# Driving Manoeuvre Recognition using Mobile Sensors

by

Christopher Woo

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2016

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

# Abstract

Automobiles are integral in today's society as they are used for transportation, commerce, and public services. The ubiquity of automotive transportation creates a demand for active safety technologies for the consumer. Recently, the widespread use and improved sensing and computing capabilities of mobile platforms have enabled the development of systems that can measure, detect, and analyze driver behaviour. Most systems performing driver behaviour analysis depend on recognizing driver manoeuvres. Improved accuracy in manoeuvre detection has the potential to improve driving safety, through applications such as monitoring for insurance, the detection of aggressive, distracted or fatigued driving, and for new driver training. This thesis develops algorithms for estimating vehicle kinematics and recognizing driver manoeuvres using a smartphone device.

A kinematic model of the car is first introduced to express the vehicle's position and orientation. An Extended Kalman Filter (EKF) is developed to estimate the vehicle's positions, velocities, and accelerations using mobile measurements from inertial measurement units and the Global Positioning System (GPS). The approach is tested in simulation and validated on trip data using an On-board Diagnostic (OBD) device as the ground truth. The 2D state estimator is demonstrated to be an effective filter for measurement noise.

Manoeuvre recognition is then formulated as a time-series classification problem. To account for an arbitrary orientation of the mobile device with respect to the vehicle, a novel method is proposed to estimate the phone's rotation matrix relative to the car using PCA on the gyroscope signal. Experimental results demonstrate that e Principal Component (PC) corresponds to a frame axis in the vehicle reference frame, so that the PCA projection matrix can be used to align the mobile device measurement data to the vehicle frame.

A major impediment to classifier-manoeuvre recognition is the need for training data, specifically collecting enough data and generating an accurate ground truth. To address this problem, a novel training process is proposed to train the classifier using only simulation data. Training on simulation data bypasses these two issues as data can be cheaply generated and the ground truth is known. In this thesis, a driving simulator is developed using a Markov Decision Process (MDP) to generate simulated data for classifier training. Following training data generation, feature selection is performed using simple features such as velocity and angular velocity. A manoeuvre segmentation classifier is trained using multi-class SVMs. Validation was performed using data collected from driving sessions. A grid search was employed for parameter tuning. The classifier was found to have a 0.8158 average precision rate and a 0.8279 average recall rate across all manoeuvres resulting in an average $F_1$ score of 0.8194 on the dataset.

## Dedications

To my parents who have unconditionally supported me.

To my brother whom is always full of surprises.

To my partners who have influenced me.

To my supervisor who always seems to run that extra mile.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Automobiles are indispensable in today's society and are ubiquitous in personal transportation, commercial activities, and public services. While they continue to prefer this convenient form of transportation, safety and environmental factors remain a large concern for consumers, businesses, and governments.

According to [11], Canada's road fatality rates are the 4th highest in the world with 7.18 fatalities per 1 billion kilometers traveled by a vehicle. The annual social costs of automobile accidents in terms of loss of life, medical treatment, lost productivity, and property damage are measured in tens of billions of dollars [11].

Although vehicle crash rates are declining, the number of driving related deaths is approximately 1,823 per year as of 2012 [12]. Collision statistics reveal that 27 % of fatalities and 19 % of serious injuries involve speeding [11], a subset of aggressive driving.

Most passive safety systems have been designed to mitigate the effects of accidents after they occur. Increasing adoption of passive vehicle safety features such as seat belts [13], airbags [14], backup cameras, and adaptive headlights have been extremely important in lowering the vehicle-related death rate. Recent developments in active safety features such as side view assists, parking assists, backup sensors, anti-lock brake, and adaptive cruise control systems aim to further improve vehicle safety. These new systems show an increasing adoption trend in modern vehicles [15].

Systems that can detect activities such as driving under the influence, drowsy driving, and aggressive driving would help identify drivers more likely to get into an accident before it happens. Transport Canada defines aggressive driving as speeding, running red lights, tailgating, weaving in and out of traffic, and failing to yield right of way, among other behaviours [11]. Detecting these features can be used to help determine aggressive driver behaviour and aid to improve safety on the road.

Driver assistance systems and the analysis of different driving styles require a more precise understanding of driver behaviours to be effective. Driver behaviour analysis has recently received much attention due to the recent availability of car telematic devices on vehicles [16], [17]. These telematic devices allow for car kinematics to be recorded and analyzed at a later time. Telematic devices that connect to the car through the onboard diagnostic (talk) port have been demonstrated to collect useful information relating to automobile control (e.g. steering, accelerator position, and brake position) [17].

The recent adoption of mobile devices [18] has the potential to facilitate driver behaviour analysis without requiring access to vehicle integrated sensors. The ubiquity of mobile devices lowers the overhead costs in designing, implementing, and deploying a sustainable large-scale smartphone-based measurement system for analyzing driver behaviour. Monitoring driving patterns on a mobile platform is facilitated due to device mobility, network connectivity, cloud storage availability, and processing algorithms to aggregate collected data [19]. The designed system can be used on a mobile device to replace a telematic device allowing easy installation for the average consumer and lowering distribution.

Driver behaviour measurement and analysis systems have a number of important applications. Usage-based Insurance is a new and developing option provided to insurance customers. UBI allows for vehicle insurance to be dependent on the driver's vehicle usage measured in distance, behaviours, and locations [20]. With premiums representing driver risks more accurately, safer drivers can benefit from reduced premiums while insurance companies have lower risk in payouts. These benefits are propelling the insurance market to quickly expand the availability of telematics-based UBI programs. This new policy benefits society as a whole by deterring dangerous drivers from owning a vehicle and driving on the road.

## 1.1 Thesis Contributions

**Online vehicle state estimation using smartphone sensors:** Modelling the car is fundamental in understanding kinematic constraints for driver manoeuvre analysis and motion simulation. This thesis develops an online vehicle state estimator using sensor fusion to estimate the position and orientation of a vehicle. A state estimator with overlapping sensor information combining multiple measurements from low cost sensors are used to improve estimate accuracy and reject measurement noise. This thesis proposes a technique to model the car as a Two Wheel Robot using a position and orientation representation in 2D space. The model of the vehicle has two driving wheels and the angular velocities of the two wheels are independently controlled. A motion update and observation model are formulated using this model to develop the state estimator.

**Phone rotation matrix estimation:** Measurements obtained by a mobile device within the car cannot be assumed to be located with a fixed and known orientation relative to the vehicle. This thesis proposes a novel technique to extract the rotation matrix between the mobile device and the vehicle using Principal Component Analysis. The method is applied to the 3-axis gyroscope signal to calculate the directions of maximum variance in the phone frame. The coefficient matrix strongly correlates to the rotation matrix transformation from the phone to the car reference frame. By applying the Principal Component vectors in the coefficient matrix, each axis signal can be extracted in the car coordinate frame using the PCA coefficient matrix. For the detection of turning manoeuvres, this technique was used to extract the angular velocity about the yaw axis.

**Vehicle motion model simulator for classifier training:** For classifier-based approaches, a major impediment is the collection of training data for the classifier [62] [10]. To alleviate the need for cumbersome data collection and labelling, a method is proposed to use simulation data for classifier training. This thesis proposes a Markov Decision Process to emulate driver behaviour so that simulated data is generated for classifier training. Furthermore, a carrot and velocity controller are also proposed in this thesis to aid in the vehicle control during simulation using the Two Wheel Robot model. The development of tools to recognize driving manoeuvres can provide tools for driver monitoring and training, improving safety and fuel efficiency.

**Driver manoeuvre segmentation as time-series classification:** Methods to analyze driver behaviour use manoeuvre detection techniques. Since smartphones sensor data are collected as a time-series, a classifier trained offline is used to label manoeuvres in realtime. This thesis proposes that a multi-class Support Vector Machine using a Radial Basis function kernel for manoeuvre classification. Using simulated data generated in the thesis with the state estimator applied, feature dimensionality reduction is performed to reduce the number of features for classifier training. The classifier is trained offline using the reduced simulated features based on linear and angular velocity. The rotation matrix estimation is then applied to extract the angular velocity in real trip data and linear velocity is pulled from the GPS sensor. The classifier is applied to trip data to label driver manoeuvres.

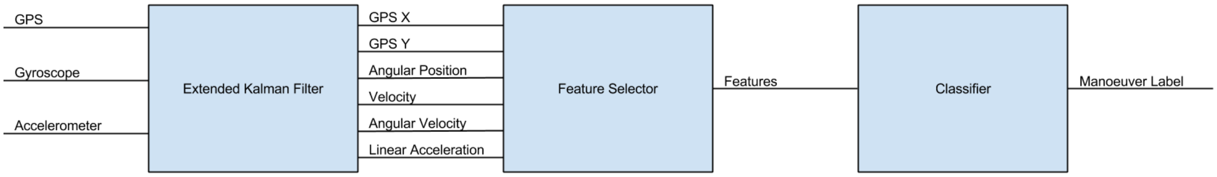


Figure 1.1: System Architecture Block Diagram

**Validation using a real dataset:** Proposed approaches in this thesis are validated using simulated data modelling data collected in real datasets and real trials with naturalistic driving. Based on the performance, the validation shows that classifiers trained on simulated data is able to achieve good results. These results are comparable with those presented in the reviewed literature [96] [38], which use non-simulation data to train their classifiers.

## 1.2 System Architecture

The proposed system has the architecture presented in Figure 1.1. Sensor measurements are obtained from a Mobile Application (MA) running on a mobile device and measurements from the OBD device attached to the car. The GPS, gyroscope, and accelerometer are used to determine the kinematics of the car. An EKF can be designed to estimate the pose and velocity of the car. With this information, feature selection is performed to extract useful features to feed into the classifier. A trained classifier uses these features to label vehicle manoeuvres.

