are simplified. Firstly, there is no control signal $\mathbf{u}_k$, and secondly, $\mathbf{A}_k$ and $\mathbf{H}_k$ are assumed to be constant and equal to 1. We define $\hat{x}^{-}$ as the *a priori* state estimate given the previous process, or in our model, just the previous state, and $\hat{x}$ as the *a posteriori* state estimate given measurement $\mathbf{z}_k$. The simplified equations used recursively to calculate the state estimate are shown in Figure 4.4. There are two filter parameters that can be tuned, namely the process noise Q and measurement error covariance R (sensor noise). The estimation error covariance P_k and Kalman gain K_k are dependent on the two filter parameters, and, in our model, converges if the parameters are kept constant.

4.3.2 Finite impulse response filter

After noise filtering, a finite impulse response (FIR) filter is used to filter out the gravitational acceleration vector from the accelerometer samples. Although the lateral axis of the vehicle is mostly perpendicular to the earth's center, banks in the road are enough for gravitational acceleration to add a significant offset to the a_x samples. A Kaiser window was used to create a high-pass FIR filter with a 3 dB cut-off frequency of 0.05 Hz. The Kaiser window is defined as

$$w[n] = \begin{cases} \frac{I_0\left(\beta\sqrt{1-(\frac{2n}{N}-1)^2}\right)}{I_0(\beta)}, & 0 \leq n \leq N, \\ 0, & \text{otherwise,} \end{cases} \tag{4.3.4}$$

where $I_0(\cdot)$ represents the zero-order modified Bessel function [51]. The Kaiser window has two parameters, namely the length $N+1$ and shape parameter $\beta = \pi\alpha$. The parameters

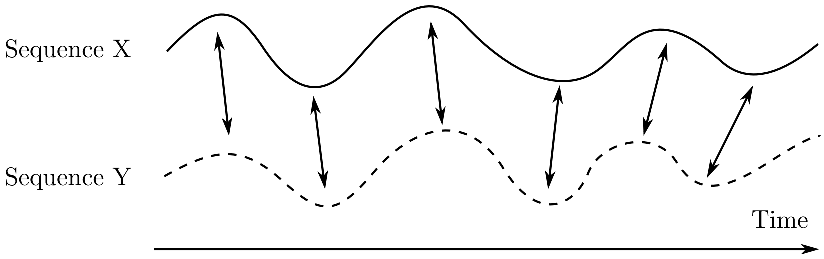


Figure 4.5: Alignment of two time-dependent sequences by points, as indicated by the arrows.

can be tuned to exchange main side-lobe amplitude for main-lobe width. The Kaiser window was chosen for its narrow main lobe width and steep cut-off. The 3-dB cut-off frequency of 0.05 Hz was determined empirically as efficient for filtering the gravitational force vector.

4.4 Classification methods

The same data set was used to train and test the two different classification methods. A Python module for machine learning, named *mlpy* [52], was used to implement the system. It is built on top of NumPy/SciPy and the GNU Scientific Libraries.

4.4.1 Dynamic time warping approach

Dynamic time warping is a technique to find an optimal alignment between two time-dependent sequences with different lengths. The two sequences are warped in a non-linear way to match each other, as shown in Figure 4.5. The DTW algorithm was originally developed to compare speech patterns in automatic speech recognition [41].

We have two sequences, $X := (x_1, x_2, \ldots, x_N)$ of length $N \in \mathbb{N}$ and $Y := (y_1, y_2, \ldots, y_M)$ of length $M \in \mathbb{N}$. A *feature space*, denoted as $\mathcal{F}$, is set for the sequences that $x_n, y_m \in \mathcal{F}$ for $n \in [1 : N]$ and $m \in [1 : M]$. To compare two different elements $x, y \in \mathcal{F}$, a *local cost measure* is needed. It is defined as a function

$$c : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_{\geq 0} \tag{4.4.1}$$

The local cost measure is calculated for each pair of elements of X and Y to obtain the *cost matrix* $C \in \mathbb{R}^{N \times M}$, as shown in Figure 4.6, where $C(n, m) := c(x_n, y_m)$. Low cost is dark coloured and high cost is light coloured. Both sequences start at the bottom left corner. An alignment is defined as a *warping path* $p = (p_1, p_2, \ldots, p_K)$ with $p_k = (n_k, m_k) \in [1 : N] \times [1 : M]$ for $k \in [1 : K]$, adhering to the following three conditions:

1. Boundary condition: $p_1 = (1, 1)$ and $p_K = (N, M)$
2. Monotonicity condition: $n_1 \leq n_2 \leq \ldots \leq n_K$ and $m_1 \leq m_2 \leq \ldots \leq m_K$
3. Step size condition: $p_{k+1} - p_k \in \{(1, 0), (0, 1), (1, 1)\}$ for $k \in [1 : K - 1]$

The boundary condition implies that the first elements of X and Y, as well as the last elements, must be aligned to each other. The monotonicity condition enforces true timing, the path cannot move backwards with an element of X or Y. The step size condition represents continuity in that no element of X or Y can be excluded and each element pair

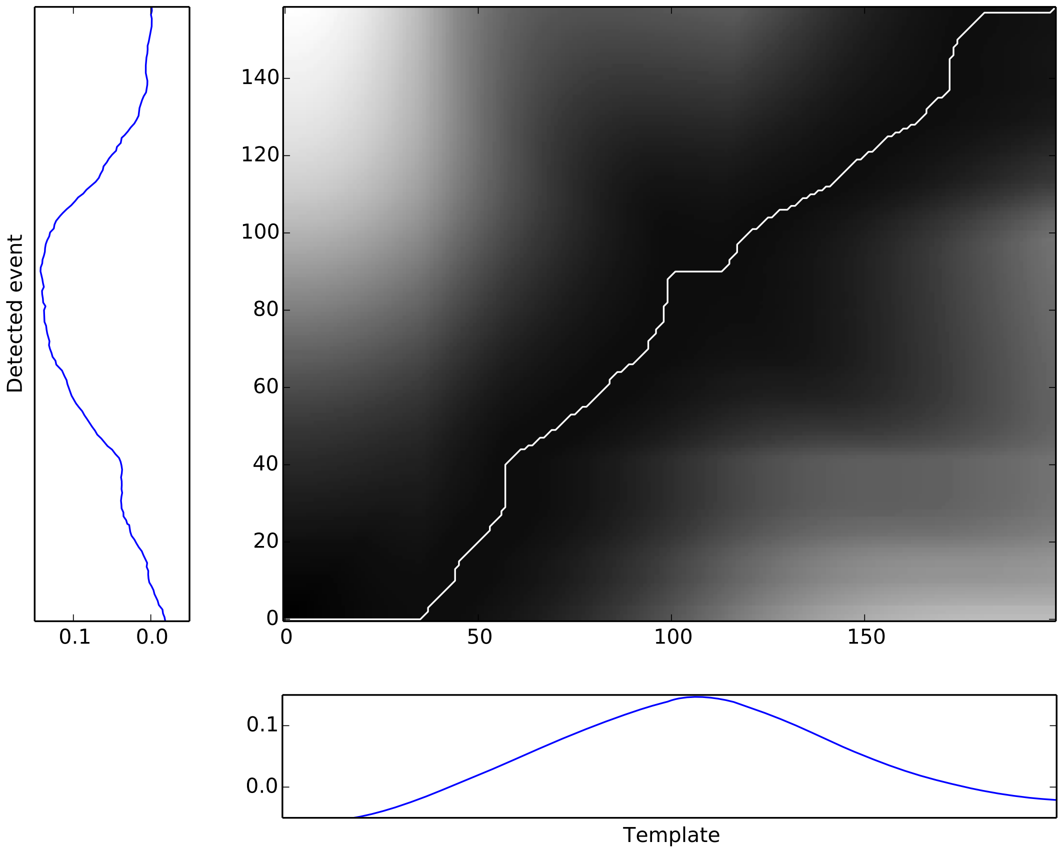


Figure 4.6: Cost matrix of two discrete ω_z signals using the Euclidean distance as cost measure. The optimal warping path is indicated by the white line.

must be unique in the warping path. The objective of DTW is to find an *optimal warping path* p^* between X and Y that has the minimal overall cost of all possible warping paths. The *DTW distance* is then defined as the total cost of p^*

$$DTW(X,Y) := c_{p^*}(X,Y) := \sum_{k=1}^{K} c(x_{n_k}, y_{m_k}) \tag{4.4.2}$$

When a valid driving manoeuvre has been detected, the signals recorded during the event are compared to a set of templates using DTW. The template with the lowest optimal warping path cost to the detected event is the closest match. It allows the system to classify a left or right bend from 1 to 3 — with 1 indicating an easy bend, 2 a medium bend and 3 a sharp bend. The acceleration and rotation rate templates are discrete Gaussian signals with fixed lengths that were created from collected driving data. There are six ω_z and a_x templates representing the 3 severities of bends taken both normally and aggressively. The ω_z and a_x templates are arranged in two separate sets and sorted in increasing amplitude. The ω_z templates are 3 pairs of 2 corresponding to the bend severity defined as h.