## 1.3 Thesis Outline

Chapter 2 provides a review of related work focused on analyzing driving manoeuvres. Different sensors used for measurement are first presented. State of the art estimation algorithms are discussed. Classification methods used in the thesis are reviewed. Finally, different driver manoeuvres and driver behaviour analysis techniques from the literature are reviewed.

Chapter 3 describes the state estimation methodology used to model the vehicle. Measurement noise model parameters are first investigated using sensor trials. These parameters are used to develop a state estimator using the Extended Kalman Filter. Estimation performance is presented for simulated trials. A carrot and velocity controller are developed for simulated vehicle control. Finally, a Markov Decision Process (MDP) is proposed to generate simulated data for classifier training.

Chapter 4 describes how a classifier is developed for time-series manoeuvre recognition. First, a novel method is described to estimate the phone's rotation matrix relative to the car. The feature selection process is then described to construct the feature vector for the classifier's input. The training process is then highlighted with methods to optimize parameter selection. The data collection process to validate the approach is described and classification results are presented.

Chapter 5 provides a summary of the methods and findings in this thesis. This chapter presents a summary of conclusions and outlines the direction for future work.

# Chapter 2

# Related Work

A great deal of research has been conducted in the area of vehicle kinematic estimation and driver behaviour analysis. This chapter begins with a brief review of the measurement sensors commonly used and provides an overview of the state of the art in measurement sensors, vehicular state estimation, classifiers, manoeuvre segmentation, and driving style recognition.

## 2.1 Measurement Sensors

A variety of sensors have been used for vehicle state and driver behaviour estimation. They can be divided into 3 broad classes:

1. Absolute Position Sensors
2. Inertial Measurement Units
3. Cameras

A modern smartphone has an array of available sensors accessible to mobile applications. The relevant sensors available on a modern smartphone's platform will be described in this section.

### 2.1.1 Absolute Position Sensors

Sensors that measure position in an inertial reference frame are known as absolute position sensors. Absolute position signals can be differentiated to calculate velocity and acceleration. Global Navigational Satellite Systems (GNSS) are publicly accessible sources for obtaining an absolute positioning in vehicle telematics.

GNSS are commonly used as the primary sensor for vehicle tracking applications. Some practical applications using absolute position sensors include Usage Based Insurance (UBI) [21] [20] [24] [25], robust localization using secondary filtering methods [3] [26] [27], road safety mobile apps [28] [17] [19] [29], and road map generation [30]. The GNSS measurement rate is device dependent, but the industry standard, especially in modern smartphones, is 1 Hz.

The Global Positioning System (GPS) is a satellite based positioning system developed by the United States Department of Defense consisting of a constellation of 24 satellites and their ground stations. It can provide users with 3D position, velocity and time. With the removal of Selective Availability (SA) [31] and international efforts such as the Russian GLONASS and Chinese BeiDou, GNSS technology is continuing to improve positioning capabilities for commercial use [32]. Ongoing efforts such as the Japanese QZSS, Indian IRNSS, Chinese COMPASS, and European Galileo aim to extend this functionality worldwide. GNSS accuracy is further improved with the use of IMU sensors with a Kalman Filter (KF) [33], Extended Kalman Filter (EKF) [34], or Unscented Kalman Filter (UKF) [35].

For standard GPS, the accuracy is typically within 10 meters [36]. It is used for applications that only require an approximate position such as car navigation systems (eg. vehicle telematics). Data received by GPS receivers contain information such as position, time, satellite information, and dilution of precision (DOP) values. By using the Doppler effect of known GPS frequencies, velocity is also estimated through post processing.

GPS accuracy and availability depend on many factors such as:

1. Signal arrival errors

2. Satellite clock errors

3. Atmospheric effects including Ionospheric and Tropospheric errors

4. Tall obstacles around GPS antenna causing signal obstruction

5. Multipath algorithm calculation errors due to signal reflection, obstacles, weather, etc.

6. Satellite ephemeris distribution and available satellites in view for localization, at least 4 are required

Dujon et al. used a GPS system for wildlife tracking called Fastloc GPS, and performed extensive testing in fixed and moving trials [1]. In their study, 257 tags of assorted models were deployed and location accuracy and how it impacted speed and heading calculations was investigated. Fixed trials including 45,157 GPS fixes were collected with the number of satellites ranging between 4 and 11. These trials showed that the use of more satellites resulted in lower position errors.

Differential GPS (DGPS) can be used for higher accuracy data collection. DGPS was designed to correct bias errors of the receiver using a known position to bypass SA. This is accomplished by using a base station with a known location and GPS pseudoranges to compute corrections for each satellite signal. DGPS can improve measurement accuracy [37], to within 1 m [36].

### 2.1.2 Inertial Measurement Units

An Inertial Measurement Unit (IMU) is a measurement device that reports a body's acceleration, angular velocity, and possibly the surrounding magnetic field. Typically, this is done using a combination of accelerometers, gyroscopes, and magnetometers. An Inertial Navigation System (INS) can use IMUs to provide state estimates through an integrative system model.

IMUs are used in a range of applications including road safety mobile applications [10] [28], driving behaviour recognition [16] [38] [39] [29] [17] [40], vehicle telematics [41] [5], and human motion tracking [42] [43].

IMU sensors have a high sampling rate when compared to GPS. For example, a common sensor used in the human motion measurement, mobile robotics, and driving behaviour analysis, the Xsens MTi IMU sensor has been sampled at 100 Hz [8] [44] [45]. Smartphone measurement rates of 100 Hz are also commonly reported [46] [19], but may have a negative impact on the battery life of the mobile device. In the field of telematics sample rates of 50 Hz [47] and 100 Hz [48] [49] have been used. Furthermore, 100 Hz has been used as an industry standard in applications such as vehicle modelling [50].

Table 2.1: Tabulated MTi IMU Errors from [7]

| | accelerometers | gyroscopes | magnetometers |
|---|---|---|---|
| Axes | 3 | 3 | 3 |
| Full Scale (FS) | ±50 $m/s^2$ | ±300 $deg/s$ | ±750 mGauss |
| Linearity | 0.2% of FS | 0.1% of FS | 0.2% of FS |
| Bias stability | 0.02 $m/s^2$ | 1 $deg/s$ | 0.1 mGauss |
| Bandwidth | 30 Hz | 40 Hz | 10 Hz |
| Max update rate | 512 Hz | 512 Hz | 512 Hz |

Table 2.2: Tabulated Xsens Fusion Algorithm Rotation Matrix Errors from [7]

| | |
|---|---|
| Static accuracy (roll/pitch) | <0.5 deg |
| Static accuracy (heading)[1] | <1 deg |
| Dynamic accuracy | 2 deg RMS |
| Angular resolution | 0.05 deg |

[1] in homogeneous magnetic environment

Higher sampling rates have also been reported, e.g., 300 Hz [42] and 512 Hz [7] in pedestrian tracking applications. Custom built IMUs have achieved even higher sampling rates, such as 1000 Hz [9].

IMUs work well when used for short-term integrative distance estimation. Over the course of a short period such as a few seconds, accelerometer integration is acceptable as a short duration distance measuring device [51]. Small errors in the distance outperform the accuracy of GNSS sensors as investigated through a static position test [52].

However, IMUs have significant bias errors and these errors also vary over time [7]. This is typically observed as a random walk for gyroscopes estimating angular position and the integral of a random walk for accelerometers estimating position. Thus, dead reckoning algorithms must handle drift during longer term measurement.

Jiménez et al. investigated IMUs in pedestrian dead reckoning (PDR) algorithms using the MTi IMU[7]. The device contains a 3-axis accelerometer, gyroscope, and magnetometer. These sensors are similar to ones found in smartphones. The noise model for the device was tabulated in Table 2.1 and illustrates typical errors found in a low cost IMU. The IMU device also used a proprietary fusion algorithm to estimate the rotation matrix, similar to ones provided by smartphone operating systems. It was observed that the performance was good as long as the earth magnetic field was not disturbed. These rotation matrix errors are tabulated in Table 2.2.

Tedaldi et al. proposed a calibration procedure to calibrate low-cost embedded IMU

Table 2.3: Summarized Accelerometer Errors Between the Measured Gravity vector ($u_g$) and Computed Gravity Vector ($u_a$) from [9]

| | Average Error (mg) | Max Observed Error (mg) |
|---|---|---|
| No Calibration | 20.1 (1.15°) | 44.9 (2.57°) |
| With Calibration | 4 (0.23°) | 28.1 (1.61°) |
| Error Reduction Ratio | 5.02 | 1.60 |

sensors in [8]. The algorithm first calibrates the accelerometer triad. Using gravity vector positions as a reference and numerical integration of angular velocities between 2 consecutive static positions, a new estimate can be obtained using the gravity positions in the new orientation. Gyroscope calibration is obtained by minimizing the errors between these estimates and the gravity references given the calibrated accelerometers. To detect static states, a static detection algorithm using Allan Variance was described. The experimental data uses an Xsens MTi IMU sensor and its uncalibrated data stream.

Significant reductions from the uncalibrated absolute errors were observed by using the calibration methods. The uncalibrated acceleration deviation errors were relatively small and uncalibrated gyroscope deviations were relatively high near sensor saturation. These results demonstrate that uncalibrated or bulk calibrated IMUs, such as low cost sensors found in smartphones, can still measure acceleration and rotation with low errors. Bias errors can be zeroed during a known static state and multiplicative errors can be mitigated when sensors are not near saturation.