4.4.1.1 Classification heuristic

A heuristic method is used to also classify any recognised turn as taken normally or aggressively, based on the vehicle's speed (obtained from the GPS) and the matching a_x and ω_z template. From experimental results it was evident that two conditions need to be satisfied to classify a turn as aggressive:

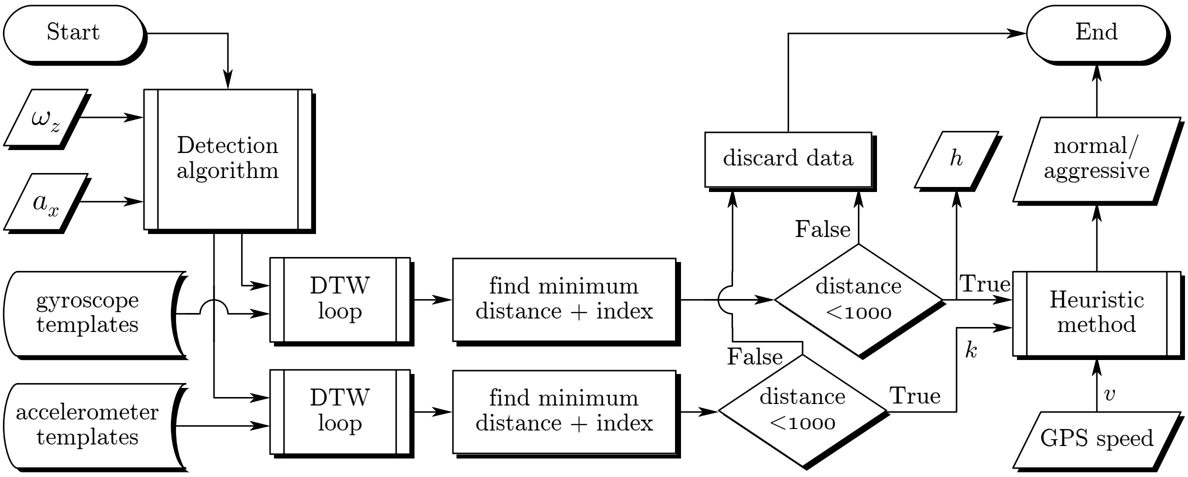


Figure 4.7: Flow diagram of the dynamic time warping classification method.

$$\begin{array}{ll} 1. & v > 50(3-h) \\ 2. & k > 4 \quad \text{or} \quad k > (h+2) \end{array} \tag{4.4.3}$$

where v is the vehicle's speed in km/h, h is the labelled bend severity from 1–3 and k is the a_x template number from 1–6.

Figure 4.7 shows a flow diagram of how the dynamic time warping classification method operates on a single event. The ω_z and a_x signals of a detected event is used as input, as well as the vehicle's speed at the start of the event. The DTW distance between the sensor signals and each template is calculated in a loop. The minima of the six distances are determined. If any of the minima exceed the empirically determined threshold, it is considered as an anomaly and the event is discarded. Otherwise, the template numbers of the minimum distances are sent to the heuristic method for style classification of the turn.

4.4.2 Maximum likelihood classifier

Maximum likelihood (ML) estimation is an algorithm that estimates (or *learns*) the parameters of a statistical model. The set of parameters $\hat{\boldsymbol{\theta}}$ under which the data $\{\mathbf{x}_i\}_{i=1}^{I}$ are most likely is equal to the product of the likelihood functions at each individual data point $\mathbf{x}_i$. The likelihood function $P(\mathbf{x}_i|\boldsymbol{\theta})$ is obtained by assessing the probability density function at $\mathbf{x}_i$. The maximum likelihood estimate of the parameters therefore is

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} P(\mathbf{x}_i|\boldsymbol{\theta}) \right], \tag{4.4.4}$$

where $\operatorname{argmax}_{\boldsymbol{\theta}} f[\boldsymbol{\theta}]$ returns the value of $\boldsymbol{\theta}$ that maximises the argument $f[\boldsymbol{\theta}]$. [53]

After the parameters $\hat{\boldsymbol{\theta}}$ of a model have been determined, they can be used for binary classification of new data. For this purpose, a training data set with selected features and class labels was pre-processed from collected driving data. It was used for supervised learning of two separate maximum likelihood classifiers. The first classifier is trained to label the severity of a bend. The second classifier is trained to label a turn as taken normally or aggressively. The trained classifiers can be used to label driving manoeuvres immediately after they are detected in a real-time system.

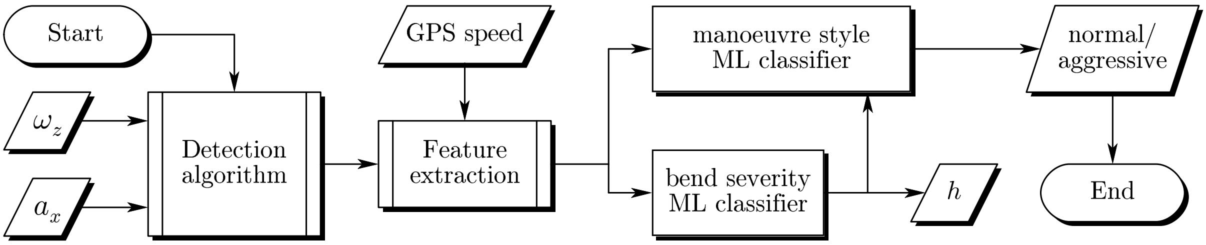


Figure 4.8: Flow diagram of the maximum likelihood classification method.

4.4.2.1 Feature selection

Figure 4.8 shows a flow diagram of how the maximum likelihood classification method would operate in a real-time system. As with the DTW method, the ω_z and a_x signals of a detected event are also used as inputs, as well as the vehicle's speed at the start of the event. However, since the data of a single event consists of a few hundred samples, it needs to be transformed to a smaller set of features to be used as input. This feature selection is a crucial part of the system. Combinations of different selected features were tested initially to acquire a feature set for the best results. The preliminary testing is discussed in Section 5.4.1 of Chapter 5.

Separate feature vectors are selected for each classifier. For the bend severity ML classifier, a 5th order polynomial function is fitted to the gyroscope signal. The six normalised coefficients of the fitted function are used as the input feature vector. For the aggressive manoeuvre ML classifier, features are selected from both the accelerometer and gyroscope signal. The minimum peak to maximum peak difference $\mathbf{a}_{x(pk-pk)}$, and energy $E_{\mathbf{a}_x}$, of the accelerometer signal are calculated and used as features. The magnitudes of both the accelerometer and gyroscope signals' fundamental frequencies are also selected as features. Additionally, the GPS speed v, and output of the bend severity classifier h, are also used as features. A total of six features are therefore used for the manoeuvre style ML classifier. Table 4.1 provides a summary of the features and classes of each classifier. All the features are rescaled to make them independent and carry equal weight. Each feature is rescaled in the range $[0, 1]$ using the general formula

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad , \tag{4.4.5}$$

where x' is the normalised value of x, and the minimum and maximum values were obtained from the training data set.

Table 4.1: Summary of the classifiers' features and classes.

Classifier	Bend severity	Aggressive manoeuvre
Features (input)	Coefficients of $ax^5 + bx^4 + cx^3 + dx^2 + ex + f$, polynomial curve fitting to $\boldsymbol{\omega}_z$	Peak-to-peak amplitude, $\mathbf{a}_{x(pk-pk)}$ Acceleration signal energy, $E_{\mathbf{a}_x}$ Forward speed, v, from GPS Fundamental frequency magnitude of $\mathbf{a}_x$ Fundamental frequency magnitude of $\boldsymbol{\omega}_z$ Classified bend severity, h
Classes (output)	$h = 1, 2, 3$	normal, aggressive $= 1, 2$

Chapter 5

Tests and results

The results and findings of all the experimental testing are presented in this chapter. The validity of the collected data and pre-processing thereof is discussed. Secondly, the performance of the implemented sensor filters and detection algorithm is evaluated. Lastly, the preliminary testing of the different classification methods are discussed and the final test results are presented and evaluated.

5.1 Validation of smartphone data

Data was recorded simultaneously on the smartphone and a dedicated GPS and IMU unit in order to check the validity of the smartphone's data. It was also used to ascertain time synchronization between the GPS data and sensor samples of the smartphone. The sensor sampling rate of the smartphone was found to generally vary slightly around the set 20 Hz. The sensor sampling did at times slow down below 20 Hz on the smartphone, or even stopped completely. After investigating the problem, it was found that the UI of the application became unresponsive during times where sampling slowed down. The application was clearly too processing intensive. The concurrent sampling and saving of sensor data on the main thread of the application was eventually narrowed down as the probable cause of the problem. The application was updated to perform all SQLite database related workload on a separate thread, which proved to alleviate the problem. After the update, for an unknown reason, the accelerometer sampling rate doubled to about 40 Hz, while the gyroscope sampling rate remained at 20 Hz.

5.2 Collected data set

A Python script was written to process the raw data of the final collection into an usable data set. The continuously recorded sensor data of each driver's runs is put through the endpoint detection algorithm to find the timestamps of possible turns. The coordinates of the detected manoeuvres are compared to the coordinates of the hand-annotated bends that is shown in Figure 4.2. Each detected manoeuvre that matches the location of a bend on the map is labelled with the corresponding bend severity and unique identifier. The rest of the detected manoeuvres are discarded, as they are either anomalies or subtle bends on the route that were never marked for classification. The identifiers of the bends labelled by hand as taken aggressively during the runs of each driver is saved in a file. The identifiers of all the matching turns are checked against this file in order to label each turn as either taken normally or aggressively.

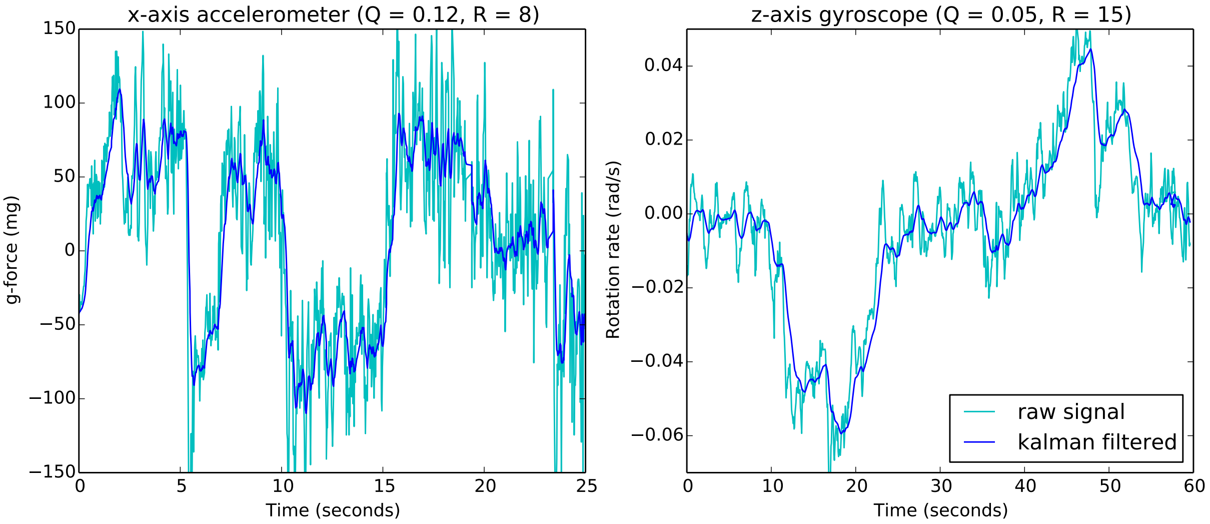


Figure 5.1: Kalman filtering of the x-axis accelerometer and z-axis gyroscope signals with the given filter parameters.

Using this automated method, valid data was successfully extracted and labelled for 387 bends. The data set was split in a 2:1 ratio for training and test data respectively.

5.3 Detection algorithm

The performance of the Kalman filter and finite impulse response (FIR) filter is evaluated in this section. The overall performance of the detection algorithm is also discussed, and examples of the typical signals that are recorded are given.

5.3.1 Kalman filter

Figure 5.1 shows an example of the accelerometer and gyroscope signal before and after Kalman filtering has been performed. It works well to smooth the signal without causing a delay. As can be seen, the accelerometer signal has a higher noise floor than the gyroscope signal, probably because it is more susceptible to engine and road vibrations. The accelerometer signal is filtered with $Q = 0.12$ and $R = 8$, while the gyroscope signal is filtered with $Q = 0.05$ and $R = 15$.

5.3.2 Finite impulse response filter

Figure 5.2 shows the frequency response of the implemented high-pass FIR filter. The filter is designed to filter out the DC offset in the accelerometer signal caused by the gravitational force vector. A Kaiser window with $\beta = 5.65$ is used to obtain a 3 dB cut-off frequency of 0.05 Hz. Figure 5.3 shows an example from the result of applying the filter to the y-axis accelerometer signal. A twenty second window of driving down a small hill is shown. As can be seen, the filter removes the offset caused by the gravitational force vector while the vehicle's pitch is not perpendicular to it. The mean acceleration of the entire data set is reduced from 0.424 m/s^2 to 0.0016 m/s^2 after applying the filter.

The 3-D instantaneous velocity readings from the u-blox GNSS receiver is also used to validate the effectiveness of the filter. The velocity is differentiated to obtain the vehicle's dynamic acceleration in the y-direction without the static gravitational acceleration.

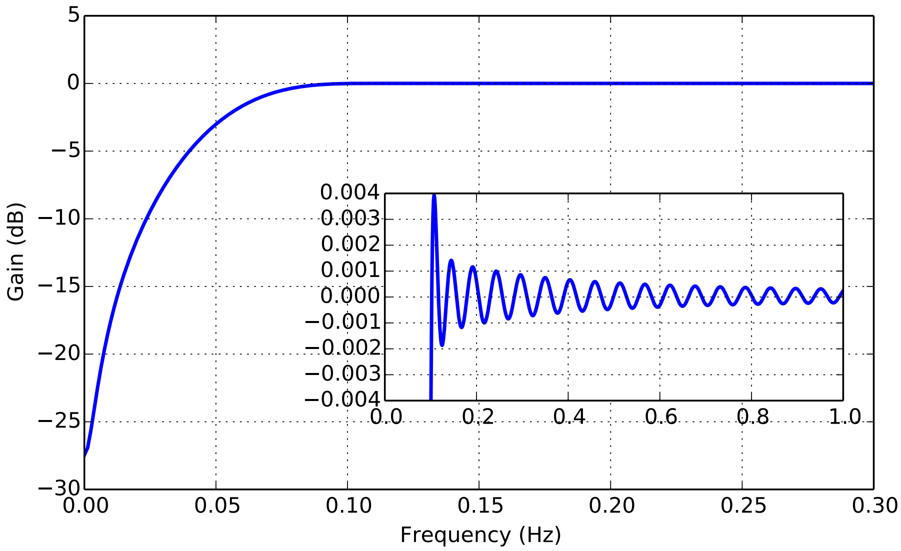


Figure 5.2: Frequency response of the high-pass FIR filter using a Kaiser window with $\beta = 5.65$.

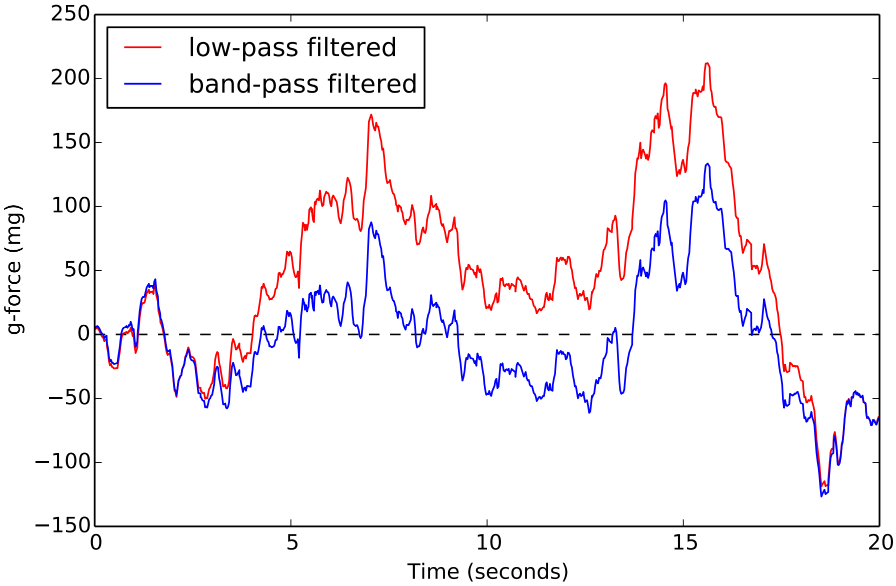


Figure 5.3: The y-axis accelerometer signal while driving down a small hill, before and after high-pass FIR filtering.

Figure 5.4 shows a comparison of this acceleration signal with the low-pass and band-pass filtered y-axis accelerometer signal for an one minute window. The same result as in Figure 5.3 is seen, with the band-pass filtered accelerometer signal's level being much closer to that of the u-blox's velocity gradient.

5.3.3 Endpoint detection algorithm

Overall, the endpoint detection algorithm successfully detected 95% of the left and right bends. However, sporadic time de-synchronisation in the recorded smartphone data caused problems with the start- and endpoints of the accelerometer signal versus the gyroscope signal. The gyroscope signal is always windowed by the algorithm as expected,

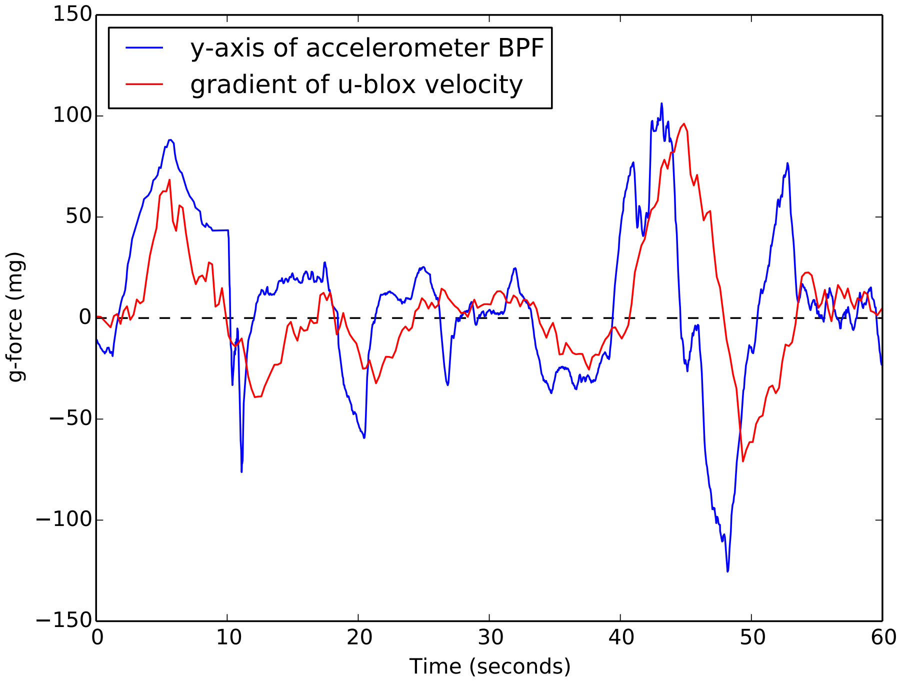


Figure 5.4: The filtered y-axis accelerometer signals versus the gradient of the u-blox velocity.

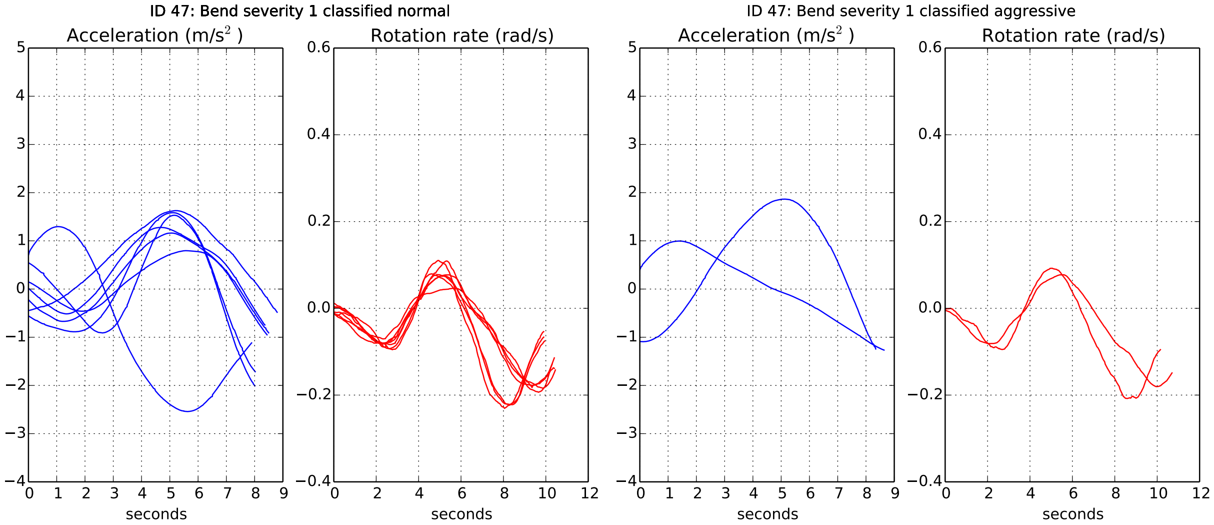


Figure 5.5: The accelerometer and gyroscope signals of multiple times a specific severity 1 bend was taken normally and aggressively.

but during the same time window of the accelerometer signal, its anticipated corresponding peak might be off center and cut in half. This leads to misrepresenting features from the accelerometer signal. Therefore we experimented with widening the window of the accelerometer signal, to ensure that the representational part is captured.