Fong et al. proposed a calibration procedure for MEMS IMUs without external equipment [9]. Their tests used a custom built IMU with an accelerometer and gyroscope. The average and maximum magnitudes of the gravity errors were calculated from the ideal 1 g. Before calibration, it was observed that the average and max errors were 1.15° (20.1 mg) and 2.57° (44.9 mg) respectively. After calibration, the average maximum errors were 0.23° (4 mg) and 1.61° (28.1 mg) respectively. Without applying the sensor error model, the divergence can cause an angular error greater than 10 degrees for the calibration set and 2 test sets. After calibration, it was observed that the gyroscopic errors were significantly reduced.

Inexpensive IMU sensors can provide frequent accurate updates for the short term while correcting with GPS in the long term as suggested in [51]. Using INS sensors in conjunction with other absolute position sensors has also been demonstrated to be effective to estimate vehicular position [54]. To counter the effects of integration drift, the drifting INS sensor can be fused with less accurate and intermittent GNSS data to correct for drift in state estimation. Using a KF for linear systems or an EKF for non-linear systems, it is possible

Table 2.4: Summarized Gyroscope Errors Between the Measured Gravity vector ($u_g$) and Computed Gravity Vector ($u_a$) from [9]

| | Calibration Set (mg) | Test Set 1 (mg) | Test Set 2 (mg) |
|---|---|---|---|
| No Calibration | 213.7 (12.30°) | 342 (19.70°) | 236.7 (13.60°) |
| With Calibration | 37.5 (2.15°) | 65.4 (3.75°) | 98.1 (5.62°) |
| Error Reduction Ratio | 5.72 | 5.25 | 2.41 |

to fuse the IMU measurements with the GNSS measurements as described in [34].

### 2.1.3 Cameras

A camera sensor allows for reliable collection of visual data that is not easily observable through other sensors. In telematic applications, the camera may be mounted [55] [56] or in a fixed smartphone device [28]. Some modern smartphones have front and rear cameras. When oriented correctly, they can be used to capture both inward and outward activity corresponding to driver and road environments respectively as demonstrated in [128].

Cameras have been used in intersection analysis [57], driving behaviour analysis [17], manoeuvre detection [58] [5] [59] [28], naturalistic studies [60] [61], driver inattention estimation [62] [63], driving condition estimation [56], and self driving cars [55].

Collected video data can be processed automatically or manually by human segmentation. For instance in [58] and [28], lane change detection is performed automatically. In studies such as [56], road signage and asphalt text is detected automatically. Alternatively, human experts also perform validation of certain events detected by sensors as demonstrated in the 100-Car Naturalistic Data Set [60].

However, privacy is a significant concern when using images or video data. There are two ways of combating these concerns. Camera or video data can be anonymized and coded through identifiers. For instance Boyce and Geller studied males of different ages to evaluate the relationships between age, personality and driving style [63]. Their video recording system discretely activates and deactivates every time a new car is followed to help segment events and is coded to ensure privacy is not compromised. Another example is the 100-car naturalistic study where collected video data blocked faces and lowered resolution for human experts to review [60]. A third active system was developed by Johnson and Trivedi to monitor driving events but does not keep record of any sensor or video data unless a potentially aggressive event is detected [38]. Their system temporarily records all data in five minute segments and deletes them if not flagged.

## 2.2 State Estimation

The sensors and instrumentation described in Section 2.1 are used to estimate the system and environment state. With overlapping measurement information, sensors can also be used together to improve estimates and filter measurement noise.

This section will provide background on models and sensor fusion techniques to be used in the later sections of this thesis. Vehicle models, Kalman Filters, and the Extended Kalman Filter are reviewed.

The standard state space representation in discrete time for an arbitrary system has a state update and observation equations as seen in Equations 2.1 and 2.2. The state update equation is dependent on the system being analyzed whereas the observation equation is dependent on the sensors available.

$$X_{t+1} = F(X_t, U_t) \tag{2.1}$$

$$Y_{t+1} = G(X_t, U_t) \tag{2.2}$$

In this system, $X_t$, $U_t$, and $Y_t$ represent the state, control, and observation vectors respectively at time $t$. The functions $F$ and $G$ represent the state update and observation functions respectively.

### 2.2.1 Vehicle State Estimation

In the literature, the vehicle's position can be represented in 2D using state space representation as illustrated in Figure 2.1. There are 3 common methods to model the vehicle kinematics:

1. Two Wheel Robot Model
2. Bicycle Model
3. Ackermann Robot Model

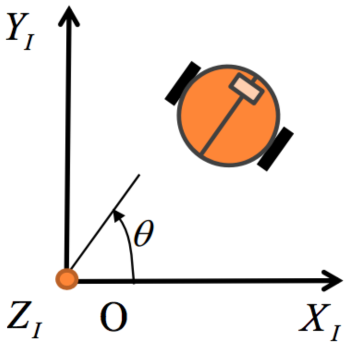


Figure 2.1: Robot World Representation in 2D Space from [2]

### Two Wheel Robot Model

A Two Wheel Robot model is the simplest representation of a position and orientation in 2D space. The model of the vehicle has two driving wheels and the angular velocities of the two wheels are independently controlled. The state and control signal are defined as follows and is further illustrated in Figure 2.2:

$$X_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \\ v_t \\ \omega_t \\ a_t \\ \alpha_t \end{bmatrix} \tag{2.3}$$

$$U_t = \begin{bmatrix} \widetilde{a}_t \\ \widetilde{\alpha}_t \end{bmatrix} \tag{2.4}$$

In these equations, the subscript $t$ represents the variable at time t. For the state vector, $x_t$ and $y_t$ represent the displacement along the x and y-axis respectively in the world reference frame. $\theta_t$ represents the angular position relative to the x-axis. $v_t$ and $\omega_t$ represent the vehicle's linear and angular velocity respectively. $a_t$ and $\alpha_t$ represent the vehicle's linear and angular acceleration respectively. For the control vector, $\widetilde{a}_t$ and $\widetilde{\alpha}_t$ represent the input linear and angular acceleration respectively.

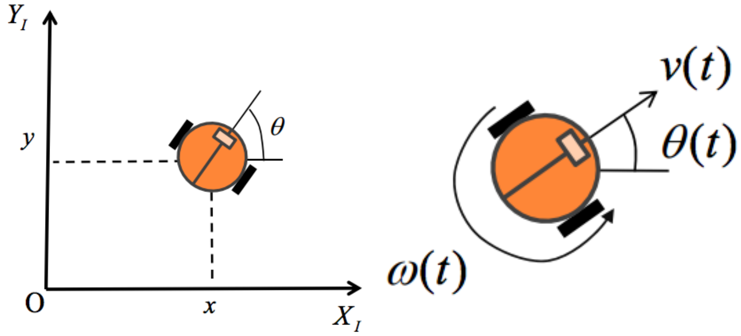


Figure 2.2: Two Wheel Robot Model from [2]

The motion update model is defined as follows:

$$X_t = \begin{bmatrix} x_{t-1} + v_t \cos\theta \Delta t + v_t \cos\theta \frac{(\Delta t)^2}{2} \\ y_{t-1} + v_t \sin\theta \Delta t + v_t \sin\theta \frac{(\Delta t)^2}{2} \\ \theta_{t-1} + \omega_t \Delta t \\ v_{t-1} + a_{t-1} \\ \omega_{t-1} + \alpha_{t-1} \\ \widetilde{\alpha}_t \\ \widetilde{a}_t \end{bmatrix} \tag{2.5}$$

### Bicycle Model

The Bicycle model is a modification to the Two Wheel Robot model. The Two Wheel robot model always rotates about the vehicle's Centre of Mass (CM). However in an actual car, the wheels are decoupled between the front and rear sets of wheels. This is represented with a non-zero length between the front and rear wheels of the bicycle model as illustrated in Figure 2.3.

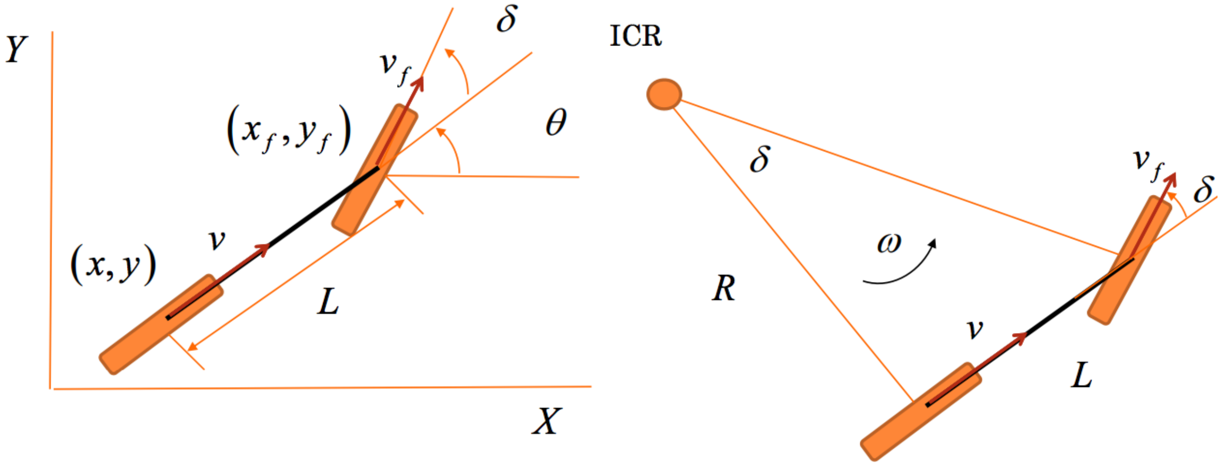


Figure 2.3: Bicycle Model from [2]

The bicycle model state is defined as follows and is further illustrated in Figure 2.3:

$$X_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \\ \delta_t \\ v_t \\ \omega_t \\ a_t \\ \alpha_t \end{bmatrix} \tag{2.6}$$

The bicycle model control signal is defined as follows for the rear wheel:

$$U_t = \begin{bmatrix} \widetilde{a}_t \\ \widetilde{\alpha}_t \\ \widetilde{\delta}_t \end{bmatrix} \tag{2.7}$$

The common state variables and input vectors defined in Section 2.2.1 are redefined in the bicycle model equations for the rear wheel. For the state vector, a new variable $\delta_t$ represents the front wheel angle relative to the vehicle's orientation at time $t$. For the control vector, a new variable $\widetilde{\delta}_t$ represents the input front wheel angle relative to the vehicle's orientation at time $t$.

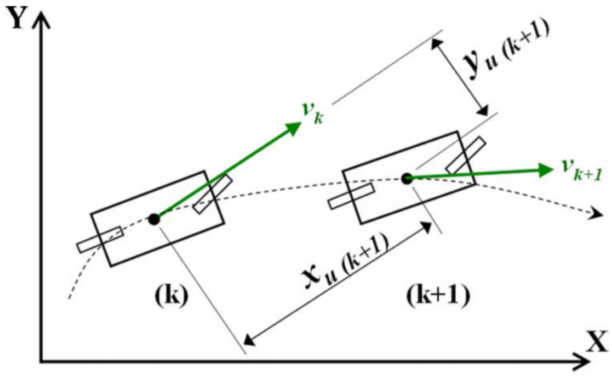


Figure 2.4: Bicycle Motion Model from [3]

The location of the front wheel can be calculated as follows, given a length $L$ representing the length between the wheels:

$$\begin{bmatrix} x_f \\ y_f \end{bmatrix} = \begin{bmatrix} x_t + L\cos\theta_t \\ y_t + L\sin\theta_t \end{bmatrix} \tag{2.8}$$

The motion update model is defined as follows:

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \\ \delta_t \\ v_t \\ \omega_t \\ a_t \\ \alpha_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t\cos\theta_{t-1}\Delta t + v_t\cos\theta\frac{(\Delta t)^2}{2} \\ y_{t-1} + v_t\sin\theta_{t-1}\Delta t + v_t\sin\theta\frac{(\Delta t)^2}{2} \\ \theta_{t-1} + \frac{v_t\tan\delta_t}{L}\Delta t \\ \widetilde{\delta}_t \\ v_{t-1} + a_{t-1}\Delta t \\ \omega_{t-1} + \alpha\Delta t \\ \widetilde{a}_t \\ \widetilde{\alpha}_t \end{bmatrix} \tag{2.9}$$

Rezaei and Sengupta [3] estimate the 2D car position using the Bicycle motion model for collision warning as illustrated in Figure 2.4. Measurement sensors included GPS, steering angle sensor, gyro, and wheel speed sensors as illustrated in Figure 2.5. Sensor fusion was performed using an Extended Kalman Filter. Manoeuvres tested include stops, intersection turns, U-turns, and lane changes, at both low and high speeds.

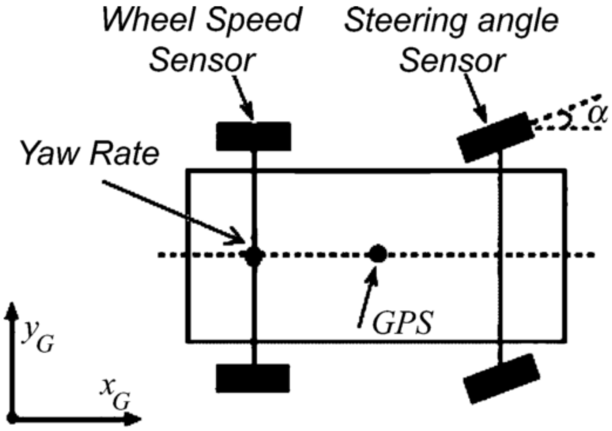


Figure 2.5: Sensor Mount Locations from [3]

### 2.2.2 Phone Reference Frame

Methods proposed in the literature that use smartphone sensors require an additional reference frame for the phone. In these cases, the phone: is already aligned with the car [64] [65] [66], requires a 2D rotation [67], or requires a 3D rotation [68] [10] [69] [70] [53]. The 3D rotation case is the most general form. Euler angles are commonly used to represent 3D rotation. Using the roll, pitch, yaw Euler angles, the rotation angles $\phi$, $\psi$, and $\theta$ correspond to roll, pitch, and yaw respectively. Given a vector $V$ in the reference coordinate frame, the vector in the target coordinate frame is calculated as $V'$

$$V' = R(\theta)R(\psi)R(\phi)V \tag{2.10}$$

$$V' = \begin{bmatrix} v'_x \\ v'_y \\ v'_z \end{bmatrix} \tag{2.11}$$

$$= \begin{bmatrix} \cos(\theta)\cos(\phi) + \sin(\theta)\sin(\psi)\sin(\phi) & -\sin(\theta)\cos(\phi) + \cos(\theta)\sin(\psi)\sin(\phi) & \cos(\psi)\sin(\phi) \\ \sin(\theta)\cos(\psi) & \cos(\theta)\cos(\phi) & -\sin(\psi) \\ -\cos(\theta)\sin(\phi) + \sin(\theta)\sin(\psi)\cos(\phi) & \sin(\theta)\sin(\phi) + \cos(\theta)\sin(\psi)\cos(\phi) & \cos(\psi)\cos(\phi) \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{2.12}$$

Multiple methods have been proposed to estimate these angles. For 3D calibration methods, the pitch and roll angles are calculated using the gravity vector measured by the accelerometer sensor. Different approaches have been proposed to constrain the last angle.

A common method to calculate the yaw angle uses the magnetometer to obtain a reference for the Earth's magnetic field [69] [68] [10]. The transformation rotates the vector with respect to the East North Up (ENU) reference frame. A transformation to the car frame is then achieved using the GPS heading data. A novel method has also been proposed which uses accelerometer measurements after the vehicle moves from a stopped state to calculate the yaw angle [67]. Direction ambiguity is resolved by determining the vehicle direction in the long term. A third method to estimate the yaw angle uses yaw angle estimates and applying a 2 component Gaussian Mixture Model (GMM) to separate the yaw angle from the noise [53] [71]. Alternate rotation representations proposed in the literature involve the use of quaternions to represent the 3D rotation transformation [72].

### 2.2.3 Kalman Filters

The Kalman Filter (KF) is an algorithm that uses a series of measurements over time to optimally estimate the system state [73]. The Kalman filter has numerous applications and is a popular state estimation technique for linear systems. It is an optimal filter for linear systems if the measurement and process noises are assumed to be zero-mean Gaussians.

A discrete linear system can be defined as:

$$X_t = AX_{t-1} + BU_t + q_{t-1} \tag{2.13}$$

$$Y_t = CX_t + r_t \tag{2.14}$$

In this system, $A$ represents the state propagation from the previous timestep. $B$ represents how the state is influenced by an input control signal. $C$ represents the observation transformation given the current state. Since $A$, $B$, and $C$ are linear transformations for a linear system, they can be represented as matrices. $q_t$ and $r_t$ represent the measurement and process noise respectively.

The noise model is defined in Equations 2.15 and 2.16 to satisfy the optimal filter assumption.

$$q_t \sim \mathcal{N}(0, Q_t) \tag{2.15}$$

$$r_t \sim \mathcal{N}(0, R_t) \tag{2.16}$$

In these noise models, $Q_t$ and $R_t$ represent the process and measurement covariance respectively.

The Kalman filter algorithm is separated into 2 steps. First, the prediction step uses the state estimate from the previous time-step to produce a current time-step state estimate *a priori* using the state prediction model. In an update step, the *a priori* prediction is combined with an observation to refine the state estimate to produce the *a posteriori* state estimate.

### Prediction Step

The state estimate is first calculated through the state update equation as shown in Equation 2.17. The estimated prediction covariance is calculated in Equation 2.18.

$$\hat{X}_{t|t-1} = A\hat{X}_{t-1|t-1} + BU_t \tag{2.17}$$

$$P_{t|t-1} = AP_{t-1|t-1}A^T + Q_k \tag{2.18}$$

In these equations, $\hat{X}_{s|t}$ represent the state belief at time $s$ given information at time $t$. On the other hand, $P_{s|t}$ represents the state covariance at time $s$ given information at time $t$.

### Update Step

The subscripts $t$ in the following equations represent the variables at time $t$. The innovation $(Z_t)$ is calculated from the prediction model in Equation 2.19. The innovation covariance $(S_t)$ is calculated in Equation 2.20. The optimal Kalman Gain $(K_t)$ is calculated in Equation 2.22 The updated state estimate is calculated in Equation 2.22. Finally, the updated estimated covariance is calculated in Equation 2.23.

$$Z_t = Y_T - C\hat{X}_{t|t-1} \tag{2.19}$$

$$S_t = CP_{t|t-1}C^T + R_t \tag{2.20}$$

$$K_t = P_{t|t-1}C^T S_t^{-1} \tag{2.21}$$

$$X_{t|t} = X_{t|t-1} + K_t Z_t \tag{2.22}$$

$$P_{t|t} = (I - K_t C)P_{t|t-1} \tag{2.23}$$

When applying the KF on real systems, the process covariance $Q_t$ is typically unknown. As such, it is used in conjunction with the measurement covariance $R_t$ as a tuning parameter to reach desired filter performance criteria.