Figures 5.5, 5.6 and 5.7 shows examples of the accelerometer and gyroscope signals recorded by the detection algorithm. Signals from when the different drivers took the same bend are overlaid on top of each other. All the signals start at 0 seconds on the graphs, although their lengths differ slightly. Three examples of where the accelerometer is not synchronised with the gyroscope can be seen.

Figure 5.8 is a map showing the GPS traces of all the detected turns in the data set

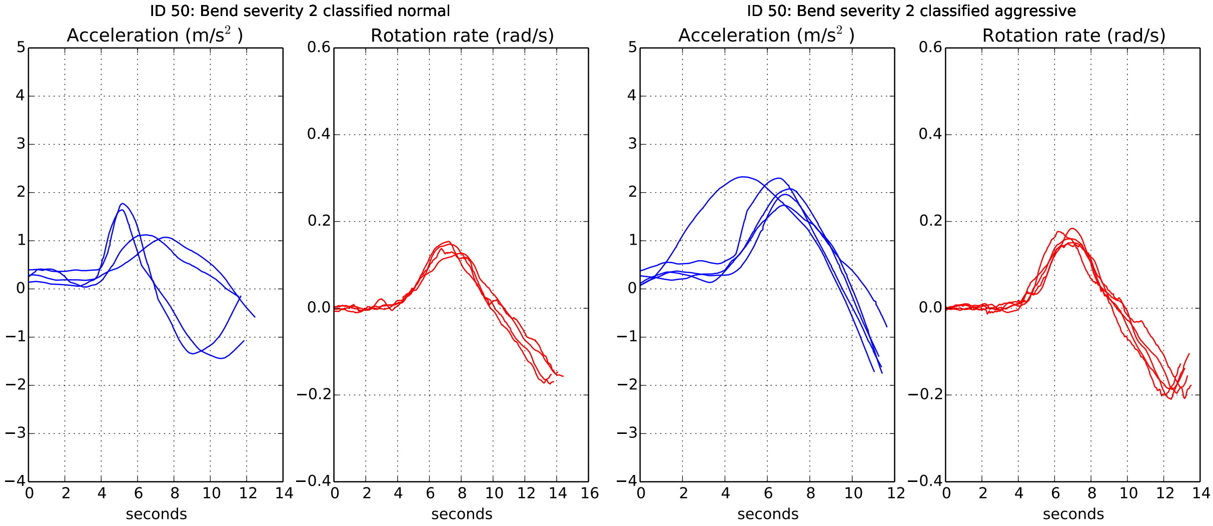


Figure 5.6: The accelerometer and gyroscope signals of multiple times a specific severity 2 bend was taken normally and aggressively.

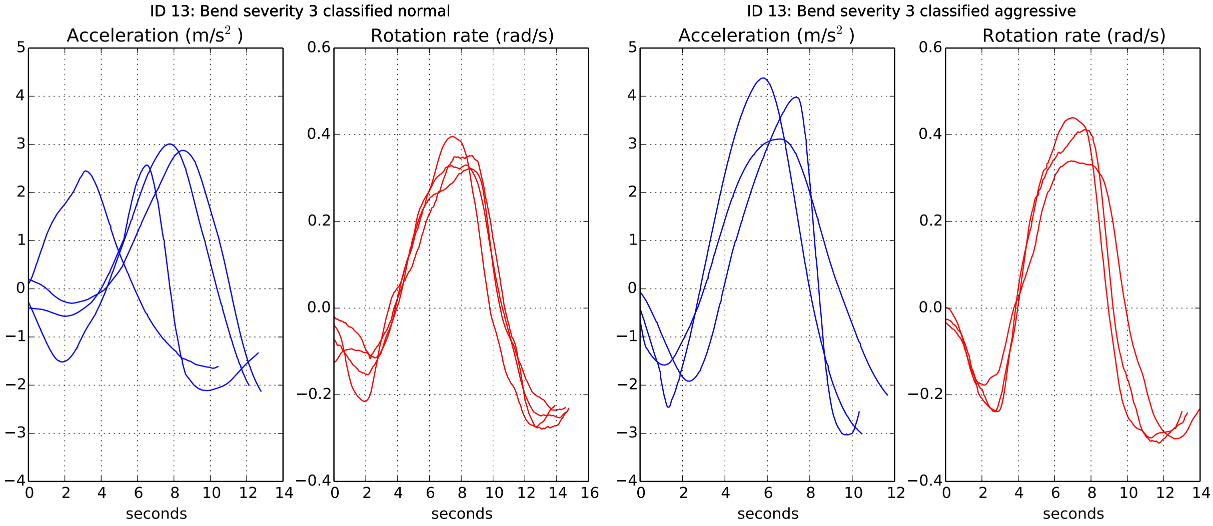


Figure 5.7: The accelerometer and gyroscope signals of multiple times a specific severity 3 bend was taken normally and aggressively.

around a large traffic circle on the route. All of these turns have been successfully labelled with the correct bend severity by the method as described in Section 5.2. The yellow pins are the hand-annotated coordinates that the detected turns are matched to. The red pins and green lines show the starting position and course of each turn.

5.4 Classification methods

All the preliminary testing and the final test results of the classification methods are discussed in this section. The performance testing of different feature selections and supervised learning classifiers are first discussed. The best selected feature set and classifier is then compared to the implemented dynamic time warping classification approach with a final test.

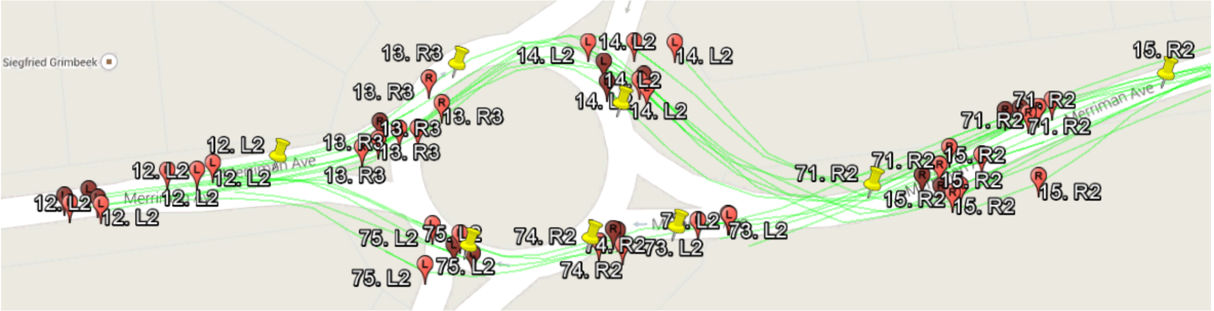


Figure 5.8: Detected turns that have been successfully labelled with the correct bend severity and that can be used as training data.

5.4.1 Preliminary testing

Preliminary testing was done with a few different supervised learning classifiers to determine which is the best for driving manoeuvre classification. Thorough testing of different feature sets were also performed to determine the best combination. These tests were done concurrently and repeatedly to refine the system's performance.

5.4.1.1 Feature selection

Since detected events typically consist of a few hundred samples from the smartphone's accelerometer and gyroscope, the data must be reduced to a small subset of relevant features to be used for classification. Various combinations of selected features were tested to find the best representation of normally versus aggressively taken turns. The list of tested features taken from both the accelerometer and gyroscope signals are:

- Polynomial curve fitting coefficients.
- Minimum, average and maximum amplitudes.
- Minimum peak to maximum peak amplitude.
- Signal energy.
- Fundamental frequency and its magnitude.
- Integral of gyroscope signal as estimation of rotation.

Figure 5.9 shows the results of fitting 5th-order polynomial curves to the accelerometer and gyroscope's signals of detected turns. Examples of a severity 2 and 3 bend, each taken both normally and aggressively, are given. The fitted curves generally match the signals quite well and can be considered as a good representation.

Also consider that, between the normally and aggressively taken turns, there is a notable difference in the peak acceleration, but not in the peak rotation rate. Amplitude features from the accelerometer signal are therefore usable, but not from the gyroscope signal. Signal energy is defined as

$$E_{\mathbf{x}} = \frac{T}{N} \sum_{n=0}^{N-1} |x[n]|^2 \quad , \tag{5.4.1}$$

where T is the duration of the signal and N the number of samples. From Figure 5.9 it can also be seen that the accelerometer signal energy is more for an aggressive turn than a normal turn by looking at the peak amplitude.