### 2.2.4 Extended Kalman Filters

Many common systems analyzed in the literature are non-linear and thus the Kalman Filter cannot be directly applied. The Extended Kalman Filter (EKF) was developed by linearizing the Kalman Filter so that it could be applied to non-linear systems [73]. An arbitrary non-linear system has the form:

$$X_t = F(X_{t-1}, U_t) + q_{t-1} \tag{2.24}$$

$$Y_t = G(X_t) + r_t \tag{2.25}$$

Using first order Taylor series expansion, the system can be linearized as follows:

$$X_t = \widetilde{X}_t + \frac{\partial F}{\partial X}(X_{t-1} - \widetilde{X}_{t-1}) + q_{t-1} \tag{2.26}$$

$$Y_t = \widetilde{Y}_t + \frac{\partial G}{\partial X}(X_t - \widetilde{X}_t) + r_t \tag{2.27}$$

Here, $\widetilde{X}_t = F(\widetilde{X}_{t-1}, 0)$ and $\widetilde{Y}_t = G(\widetilde{X}_t, 0)$ are the noiseless state and measurement estimates. The updated filter equations are as follows:

**Prediction Step**

$$\hat{X}_{t|t-1} = F(\hat{X}_{t-1|t-1}, U_t) \tag{2.28}$$

$$P_{t|t-1} = A_t P_{t|t-1} A_t^T + Q_t \tag{2.29}$$

**Update Step**

$$Z_t = Y_t - G(\hat{X}_{t|t-1}) \tag{2.30}$$

$$S_t = C_t P_{k|k-1} C_t^T + R_k \tag{2.31}$$

$$K_t = P_{t|t-1} C_t^T S_t^{-1} \tag{2.32}$$

$$X_{t|t} = X_{t|t-1} + K_t Z_t \tag{2.33}$$

$$P_{t|t} = (I - K_t C_t) P_{t|t-1} \tag{2.34}$$

Here, the Jacobians are defined as follows

$$A_{t-1} = \left.\frac{\partial F}{\partial X}\right|_{\hat{X}_{t-1|t-1},U_t} \tag{2.35}$$

$$C_t = \left.\frac{\partial G}{\partial X}\right|_{\hat{X}_{t|t-1}} \tag{2.36}$$

Since the system is non-linear, the linear assumption is broken and the filter is not optimal. However in most applications, the performance of the EKF is still acceptable when proper tuning parameters for the process and measurement covariances are used.

Suliman et al. compare the localization accuracy between a Kalman Filter (KF) and an Extended Kalman Filter (EKF) for an autonomous mobile vehicle that uses Ackermann steering in [34]. Near the beginning of the simulation, KF errors were similar to the EKF errors. However as the simulation progressed, the EKF retained smaller errors than the KF.

Dorobantu and Zebhauser investigate the accuracy of an INS system of a car using a Two Wheel Robot model for the EKF [74]. The sensor platform uses a DGPS, accelerometer, and gyroscope sensor. The system performance is evaluated when GPS sample rates are 0.25 Hz and 1 Hz. The 1 Hz estimate is preferred as the 0.25 Hz estimate is less accurate and would not be able to distinguish between lanes.

### 2.2.5 Summary

This section highlights modelling and state estimation techniques. The reviewed literature describes methods for vehicle state estimation using combinations of absolute position and inertial measurements. However, these techniques depend on permanently fixed sensors or restrict the vehicle model to 2D. Expecting drivers to have smartphones always fixed in a pre-defined orientation is unrealistic as unmounted sensors may introduce external rotations and accelerations. Additionally, large road elevation changes will reduce model accuracy in 2D models. A method for vehicle state estimation from an unrestrained phone remains an open problem.

## 2.3 Classification

Classification is a machine learning task to learn a mapping from input data to class labels. Classification can be decomposed into the following subtasks:

**Feature Selection** Relevant features for classification are selected from the feature set available in the dataset.

**Dimensionality Reduction** The selected feature vector dimensionality is reduced to reduce the required classifier complexity.

**Classifier Learning** A selected machine learning algorithm learns the structure using example data.

**Ensemble Methods** Classification is improved by aggregating the results of multiple classifiers.

In this section, the classification techniques used in this thesis are reviewed.

### 2.3.1 Dimensionality Reduction

The ”curse of dimensionality” [75] occurs when analyzing and classifying data in high dimensional spaces. Estimators trained in these high dimensional spaces will be easily induced into overfitting the data. Dimensionality reduction algorithms either select a subset of features (feature selection), or create new features as combinations of the input features to reduce the dataset dimensionality. The classifier trained using the lower dimensionality representation is less computationally expensive and less prone to overfitting [76].

#### Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a classical dimensionality reduction method to calculate an orthogonal rotational transform to form a set of linearly uncorrelated variables called principal components (PC) [77]. The aim of PCA is to determine another basis that best re-expresses the data set as a linear combination of the original bases. Suppose the dataset is expressed as a matrix $X_{m\times n}$, where $m$ is the number of features and $n$ is the number of observed examples. Let the linear rotation transformation be represented by $C_{m\times m}$. A new representation calculated by applying the linear transformation is denoted by $Y_{m\times n}$. The relation is presented in Equation 2.37.

To calculate the matrix $C_{m\times m}$, eigenvector decomposition is performed [78]. After normalization with respect to the other eigenvalues, each eigenvalue corresponds to the variance explained by the eigenvector projection [79]. By sorting the eigenvector rows by the eigenvalues in descending order, the eigenvectors will correspond to the strongest direction of variance in descending order as well. Since the computational complexity of eigenvector decomposition is $\mathcal{O}(n^3)$, the algorithm's runtime scales to a cubic degree with the number of observed examples.

$$Y_{m\times n} = C_{m\times m} X_{m\times n} \tag{2.37}$$

Since the eigenvalues are proportional to the explained variance, the proportion of variance explained by the corresponding PC ($e_i$) can be calculated by dividing the eigenvalue ($\lambda_i$) by the sum of all eigenvalues as represented in Equation 2.38. When a plot of the explained variance is generated, it is called a Scree plot.

Truncating the coefficient matrix to $p$ rows produces a new matrix $C_{p\times m}$ that reduces the number of PC dimensions. To determine the optimal number of PCs to truncate for dimensionality reduction, the two common approaches are to use the Scree plot's inflection point or to threshold the cumulative explained variance.

$$e_i = \frac{\lambda_i}{\sum_{j=1}^{m} \lambda_j} \tag{2.38}$$

### 2.3.2 Classifiers

A classifier is an algorithm that finds a mapping from an input vector to a class label, trained from a set of observed data known as training examples. Classifier training is typically supervised.

#### K-Nearest Neighbours (kNN)

K-Nearest Neighbours (kNN) is a classification algorithm that estimates the class label of an input data point by looking up the $k$ nearest similarity matches using a database of pre-labelled data [80]. It is among the simplest of all machine learning algorithms. It is commonly used as a performance baseline due to ease of implementation and robustness.

There are 3 main components to the algorithm. Since training example data take many forms, a preprocessing method must first project the training example onto a defined feature space. To calculate similarity between points, a distance metric must be defined to allow observations in the feature space to be compared. The $k$ nearest training examples can be used as a sample set for labelling new observations.

A classification label can be determined from the sample set through a vote. Majority voting provides a mechanism for conflict resolution in ambiguous cases. For a binary classification problem, selecting an odd $k$ parameter guarantees a result without conflict.

### Support Vector Machines (SVM)

Support Vector Machines (SVM) [81] is a linear classifier that calculates a decision boundary by maximizing the margin between the closest samples [82]. The margin is defined by the closest samples from both classes, which are called Support Vectors (SVs). In the case of separable datasets, the boundary is simply the bisecting hyperplane between the support vectors using the hard-margin definition. In the case of non-separable datasets, a soft-margin is defined or non-linear kernel can be used. In soft-margin SVM, misclassifications are incorporated into the cost function by using a slack variable proportional to the distance to the hyperplane boundary. SVM kernels use non-linear kernels to transform the data into a higher dimensional space to allow linear separability.

Suppose a training dataset is defined by $d$ dimensions. Given $l$ feature vectors and target pairs corresponding to $\{x_i, y_i\}$ where $i = \{1, \cdots, l\}, y_i \in \{-1, 1\}, x_i \in \mathbb{R}_d$. A point $x$ that lies on the hyperplane satisfies $w \cdot x + b = 0$, where $w$ is the normal vector to the hyperplane, $\frac{|b|}{||w||}$ is the perpendicular distance from the hyperplane to the origin, and $||w||$ is the Euclidean norm of $w$. Thus the constraints can be summarized by inequalities described by Equations 2.39 and 2.40. Equation 2.41 combines the two constraints into a single inequality.

$$x_i \cdot w + b \geq 1 \qquad\qquad y_i = +1 \tag{2.39}$$

$$x_i \cdot w + b \leq +1 \qquad\qquad y_i = -1 \tag{2.40}$$

$$y_i(x_i \cdot w + b) \geq 0 \qquad\qquad \forall i \tag{2.41}$$

Let $d_+$ and $d_-$ be the shortest distances from the separating hyperplane to the closest positive and negative examples respectively. The margin of a separating hyperplane is defined to be $d_+ + d_-$. For the separable case, the SVM algorithm optimizes for the separating hyperplane with largest margin. This can be represented with a Lagrangian formulation in Equation 2.42.

$$L_P = \frac{1}{2}||w||^2 - \sum_{i=1}^{l} \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^{l} \alpha_i \tag{2.42}$$

To optimize $L_P$, we can take the partial derivatives with respect to $w$ and $b$ and set them to zero. That is, $\frac{\partial L_P}{\partial w} = 0$ and $\frac{\partial L_P}{\partial b} = 0$ resulting in Equations 2.43 and 2.44.

$$w = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{2.43}$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \tag{2.44}$$

The Lagrangian formulations can be solved programmatically as a convex quadratic problem. The Lagrangian also has a dual form which can be represented geometrically as described in [83]. The dual equivalent reformulates the problem of finding the SV in the reduced convex hulls. The benefit of this formulation is that the scalar cost is efficiently calculated if there are only few SVs and the kernel trick can be applied.