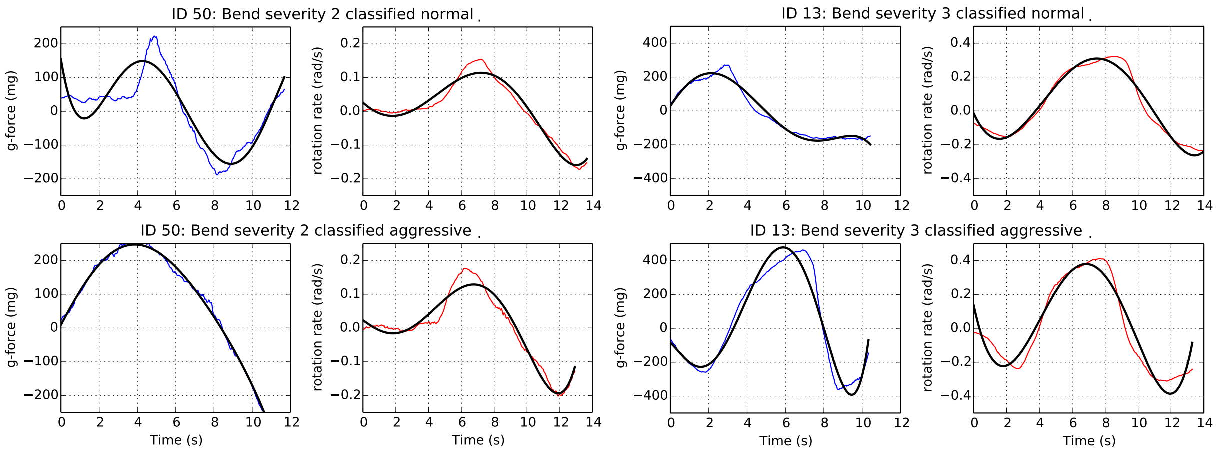


Figure 5.9: Curve fitting to the accelerometer and gyroscope signals for specific bends taken normally and aggressively.

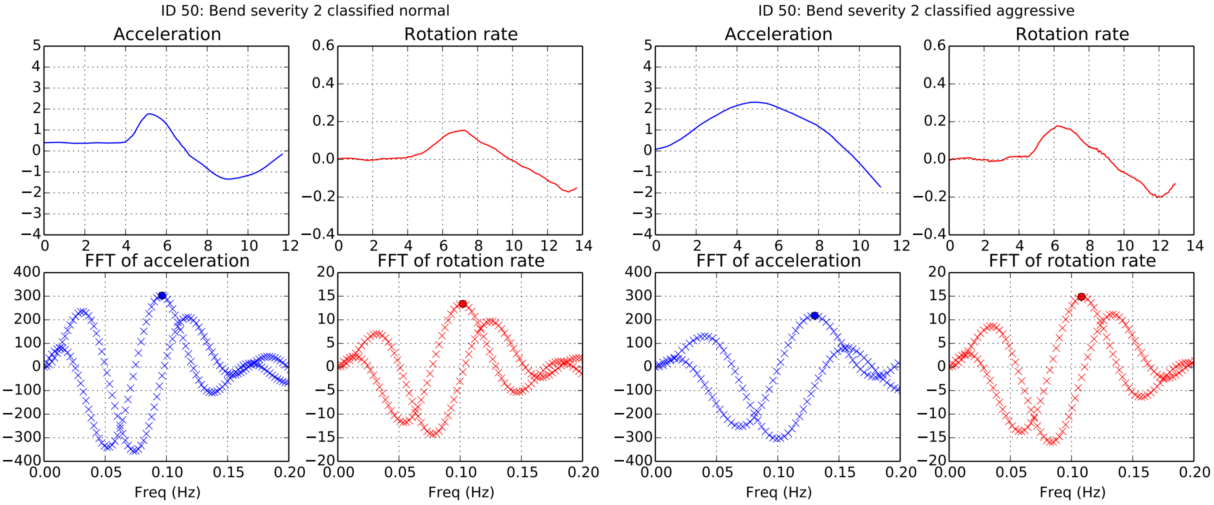


Figure 5.10: The accelerometer and gyroscope signals, and their FFTs, for a specific bend taken normally and aggressively.

Figure 5.10 shows signal plots from a specific bend. The four plots on the left shows an example of the accelerometer and gyroscope signals, and their fourier transforms (FFT), of when the bend was taken normally. The plots on the right shows an example of when the same bend was taken aggressively. In Table 5.1, the medians were calculated of all the accelerometer and gyroscope signals' fundamental frequencies (f_0) for each bend severity taken normally and aggressively. In general, for each type of turn, the mean f_0 of the accelerometer signals is lower for the higher bend severities. The difference in the mean f_0 of the gyroscope signals is negligible, indicating a weak correlation of the signals' f_0 to bend severity and aggressiveness. The standard deviation (SD) from the mean f_0 for both signals of each type of turn is at least 27% or more, further pointing to a weak correlation. There is a considerable difference in the mean magnitude of f_0 between the three bend severities for both the accelerometer and gyroscope signals. There is also a larger difference in the mean magnitude between normal and aggressive turns than with the mean f_0, from which we can surmise that the magnitude of f_0 is a better feature than f_0 itself.

Table 5.2 shows the performance measures obtained by using different combinations of features for aggressive manoeuvre classification. For the sake of simplicity, minimum, average and maximum amplitudes are not included, as it was found that peak-to-peak amplitude, on its own, always resulted in better performance than any combination of amplitude features. The results also show that, as expected, the fundamental frequency magnitude is a better feature than the fundamental frequency itself. As mentioned in Section 4.4.2.1, given these results, the peak-to-peak amplitude $\mathbf{a}_{x(pk-pk)}$, energy $E_{\mathbf{a}_x}$, and f_0 magnitude of the accelerometer signal, as well as the f_0 magnitude of the gyroscope signal, the GPS speed v, and output of the bend severity classifier h, are selected as the best feature set.

Table 5.1: The mean fundamental frequency (f_0), mean magnitude of f_0, and standard deviations (SD) of the accelerometer signals.

		Mean f_0 (mHz)		**SD of f_0 (mHz)**		**Mean magnitude**		**SD of magnitude**	
	Bend severity	Normal	Aggressive	Normal	Aggressive	Normal	Aggressive	Normal	Aggressive
Accelerometer	1	120	102	51	41	84	188	62	16
	2	99	97	31	26	202	265	103	111
	3	82	84	32	31	265	455	129	218
Gyroscope	1	102	132	35	9	5.9	8.6	2.9	2.1
	2	104	109	17	20	12.5	14	5.0	4.2
	3	113	108	27	34	27.5	29	8.2	13.3

Table 5.2: Performance measures (%) of different combinations of feature sets for aggressive turn classification.

Precision	**Recall**	**Specificity**	**Accuracy**	$\mathbf{a}_{x(pk-pk)}$	$E_{\mathbf{a}_x}$	v from GPS	h	f_0 mag of $\mathbf{a}_x$	f_0 of $\mathbf{a}_x$	f_0 mag of $\boldsymbol{\omega}_z$	f_0 of $\boldsymbol{\omega}_z$	Area of $\boldsymbol{\omega}_z$
75	50	96	88	×	×	×	×	×				
80	50	97	88	×	×	×	×	×			×	
86	50	98	89	×	×	×	×	×		×		
75	50	96	88	×	×	×	×	×				×
79	46	97	88	×	×	×	×	×		×		
56	42	92	83	×	×	×		×	×			
70	67	93	88	×	×	×			×			

Table 5.3: Performance measures (%) of different supervised learning classifiers.

	Aggressive manoeuvre				**Bend severity**
Classifier	Precision	Recall	Specificity	Accuracy	Accuracy
Maximum likelihood	85.7	50.0	98.1	89.1	95.3
k-Nearest-neighbour	66.7	33.3	96.2	84.5	91.5
Classification tree	52.6	41.7	91.4	82.2	80.6
LD Analysis	90	37.5	99	87.6	92.2
DLD Analysis	60	50	92.4	84.5	75.2

5.4.1.2 Supervised learning classifiers

Table 5.3 provides a comparison of a few different supervised learning classifiers in successfully classifying manoeuvres. The same data set with the same selected features was used to train and test each classifier. The given performance measures were obtained by creating a confusion matrix. The maximum likelihood (ML), k-nearest neighbour and classification tree classifiers have non-linear decision boundaries. The linear discriminant analysis (LDA) and diagonal linear discriminant analysis (DLDA) classifiers have linear decision boundaries.

The ML classifier achieves the best compromise between precision and recall with aggressive manoeuvre classification. It also has the highest accuracy with both classifications. The LDA classifier's performance comes very close to that of the ML classifier. The best way to compare the classifiers' trade-off between precision and recall is calculating a weighted average, or F-measure, defined as

$$F_\sigma = (1+\sigma^2)\cdot\frac{\text{precision}\cdot\text{recall}}{(\sigma^2\cdot\text{precision})+\text{recall}} \tag{5.4.2}$$

where if $\sigma < 1$, it increases the weight of precision. The balanced F_1 score of the ML and LDA classifiers are 0.632 and 0.529, respectively, for aggressive manoeuvre classification. However, precision is considered more important than recall, because it is considered unfair to label a driver as aggressive based on false positively identified aggressive manoeuvres. Therefore we also consider the $F_{0.5}$ score, which weights precision higher than recall. The $F_{0.5}$ scores are 0.714 and 0.6 for the ML and LDA classifiers respectively. Based on these F-measures, the ML classifier is the best suited for our aggressive driving model.

5.4.2 Final tests and comparison

Detailed results of the dynamic time warping approach and maximum likelihood classifier on the test data set is given and discussed in this section. The two methods' performance, advantages, and room for improvement and expansion are compared. The complete data set was first sorted by bend severity, and then by normal and aggressive turns, before splitting it into the training and test set. This ensures that the different combinations of labelled turns are evenly distributed between the sets. The final test set contains 129 turns, of which 24 are labelled aggressive. In addition to the performance measures, the confusion matrices from which they are calculated are given. A confusion matrix shows the number of true positive, true negative, false positive and false negative aggressive

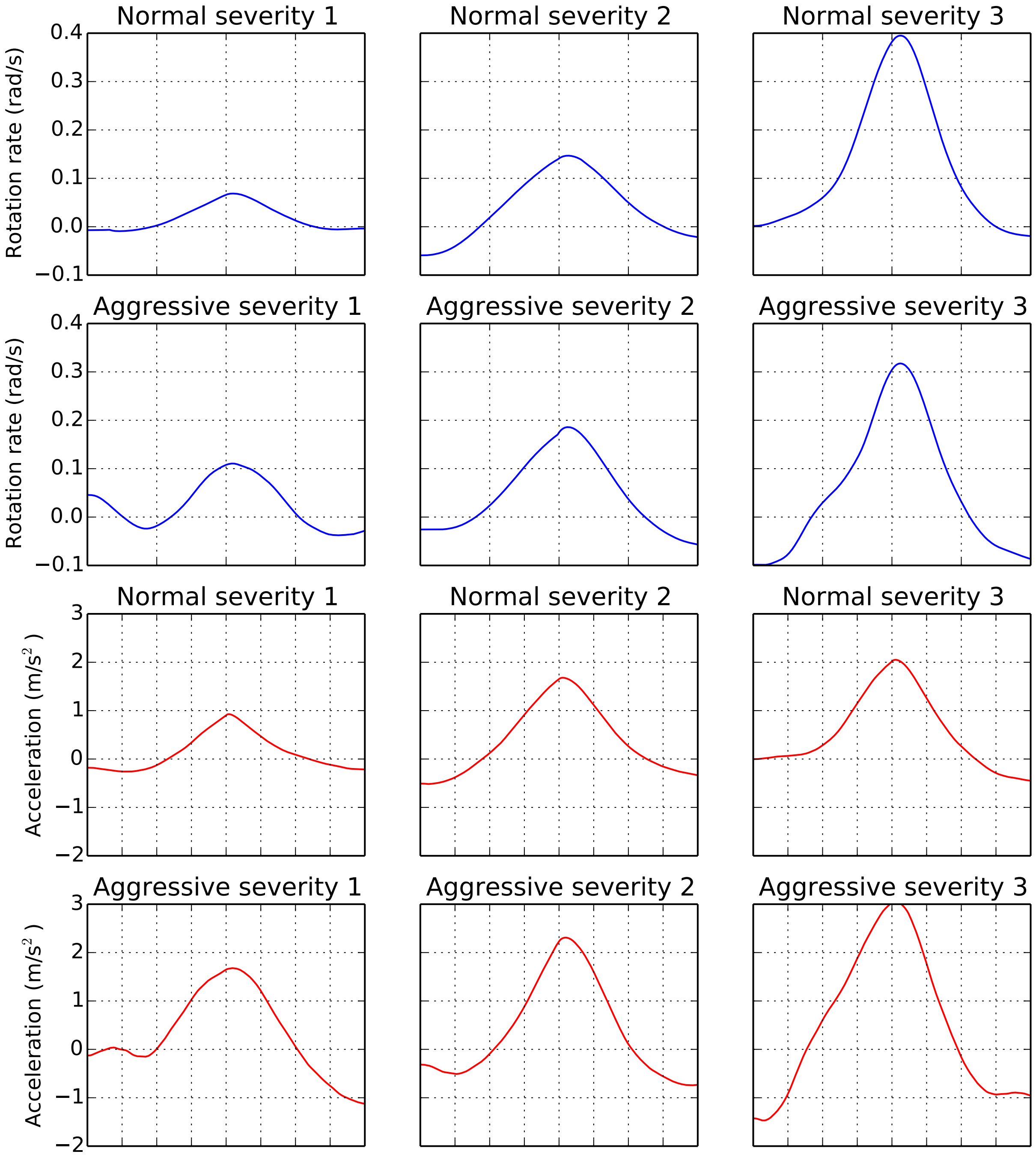
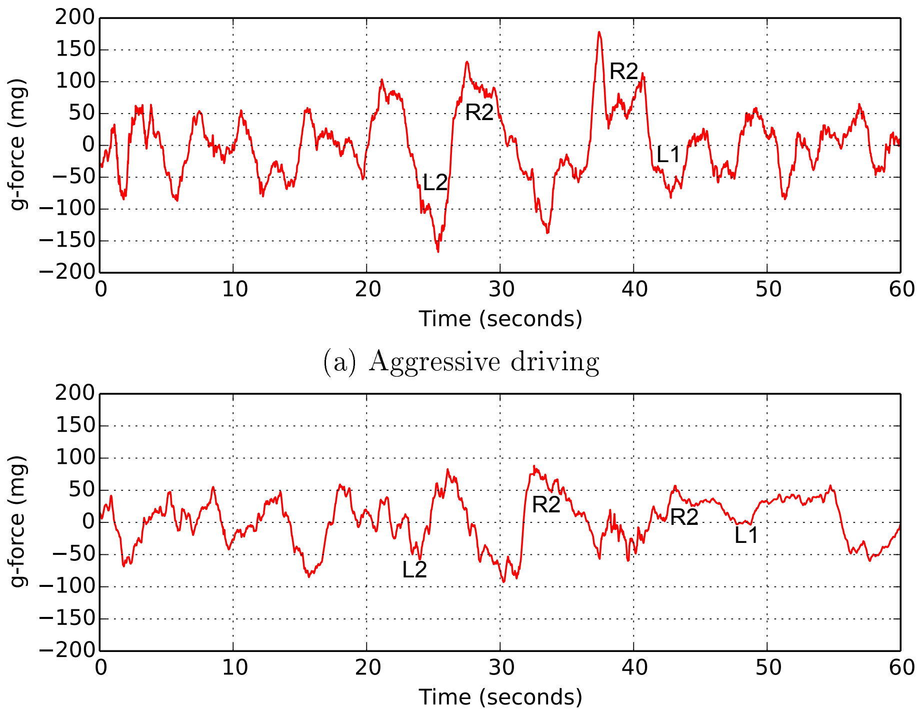


Figure 5.11: Gyroscope and accelerometer signal templates for the three bend severities taken both normally and aggressively.

turns. The number of true positive and false negative turns adds up to the 24 aggressively labelled turns in the test set, while all four values together adds up to the total number of 129 turns.

5.4.2.1 Dynamic time warping approach

The training data set was used to create gyroscope and accelerometer signal templates for the three bend severities taken both normally and aggressively. Twelve templates were thus created from the gyroscope and accelerometer data in total. A function was written to automatically generate the templates from the training data set. The accelerometer



(a) Aggressive driving

(b) Normal driving

Figure 5.12: Band-pass filtered lateral acceleration of left (L) and right (R) turns labelled with the bend severity (1–3).

Table 5.4: Dynamic time warping approach results.

Bend severity classification:	
Accuracy	= 83.7%

Aggressive manoeuvre classification:						
Precision	= 64.3%					
Recall	= 37.5%	TP	FP	=	9	5
Specificity	= 95.2%	FN	TN		15	100
Accuracy	= 84.5%					

and gyroscope signals of every label matching turn were summed and averaged. For both signals of each turn, a fixed number of samples centered around the maximum peak of the signal are cut out and summed sample-by-sample. This must be done to ensure a template with a representative curve is obtained. Figure 5.11 shows the automatically generated templates. The length of the gyroscope and accelerometer signal templates are fixed at 200 and 400 samples respectively, but they are the same length in time, namely 10 seconds.

The test data set was used to obtain the results given in Table 5.4. Performance measures and the confusion matrix from which they are calculated are shown. The DTW based bend severity and aggressive manoeuvre classifiers achieved an overall accuracy of 83.7% and 84.5%, respectively. The aggressive manoeuvre classifier obtained a very poor recall of only 37.5%, but an excellent specificity of 95.2%.

More specifically, for the driver labelled as most aggressive from first-hand observation,

Table 5.5: Maximum likelihood classifier results.

Bend severity classification:	
Accuracy	= 91.5%

Aggressive manoeuvre classification:						
Precision	= 77.8%					
Recall	= 58.3%	TP	FP	=	14	4
Specificity	= 96.2%	FN	TN		10	101
Accuracy	= 89.1%					

the classifier achieved a FN and FP rate of 80% and 10.5%, respectively. The high FN rate indicates that the classifier is biased towards classifying a turn as normal rather than aggressive. This is too be expected however, given the fact that only 18% of the data set consists of aggressively labelled turns. Figure 5.12a shows the lateral acceleration of a one minute section where 4 of his aggressive turns occurred. The vehicle's average speed was 85 km/h through this section.

Figure 5.12b shows the lateral acceleration of another driver for the same section of road as in Figure 5.12a. All of the second driver's turns were observationally labelled as normal, and his average speed was 70 km/h through the section. The lateral acceleration never exceeded $0.1g$, whereas with the aggressive driver the acceleration exceeded $0.1g$ for all four turns. The classifier achieved a FN and FP rate of 0% and 5.9%, respectively, for this driver.

With 24 aggressive turns out of 129 in the test set, the aggressive turn labelling heuristic achieved a FN and FP rate of 62.5% and 4.8%, respectively. Although the FN rate is high, a lower FP rate is desirable. It is biased to label a driver as aggressive based on falsely identified aggressive manoeuvres. The heuristic was tuned to obtain the least false positives, at the expense of missing many true positives (TP). Although the sample size was small, it is clear that the classifier's precision and recall is poor and could be improved. The strength of the system is that it can definitely be expanded to recognize other manoeuvres by preparing relevant templates for the same or other axes of the sensors.

5.4.2.2 Maximum likelihood classifier

Table 5.5 shows the results of the ML classifier on the test data set. Performance measures and the confusion matrix are once again shown. The feature set as described in Chapter 4, Section 4.4.2.1, was used to obtain these results. The ML bend severity and aggressive manoeuvre classifiers obtained an accuracy of 91.5% and 89.1%, respectively. That is 7.8% and 5.2% higher than the DTW based classifiers' accuracies. The ML aggressive manoeuvre classifier obtained a recall of 58.3% and specificity of 96.2%. The difference in specificity between the DTW and ML classifier is negligible. The ML classifier had only one less false positive compared to the DTW classifier, out of 129 turns. The ML classifier's recall of 58.3%, however, is comparatively a significant improvement.