### Decision Trees (DT)

A Decision Trees (DT) is a machine learning technique that uses a collection of decision nodes, navigational branches, and end nodes to generate a classification result using decision flows. Decision nodes represent a logical test using a feature. A branch represents the test outcome path and navigates the current state towards the next node. End nodes (leaves) represent a decision taken after accounting for the series of previously tested features. The decision state is initialized at the root decision node of the tree and continues until an end node is reached.

The underlying learning strategy is developed using a top down approach, guided by frequency information from training examples [84]. The Iterative Dichotomiser 3 (ID3) algorithm is an iterative approach to generate a DT. Optimality is not guaranteed, but the algorithm generates simple decision trees.

Suppose for a set of training examples, a training target is given by $c$ and the set of unique training targets is $C$. Additionally, $X$ is a set of current data for which entropy is being calculated. Then $p(c)$ is the proportion of the number of elements in class $c$ relative to the number of elements in set $C$. The entropy is then defined in Equation 2.45.

$$H(X) = -\sum_{c \in C} p(c) \log_2 p(c) \tag{2.45}$$

When $H(X) = 0$, the set $X$ is perfectly classified as all the elements in $X$ are of the same class.

The collection of subsets created from splitting $X$ by feature $F$ is represented by $T$ and each data subsets is denoted by $t$. A metric used to measure the difference in entropy before and after the split is called Information Gain (IG). By defining the proportion of the number of elements in $t$ to the number of elements in set $X$ as $p(t)$ and the entropy of the subset $t$ as $H(t)$, we can calculate IG using Equation 2.46.

$$IG(F, C) = H(C) - \sum_{t \in T} p(t)H(t) \tag{2.46}$$

IG is analogous to measuring the uncertainty in $X$ reduced after splitting $X$ by features $F$. Using these definitions, the ID3 algorithm is presented below:

1. Calculate IG of every feature in $S$
2. Split $S$ into subsets using the feature by selecting the minimum entropy calculated by maximizing IG
3. Make a decision tree node using the feature with the maximum IG
4. Recurse on subsets using remaining features

The C4.5 algorithm [85] builds on the ID3 by also using information entropy, but the splitting criterion is the normalized information gain calculated by taking the difference in entropy. The attribute with the highest normalized information gain is selected to make the decision node.

## 2.4 Manoeuvres

Driver manoeuvres are trip segments characterizing driving patterns that the driver controls. There are no standard definitions in the literature, but generally manoeuvres are defined such that they correspond to actions that drivers take. A set of actions capture the essential characteristics of individual driving behaviour [10].

Driving events have been used in several vehicular applications including driver modelling [86] [87] [88] [49], manoeuvre segmentation [47] [16] [89] [86], intersection driving behaviour [90] [91] [92], driving safety [93] [29] [28], and autonomous driving [48] [94] [95]. Common driving manoeuvres from the literature include:

1. Stop/Acceleration/Deceleration
2. Left/Right Lane Change
3. Left/Right Intersection Turns
4. Roundabouts
5. Highway Ramps

Manoeuvre analysis requires decomposing each manoeuvre to understand the vehicle kinematics characterizing the manoeuvre. Stop, acceleration, and deceleration events are typically characterized through longitudinal acceleration [38] [10] or speed [96]. Lane changes and intersection turns are also characterized by angular velocity [5] [97] and lateral acceleration patterns. Strong cross-correlation between the accelerometer and OBD pedal positions demonstrate that IMUs can accurately estimate these vehicle kinematics [39].

Lateral acceleration has been demonstrated to help discriminate between straight and turn manoeuvres. A neural network (NN) was used to label longitudinal acceleration and turns using a GPS and biaxial accelerometer [96]. IMUs used in this study was in a fixed orientation with respect to the car. Seven regions are defined to represent different acceleration conditions using a scatter plot of the biaxial accelerations. Their method uses a tuple of lateral/longitudinal accelerations, velocity, radius of curvature, and orientation sampled at 1 Hz for 3 s. Standard gradient descent, variable learning-rate gradient descent, Levenberg Marquardt, and Bayesian Regularization were employed during training. The best classifier was a 2 layer NN trained using the LM backpropagation, achieving an accuracy of 88 % after 200 epochs. No false-positive rates were reported.

Gyroscopes measuring the yaw rate have also been demonstrated in the literature to improve classification performance when turn manoeuvres are involved. A study conducted by Johnson and Trivedi introduces a Dynamic Time Warping (DTW) algorithm to detect Lane Changes and Intersection Turns using IMU and GPS sensors on a smartphone [38].

Table 2.5: Event Acceleration Thresholds from [10]

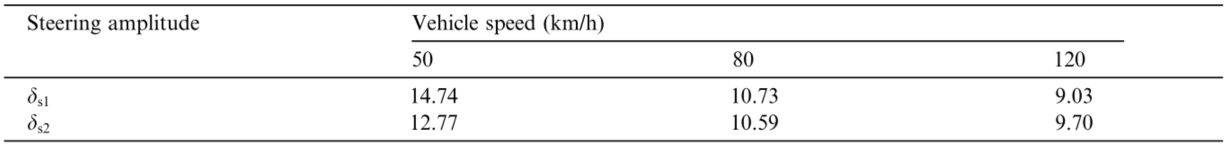
| Event Type | Threshold Sensitivity (g) |
|---|---|
| Acceleration | $a_x > 0.1$ |
| Braking | $a_x < -0.1$ |
| Left Turn | $a_y > 0.2$ |
| Right Turn | $a_y < -0.2$ |

Their method uses a moving average filter to first threshold the rotational energy. Measurements were taken from the accelerometer, gyro, and magnetometer in a fixed orientation relative to the car. An OBD device was also used to collect ground truth data for validation. It was observed that classifiers trained on the accelerometer and gyroscope features obtained an accuracy of 77 % and 79 % respectively. After combining features from all 3 sensors, the accuracy improved to 91 % with a false positive rate of about 5 %. This result demonstrates that a combination of IMU sensors will help in achieving better classification results.

Simple thresholds have been demonstrated to be useful in manoeuvre classification. Braking and acceleration events are generally easy to detect using thresholds [38]. In the previous study by Johnson and Trivedi [38], their method segments events by thresholding the mean squared yaw rate over a 15 s event window. Another study by Paefgen et al. investigated driver event detection using a smartphone in [10]. The study aimed to detect acceleration, braking, and left/right turns. In the study, IMUs were fixed with respect to the car and OBD data was collected for ground truth validation. Acceleration thresholds were introduced with hysteresis to detect different driving events as presented in Table 2.5.

It was observed that detecting driving direction did not work well and manual angle calibration was required. User interaction with the smartphone was found to significantly change measurement results due to changing device orientation. Classification using smartphone also generated many false positives as the measurements were found to be exaggerated. This issue was attributed to bad calibration due to strong deviation from the actual rotation matrix despite manual calibration.

Methods using accelerometers are affected by the sensor position within the car. In the same study by Paefgen et al., the position of the smartphone affected the number of manoeuvres detected by the proposed thresholding method. In different trips with different drivers, the smartphone device was placed in different positions within the car including the co-driver seat, between the driver and co-driver seat, and mounted to the front windshield. Braking events were found to be the most prevalent event type when the phone was in the co-driver seat whereas acceleration was the most noticeable event



| Steering amplitude | Vehicle speed (km/h) | | |
|---|---|---|---|
| | 50 | 80 | 120 |
| $\delta_{s1}$ | 14.74 | 10.73 | 9.03 |
| $\delta_{s2}$ | 12.77 | 10.59 | 9.70 |

Table 2.6: Average Steering Wheel Amplitudes $\delta_{s1}$ and $\delta_{s2}$ and Vehicle Speed from [4]

when the smartphone was in the windshield mount. It was found that acceleration and braking events captured by the smartphone had similar accuracy as the reference IMU. However, turn events captured by the smartphone deviated from those measured by the reference IMU, especially when positioned in the co-driver's seat. This study demonstrates a weakness in accelerometer-based methods as manoeuvre detectability was sensitive to sensor position.

A lane change is a more complex manoeuvre than an intersection turn or straight manoeuvre. Van Winsum et al. investigated different driving conditions to observe and identify which factors affected driver behaviour during a lane change manoeuvre in simulation [4]. The study defines the Time to Line Crossing (TLC) vehicle metric that measures the time the current vehicle takes to cross the line if the steering is left untouched. It is an important measure because it assesses the possibility of a lateral crash or driving off the road [98].

The factors tested in the study included changing the lane, vehicle speed, and direction. The lane change manoeuvre was divided into 3 phases for analysis as seen in Figure 2.6. The steering position peaks were found to be strongly affected by speed but not by lane width or manoeuvre direction. As such, the average steering wheel amplitudes can be represented as an inversely proportional function of speed.

In their study, Analysis of Variance (ANOVA) was performed on the duration of each phase, steering position peaks of phase 1 and 3, and the Time-to-Line-Crossing (TLC) at critical points of each phase. Findings show that average steering wheel amplitudes can be represented as an inversely proportional function of speed as indicated in Table 2.6, but not lane width or manoeuvre direction. A second test in their study involves the correlation of metric z scores through direct and indirect methods as summarized in Figure 2.7. These factors will be considered when generating a simulation for the driver decision.