Most importantly though, the precision of the ML classifier is substantially better than the DTW classifier's, 77.8% compared to 64.3%, respectively. As previously discussed in Section 5.4.1.2, the $F_{0.5}$ score is an useful performance measure, as it weights precision higher than recall. The $F_{0.5}$ scores of the ML and DTW classifiers are 0.729 and 0.563,

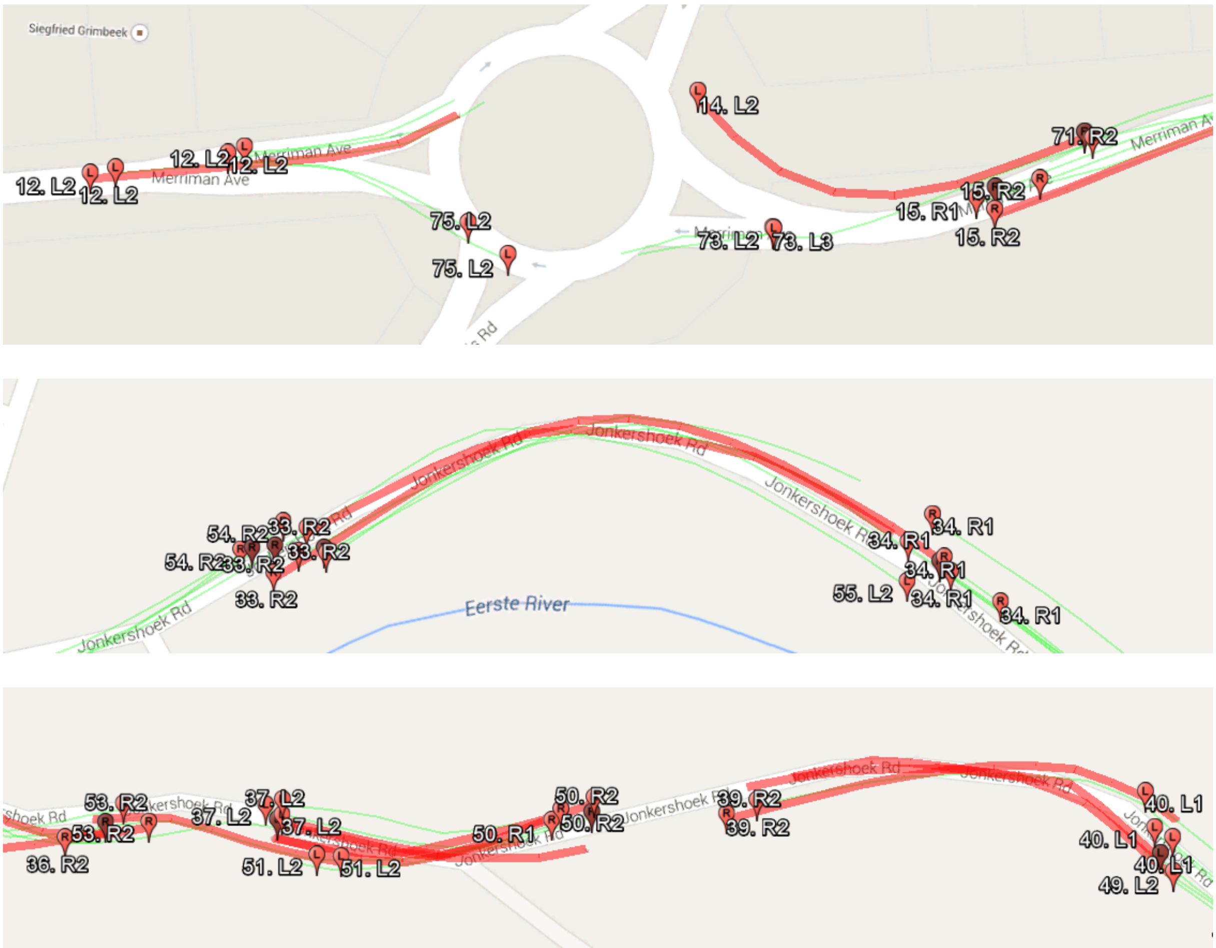


Figure 5.13: The ML classifiers' labelling of the turns in the test data set.

respectively. Considering all the mentioned scores, the ML classifier is clearly better for both bend severity and aggressive manoeuvre classification.

Once again, the results for the most aggressive driver are 40% and 10.5% for the FN and FP rate, respectively. The ML classifier obtains half the FN rate of the DTW classifier, while maintaining the same FP rate, which is a crucial improvement. These results are further evidence of the ML classifier's superiority in classifying aggressive manoeuvres.

An interesting observation is that, although the DTW and ML aggressive manoeuvre classification results are the same for the normal driver, the ML bend severity classifier obtains an accuracy of 94.1% compared to 76.5% of the DTW classifier. This suggests that the DTW bend severity classifier struggles with bends that were taken slower, were the gyroscope signal is stretched out over time with a lower peak.

Figure 5.13 shows a few maps overlaid with the GPS traces of turns included in the test data set. It serves as an example of how feedback can be presented to a driver. Each turn's classifications, as made by the ML classifiers, accompanies the traces. The red pins indicate the starting position of a turn, labelled with the turn's identifying number, an L or R for a left or right turn, and the classified bend severity. The turns with green and red traces are classified as normally and aggressively taken, respectively.

Chapter 6

Conclusion

The work in this thesis is concluded in this chapter. A summary of the work is given, whereafter the conclusions drawn from the test results are discussed, and the hypotheses that were made are evaluated. Lastly, possible applications and recommendations for future work are discussed.

6.1 Summary of work

This thesis focuses on the use of smartphones to monitor a person's driving behaviour. Aggressive driving is one of the major causes of road accidents, and it is therefore important to investigate ways to improve people's driving habits. To this end, a system was developed that specifically detects lateral driving manoeuvres and that classifies them as aggressive or not. Various supervised learning classification algorithms and a dynamic time warping approach was implemented and tested for aggressive manoeuvre classification. The work serves as a framework for developing a smartphone-based system that can detect and classify various aggressive driving manoeuvres.

In Chapter 2, a comprehensive literature survey of the current state of smartphone-based vehicle monitoring systems were presented. Work falling into the categories of road condition monitoring, vehicle telematics, driver behaviour monitoring and collaborative driving were discussed. Particular focus was placed on a few examples of smartphone-based driver behaviour monitoring systems. The best existing solution was implemented, and then compared to the most effective supervised learning classification algorithm that was tested. A design overview of the manoeuvre detection and classification system was given in Chapter 3, as well as a discussion of the aggressive driving model that was used. The model is based on the angle of a turn, the lateral force exerted on the vehicle and its speed through the turn. Chapter 4 described the system design in further detail. The data acquisition system and collection process was discussed, as well as the two implemented aggressive driving recognition classifiers. Lastly, in Chapter 5 the tests and results of the implemented manoeuvre detection and classification algorithms were presented and thoroughly discussed. The performance of each classifier was tested using the same data set and a quantitative comparison was made between them. In the next section the results are also used to confirm the hypotheses made in Chapter 1.

6.2 Conclusions

The goal of this study was to determine if aggressive and normal lateral driving manoeuvres could be successfully detected and classified by a smartphone, using data from its embedded accelerometer, gyroscope and GPS. We also wanted to determine if a supervised learning classifier could outperform a dynamic time warping classification approach in aggressive driving recognition.

The lateral driving manoeuvre detection algorithm that is described in Section 4.3 is evaluated in Section 5.3.3. It successfully detected 95% of the turns in the test data set using only gyroscope data that was sampled at 20 Hz on the smartphone. The maximum likelihood and dynamic time warping classifiers, as described in Section 4.4, managed to correctly label the bend severity of 92% and 84% of the turns, respectively, as shown in Section 5.4.2. Both classifiers only used the gyroscope signals of the turns. These results confirms Hypotheses 1 and 2.

Hypothesis 1:

> Smartphone sensors can be sampled and processed fast enough to detect and classify driving manoeuvres.

Hypothesis 2:

> Rotation rate measurements from a gyroscope are sufficient to detect the start and end of a lateral driving manoeuvre, and to classify the severity of a road bend when driving through it.

The implemented Kalman filter described in Section 4.3.1, is evaluated in Section 5.3.1. It is shown to work well smoothing the accelerometer and gyroscope signal and effectively removing noise. The finite impulse response filter that is designed as in Section 4.3.2, is evaluated in Section 5.3.2. The high-pass filter succeeds in effectively removing the gravitational force vector from the accelerometer signal. These results confirms Hypothesis 3.

Hypothesis 3.

> Acceleration and rotation rate measurements can be successfully filtered to remove unwanted noise and offsets.

The two aggressive driving manoeuvre classification methods are described in Section 4.4. Both the maximum likelihood and dynamic time warping classifiers use the classified bend severity, vehicle speed, and data from the accelerometer to identify aggressive turns. The performance results of the two aggressive manoeuvre classifiers are presented in Section 5.4.2. The ML and DTW classifiers achieved an accuracy of 89% and 85%, respectively, in labelling turns as aggressive or normal, which confirms Hypothesis 4. Furthermore, the $F_{0.5}$ scores of the ML and DTW classifiers are 0.73 and 0.56, respectively. The $F_{0.5}$ scores provides the best performance comparison of the aggressive manoeuvre classifiers. Hence, the ML classifier is clearly better, and since it is a supervised learning classifier, it proves Hypothesis 5.

Hypothesis 4.

> Aggressive lateral driving manoeuvres can be identified from acceleration, speed and bend severity information.

Hypothesis 5.

> Supervised learning classification algorithms can be successfully applied to aggressive driving recognition, and they can perform better than the existing dynamic time warping approach as found in literature.

6.3 Future work

The aim of this study was to provide a framework for developing a smartphone-based driving manoeuvre recognition system. Valuable insight was gained in potential implementations and the performance that can be attained. Various recommendations can be made for future work and potential applications.

The logical next step is to expand this system to detect longitudinal manoeuvres, such as braking and forward acceleration. Then the aggressive driving model can be improved and used to classify aggressive braking and acceleration. The system can also be easily upgraded to classify other driving manoeuvres such as lane changing and swerving, if adequate data can be collected. Performing further tests with a larger data set of various drivers, vehicles and manoeuvres is recommended, to refine the system's performance. Only thereafter would it be sensible to fully implement the improved system on a smartphone, because the classifiers must already be trained in order to operate in a real-time application. When developing the system into a smartphone application, it is strongly recommended to implement and thoroughly test a phone-orientation calibration process. It is needed for the system to work reliably, without requiring the user to keep the smartphone in a fixed orientation inside their vehicle.