Chen et al. proposed a method to detect lane changes and intersection turns using the gyroscope sensor on a smartphone in [5]. IMU sensors are mounted in the car and the rotation matrix is calculated using gravity and magnetometer. Additionally, an OBD device is used to collect data for validation. By detecting the number of bumps as seen in

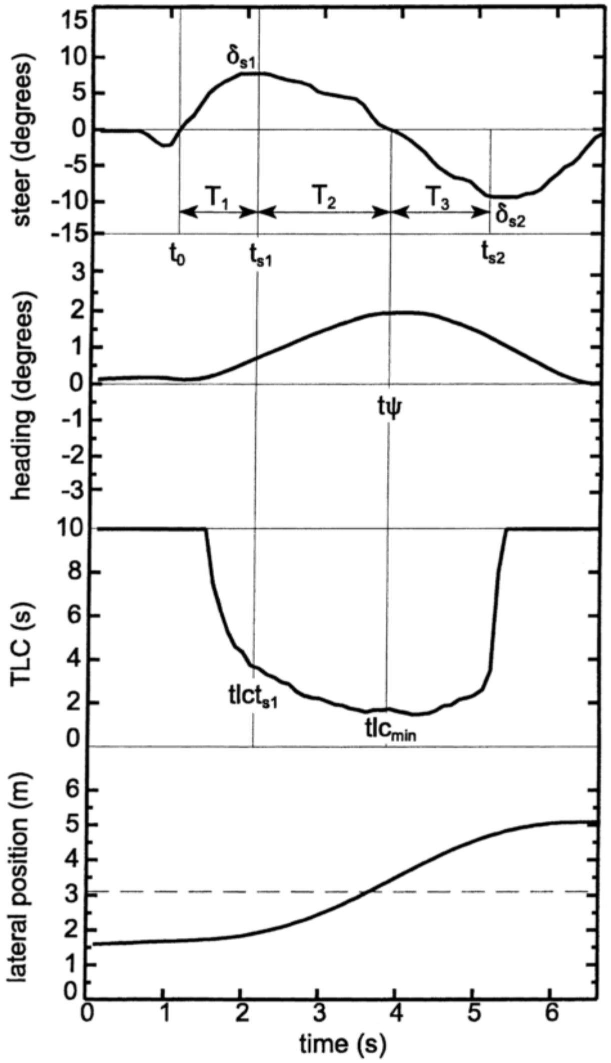


Figure 2.6: Exemplar Lane Change Manoeuvere from [4]

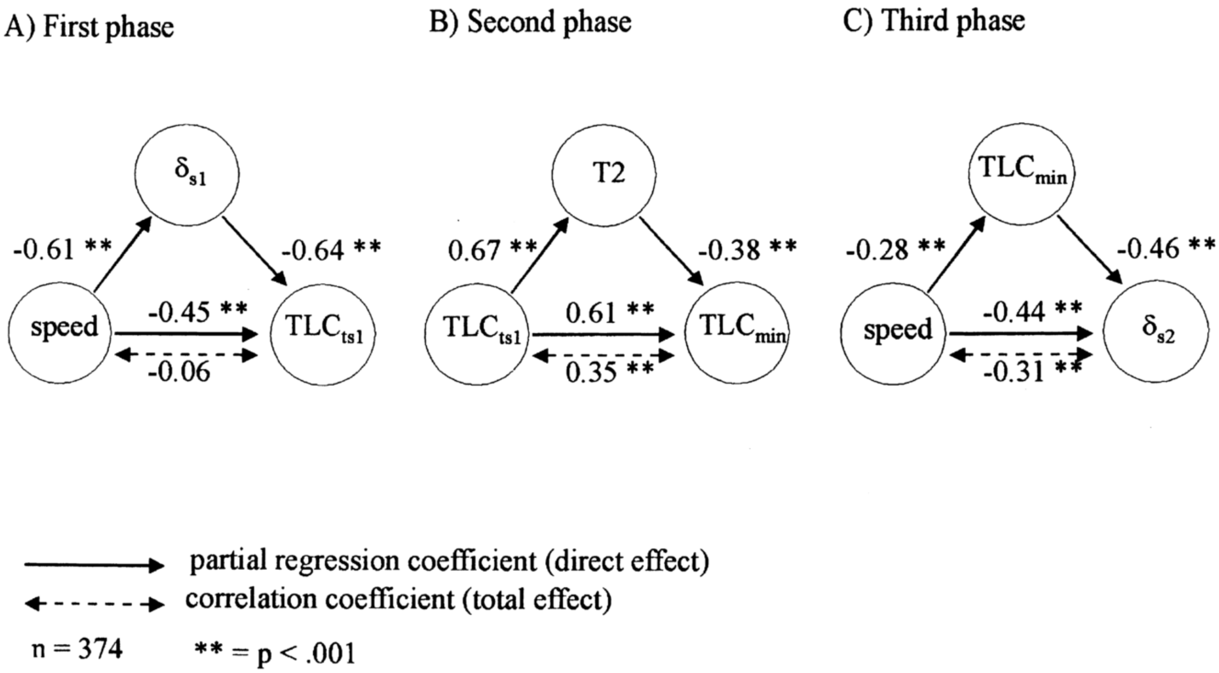
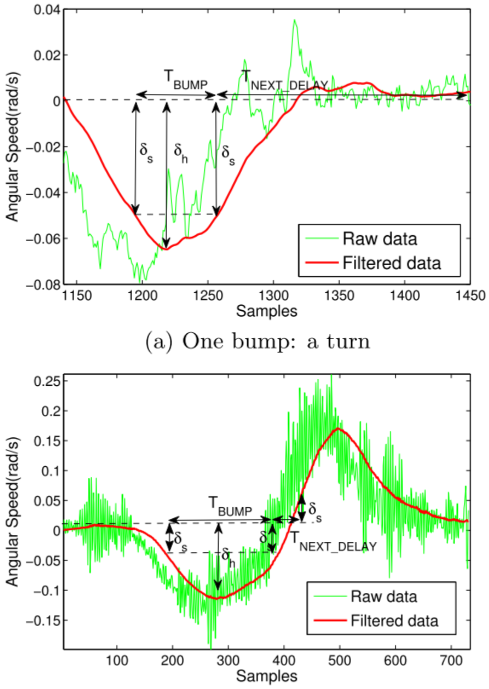


Figure 2.7: Regression Models of Each Lane Change Phase from [4]



(a) One bump: a turn

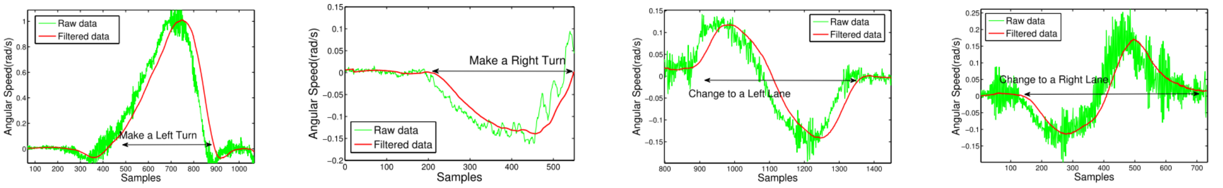
(b) Two consecutive bumps: lane change

Figure 2.8: Bump Count and Manoeuvre from [5]

Figure 2.8 and the direction of the bumps as illustrated in Figure 2.9, a lane change and intersection turn is inferred. Thresholds were empirically determined to design the manual decision tree classifier for manoeuvre detection.

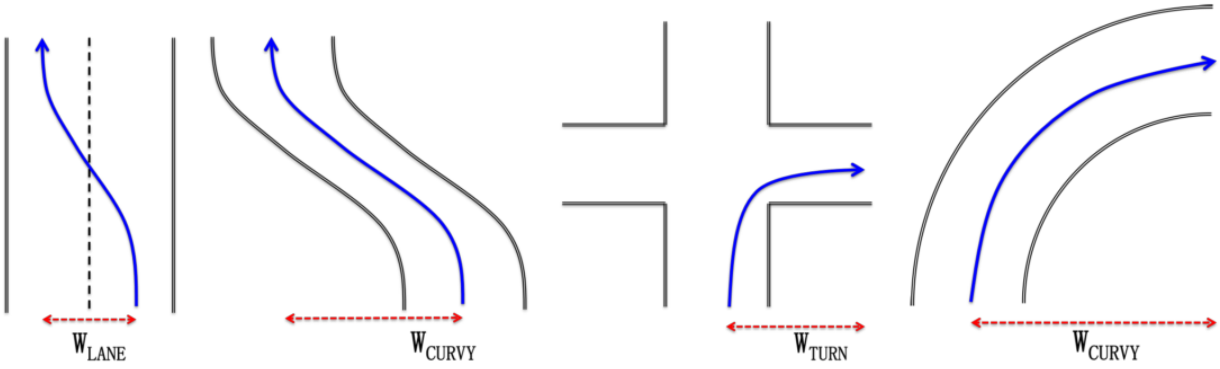
Lateral displacement and heading angle displacement were selected as a feature to distinguish between lane changes, turns, and curved roads as illustrated in Figure 2.10. Empirical data illustrating typical lateral displacements and angle changes for lane changes and U-turns are illustrated in Table 2.7. Furthermore lateral distance from this numerical calculation was used to distinguish between integer multiples of lane changes as illustrated in Figure 2.11. Based on angle displacement, turns were further subclassified as U-Turns or Intersection turns. To further mitigate measurement noise, a Kalman Filter was implemented with GPS velocity and rotated accelerometers in the car frame as measurements. Example use cases included detecting lane changes without signalling and detecting improper lane changes at intersections.