Overall, the work in this thesis accomplished its research objectives. A lateral driving manoeuvre detection and recognition system was successfully developed, and its potential to be implemented on a smartphone was substantiated. The suitability of supervised learning classifiers for classifying aggressive driving, in comparison to dynamic time warping classification, was demonstrated and used to validate our aggressive driving model. Conceivably, this work can be employed in the future to develop an holistic smartphone-based driver behaviour monitoring system, which can be easily deployed on a large scale to help make the public drive better. This would make our roads safer, reducing the occurrence of road accidents and fatalities.

References

[1] R. Xu, S. Zhou, and W. J. Li, "MEMS accelerometer based nonspecific-user hand gesture recognition," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1166–1173, 2012.

[2] A. Akl, C. Feng, and S. Valaee, "A novel accelerometer-based gesture recognition system," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6197–6205, 2011.

[3] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, p. 13, 2010.

[4] "Global status report on road safety: Time for action," Geneva, World Health Organization, 2009, [ONLINE] Available: http://whqlibdoc.who.int/publications/2009/9789241563840_eng.pdf. [Accessed 28 August 2014].

[5] "Aggressive driving: Research update," AAA Foundation for Traffic Safety, April 2009, [ONLINE] Available: http://www.aaafoundation.org/pdf/AggressiveDrivingResearchUpdate2009.pdf. [Accessed 28 August 2014].

[6] "Road safety annual report 2013," Paris, International Transport Forum, OECD, May 2013, [ONLINE] Available: http://www.internationaltransportforum.org/Pub/pdf/13IrtadReport.pdf. [Accessed 17 September 2014].

[7] "DriveCam programs," Lytx, 2014, [ONLINE] Available at: http://www.lytx.com/our-solutions/drivecam-programs. [Accessed 28 August 2014].

[8] "How it works," AutoHabits, 2012, [ONLINE] Available at: http://autohabits.com/how-it-works. [Accessed 28 August 2014].

[9] "Fleet safety," FleetMind, 2014, [ONLINE] Available at: http://www.fleetmind.com/fleet-management-products/fleet-safety. [Accessed 28 August 2014].

[10] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *14th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2011, pp. 1609–1615.

[11] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt, "WreckWatch: automatic traffic accident detection and notification with smartphones," *Mobile Networks and Applications*, vol. 16, no. 3, pp. 285–303, 2011.

[12] M. J. Booysen, J. Gilmore, S. Zeadally, and G. J. Van Rooyen, "Machine-to-machine (M2M) communications in vehicular networks," *KSII Transactions on Internet and Information Systems*, vol. 6, no. 2, pp. 529–546, 2012.

[13] D. Trossen and D. Pavel, "NORS: An open source platform to facilitate participatory sensing with mobile phones," in *Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (MobiQuitous).* IEEE, 2007, pp. 1–8.

[14] E. Koukoumidis, M. Martonosi, and L. S. Peh, "Leveraging smartphone cameras for collaborative road advisories," *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 707–723, 2012.

[15] M. V. Ramesh, A. Jacob, and R. Aryadevi, "Participatory sensing platform to revive communication network in post-disaster scenario," in *21st Annual Wireless and Optical Communications Conference (WOCC).* IEEE, 2012, pp. 118–122.

[16] S. S. Kanhere, "Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces," in *Distributed Computing and Internet Technology.* Springer, 2013, pp. 19–26.

[17] C. Costa, C. Laoudias, D. Zeinalipour Yazti, and D. Gunopulos, "SmartTrace: Finding similar trajectories in smartphone networks without disclosing the traces," in *27th International Conference on Data Engineering (ICDE).* IEEE, 2011, pp. 1288–1291.

[18] B. Predic, Z. Yan, J. Eberle, D. Stojanovic, and K. Aberer, "ExposureSense: Integrating daily activities with air quality using mobile participatory sensing," in *11th International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops).* IEEE, 2013, pp. 303–305.

[19] T. Ichimura and S. Kamada, "A generation method of filtering rules of Twitter via smartphone based participatory sensing system for tourist by interactive GH-SOM and C4.5," in *2nd International Conference on Systems, Man, and Cybernetics (SMC).* IEEE, 2012, pp. 110–115.

[20] M. Liu, "A study of mobile sensing using smartphones," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[21] N. De Caro, W. Colitti, K. Steenhaut, G. Mangino, and G. Reali, "Comparison of two lightweight protocols for smartphone-based sensing," in *Communications and Vehicular Technology in the Benelux (SCVT), 2013 IEEE 20th Symposium on*, Nov 2013, pp. 1–6.

[22] C. Campolo, A. Iera, A. Molinaro, S. Y. Paratore, and G. Ruggeri, "SMaRTCaR: An integrated smartphone-based platform to support traffic management applications," in *First International Workshop on Vehicular Traffic Management for Smart Cities (VTM).* IEEE, 2012, pp. 1–6.

[23] O. Briante, C. Campolo, A. Iera, A. Molinaro, S. Y. Paratore, G. Ruggeri, and M. J. Booysen, "ITSPhone: An integrated platform for participatory ITS data collection and opportunistic transfer," *IEEE Infocom 2013*, pp. 1420–1421, 2013.

[24] K. Perera and D. Dias, "An intelligent driver guidance tool using location based services," in *1st International Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM).* IEEE, 2011, pp. 246–251.

[25] K. Ali, D. Al Yaseen, A. Ejaz, T. Javed, and H. S. Hassanein, "CrowdITS: Crowdsourcing in Intelligent Transportation Systems," in *Wireless Communications and Networking Conference (WCNC)*. IEEE, 2012, pp. 3307–3311.

[26] X. Zhang, H. Gong, Z. Xu, J. Tang, and B. Liu, "Jam eyes: A traffic jam awareness and observation system using mobile phones," *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.

[27] E. Koukoumidis, L. S. Peh, and M. R. Martonosi, "Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 2011, pp. 127–140.

[28] M. Lan, M. Rofouei, S. Soatto, and M. Sarrafzadeh, "SmartLDWS: A robust and scalable lane departure warning system for the smartphones," in *12th International Conference on Intelligent Transportation Systems (ITSC'09)*. IEEE, 2009, pp. 1–6.

[29] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *Intelligent Vehicles Symposium (IV)*. IEEE, 2012, pp. 234–239.

[30] J. Dai, J. Teng, X. Bai, Z. Shen, and D. Xuan, "Mobile phone based drunk driving detection," in *4th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*. IEEE, 2010, pp. 1–8.

[31] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, and M. C. González, "Safe driving using mobile phones," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1462–1468, 2012.

[32] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM conference on Embedded Network Sensor Systems*. ACM, 2008, pp. 323–336.

[33] A. Ghose, P. Biswas, C. Bhaumik, M. Sharma, A. Pal, and A. Jha, "Road condition monitoring and alert application: Using in-vehicle smartphone as internet-connected sensor," in *10th International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2012, pp. 489–491.

[34] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using android smartphones with accelerometers," in *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*. IEEE, 2011, pp. 1–6.

[35] Y. Yang, B. Chen, L. Su, and D. Qin, "Research and development of hybrid electric vehicles can-bus data monitor and diagnostic system through obd-ii and android-based smartphones," *Advances in Mechanical Engineering*, vol. 2013, 2013.

[36] J. Zaldivar, C. T. Calafate, J. C. Cano, and P. Manzoni, "Providing accident detection in vehicular networks through OBD-II devices and Android-based smartphones," in *36th Conference on Local Computer Networks (LCN)*. IEEE, 2011, pp. 813–819.

[37] C. Thompson, J. White, B. Dougherty, A. Albright, and D. C. Schmidt, "Using smartphones and wireless mobile sensor networks to detect car accidents and provide situational awareness to emergency responders," in *ICST Conference, June*, 2010.

[38] V. C. Magaña and M. M. Organero, "Artemisa: Using an Android device as an eco-driving assistant," *Cyber Journals: Multidisciplinary Journals in Science and Technology: Journal of Selected Areas in Mechatronics (JMTC)*, 2011.

[39] R. Araujo, A. Igreja, R. de Castro, and R. Araujo, "Driving coach: A smartphone application to evaluate driving efficient patterns," in *Intelligent Vehicles Symposium (IV)*. IEEE, 2012, pp. 1005–1010.

[40] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*. ACM, 2008, pp. 29–39.

[41] M. Müller, *Information retrieval for music and motion*. Springer, 2007, vol. 2.

[42] M. J. Booysen, S. J. Andersen, and A. S. Zeeman, "Informal public transport in Sub-Saharan Africa as a vessel for novel intelligent transport systems," in *16th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2013.

[43] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[44] R. Muscillo, S. Conforto, M. Schmid, P. Caselli, and T. D'Alessio, "Classification of motor activities through derivative dynamic time warping applied on accelerometer data," in *Engineering in Medicine and Biology Society (EMBS), 29th Annual International Conference of the IEEE*. IEEE, 2007, pp. 4930–4933.

[45] "Speed factor," Arrive Alive South Africa, 2014, [ONLINE] Available at: https://www.arrivealive.co.za/Speed-Factor. [Accessed 28 November 2014].

[46] B. Phillips and B. Hardy, *Android Programming: The Big Nerd Ranch Guide*. Pearson Education, 2013.

[47] "MinIMU-9 v2 gyro, accelerometer, and compass," Pololu Robotics & Electronics, 2014, [ONLINE] Available: http://www.pololu.com/product/1268/. [Accessed 7 October 2014].

[48] "ITEAD GPS shield," ITEAD Intelligent Systems, 2012, [ONLINE] Available: http://imall.iteadstudio.com/im120417017.html. [Accessed 7 October 2014].

[49] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[50] G. Welch and G. Bishop, "An introduction to the kalman filter," 1995.

[51] A. V. Oppenheim, R. W. Schafer, J. R. Buck *et al.*, *Discrete-time signal processing*. Prentice-hall Englewood Cliffs, 1989, vol. 2.

[52] D. Albanese, R. Visintainer, S. Merler, S. Riccadonna, G. Jurman, and C. Furlanello, "mlpy: Machine learning python," 2012.

[53] S. J. Prince, *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.