In the study, 5 participants were asked to drive 2 routes twice, totalling 20 trips. No false-positive rates were reported but the performance is summarized in Table 2.8. The



(a) Make a left turn (b) Make a right turn (c) Change to a left lane (d) Change to a right lane

Figure 2.9: Bump Direction and Manoeuvre from [5]



(a) Lane change (b) A S-shaped curvy road (c) Turning (d) A L-shaped curvy road

Figure 2.10: Similar Trajectory Shapes for Different Manoeuvres from [5]

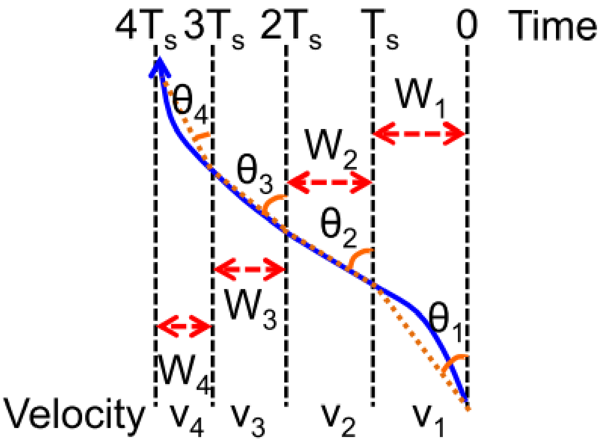


Figure 2.11: Lateral Displacement from Gyroscope and Velocity Measurements from [5]

| | **Lane Change** | | | | **U-turn** | | | |
|---|---|---|---|---|---|---|---|---|
| | **#1** | **#2** | **#3** | **Average** | **#1** | **#2** | **#3** | **Average** |
| **Displacement [m]** | 4.29 | 3.49 | 3.59 | 3.79 | 14.47 | 15.66 | 14.46 | 14.86 |
| **Angle Change [deg]** | 2.03 | 7.49 | 4.12 | 4.54 | 193.73 | 179.85 | 184.41 | 185.99 |

Table 2.7: Lateral Displacement and Angle Change of Lane Changes and U-Turns from [5]

Table 2.8: Classification Performance from [5]

| Manoeuvre | Mounted (%) | In Pocket (%) |
|---|---|---|
| Intersection Turns | 100 | 100 |
| Lane Changes | 93 | 85 |
| Curvy Road | 97 | 92 |

top performing manoeuvres were turns, curvy roads, and lane changes. Furthermore, classification using a mounted device was found to perform better than one in the driver's pocket. Since the algorithm was used with empirically tuned threshold parameters, these performance results are limited to the routes tested and may not generalize well.

Despite several proposed methods in the literature, road curvature and measurement noise make lane changes more difficult to detect compared to other methods. For instance, the study by Johnson and Trivedi excludes non-aggressive lane changes as it was difficult to distinguish the signal from normal driving IMU noise [38]. The smartphone's camera has also been used as a sensor for manoeuvre segmentation. Liu et al. present a method to detect lane changes with Hidden Markov Models (HMM) using observations captured during the beginning of the lane change manoeuvre [58]. They use statistical correlations to predict the trajectory of the remaining manoeuvre and identify if the driving style is normal or dangerous. In their study, the lane change manoeuvre is broken into 3 separate phases: the beginning ($Theta_b$), cutting-in ($Theta_c$), and ending ($Theta_e$) phases. In their study, video data from the SHRP2 and 100-Car naturalistic driving studies are used to train and test the data. Video reference points are used to extract 10 lateral position measurements prior to the cutting-in phase. Using this data, 2 HMM classifiers are trained: one labels normal lane changes and the second labels dangerous lane changes. The classifier attained a true positive (TP) rate of 92.5 % for normal lane changes and 80 % TP for dangerous lane changes. False positives (FP) were not indicated but it was noted that the FP rate was lower than the false negative (FN) rate.

A study by McCall et al. also uses the camera for manoeuvre classification. Their study proposed a lane detection and prediction method using sparse Bayesian learning (SBL) [87]. The sensory platform collects 8-channel video from a fixed camera, OBD, and

GPS data. Lane markers and circular reflectors were detected using 3 directional Gaussians filters and matching road curvature using an adaptive parabolic template. Lane marking positions were filtered using a Kalman filter and achieved experimental position mean absolute error (MAE) of 10.16 cm and standard deviation of 13.17 cm. Head detection was achieved using eye, nose, and mouth detectors. Head pose was estimated using block matching where the driver head is matched between frames. It was observed that driver head motions increased before and during lane changes. The feature vector consisted of 1 second sliding window containing lateral lane position, driver head movement, acceleration, speed, yaw rate, road curvature, heading, and pedal position data. A lane change vs keeping classifier was developed by thresholding the SBL output. It was observed that classification performance was similar between a feature vector with head movement 3 s prior to the manoeuvre is equal to performance when head movement is excluded 2.5 s prior. In situations when head movement did not improve performance, it was observed that multiple consecutive lane changes were being performed. This study demonstrated improved classification performance when the camera is used for lane change classification.

Despite performance advances in vision technology, cameras still have limitations as they depend on visibility [5]. Bad lighting at night, low visibility weather with and snow, pavement wear, and poor camera placement all contribute to lowering the effectiveness of camera dependent solutions.

Different methods for feature reduction have also been proposed in the literature. In a previously reviewed study by McCall et al. feature selection was performed using SBL to prune redundant features generating highly sparse representations. To avoid overfitting an under constrained representation, an independent Gaussian prior distribution is applied to the unknown weights, each with its own independent variance parameter.

Another study by Van Ly et al. investigated using smartphone IMUs to distinguish between two drivers by training SVM and k-means classifiers [39]. Driving manoeuvres in the study analyzed acceleration, deceleration, and turning. Smartphone sensors were used with a known orientation and measurements were validated using OBD measurements as the ground truth reference. A feature vector was constructed using angular velocity, longitudinal acceleration, and event duration. A histogram with 5 bins using min, max, mean, and variance was used. Subsets of these bins were selected as features for classifier training. It was observed that using more bins did not improve performance and keeping the feature vector small prevents overfitting. While this method was used for driver identification, they demonstrate that statistical feature vectors can be used for feature selection.

### 2.4.1 Summary

The reviewed literature includes a number of methods for segmenting and recognizing manoeuvres. These include simple methods to threshold accelerometer and gyroscope measurements, but these methods may not generalize well as they depend on a the phone remaining fixed within the vehicle. Cameras have been demonstrated to be effective to detect lane markers under the right conditions, but they are inconvenient to mount and are considered invasive to privacy.

For each of the classification papers reviewed with IMUs, the sensors were placed in a known orientation and position within the car. Validation datasets were collected using an OBD or camera for ground truth validation. These methods required training data to be labelled by an expert or automatic threshold algorithm. Validation dataset collection and expert labelling is time consuming.

## 2.5 Driver Behaviour Analysis

Driver behaviour is a complex and dynamic interaction describing a sequence of state transitions between the driver, the vehicle, and the environment [99]. It can characterize the driver's response to a situation or stimulus, or describe a personality or driving style. For instance, fatigued driving is a symptom of sleep deprivation. On the other hand, manoeuvres consist of short term actions under the driver's control. An example is a double lane change to switch to a specific lane. Over the course of a driving session, a driver may exhibit a particular driving pattern in response to these states. Grouping these patterns allows a driver's behaviour to be labelled for further analysis. Since many methods for driver behaviour analysis depend on proper manoeuvre detection, improving manoeuvre detection has a critical role in improving the baseline performance in these methods. Certain methods to analyze driver behaviour are also common in the reviewed literature. As such, the reviewed literature in this section will be organized by analysis techniques. In this section, a summary of driver behaviours and techniques for behaviour characterization is provided.

### 2.5.1 Types of Behaviours

There are a number of endogenous driver behaviours frequently classified in the literature. The ones that will be reviewed in this section are:

1. Speeding
2. Driver Intoxication
3. Aggressive Driving
4. Fatigued Driving
5. Distracted Driving

### Speeding

Speed limits are posted on roads to determine safe maximum or minimum velocities drivers should maintain. Exceeding these limits is known as speeding or over-speeding. Drivers commonly exceed these limits and local law enforcement typically has a certain threshold before starting to enforce these limits.

Detecting speeding behaviour requires a measurement device to measure the vehicle's velocity. This has been measured in the literature using GPS [38] [100] [101], the speedometer through the OBD port [17], and speedometer video data [63]. While speed measurements from the smartphone may be inferior in accuracy due to signal interference [19], GPS is highly feasible and portable for telematic applications.

Different speeding measures have been developed in the literature. The most common approach is to simply threshold the current vehicle speed with respect to posted speed limits [100] [17]. Certain studies incorporate speeding into a safety score by deducting the score proportional to the over-speed amount [29] .

### Driver Intoxication

Driver intoxication occurs when a person drives under the influence of sensory impairing substances such as alcohol or drugs. Different blood alcohol levels (BAL) affect sensory systems such as visual perception ($BAL \geq 0.05\,\%$), judgment ($BAL \geq 0.07\,\%$), and operation errors ($BAL \geq 0.15\,\%$) [102].

Several methods in the literature have been proposed to determine if a driver is under the influence of alcohol or drugs. These methods can be either invasive or non invasive. Invasive methods include the use of electroencephalogram (EEG) sensors [103] [104], breath alcohol concentration (BAC) device [105], or camera for video recording [106] [107]. Non-invasive sensory methods include sensors such as accelerometers [67].

### Aggressive Driving

Aggressive driving is exhibited by a driver who drives at high speed and a high degree of acceleration putting other drivers at risk [99]. It is one of the most frequently investigated driving behaviours in the literature.

A wide variety of sensors are used to detect aggressive driving such as accelerometers [108] [109], gyroscopes [38] [28], magnetometers [110], OBD devices [111], Wii remote [17], and GPS. Classification methods for aggressive driving include fuzzy logic [108], Bayes classifiers [17], logistic regression [111], SVM [39], and decision trees [110].

Aggressive driving is a dynamic process involving other drivers on the road. Interactions with other vehicles involve more than simple acceleration. Respecting rules such as yielding to the right of way is important in understanding the driving context when classifying aggressive driving. To fully understand aggressive driving, future research using other methods such as large naturalistic driving studies and assessment of psychological state are required.

### Fatigued Driving

Fatigue is an evolving exhaustion process that increases during driving [99]. Common symptoms include inability to focus, forgetfulness, poor judgement, slow reaction times, daydreaming, lane drifting, yawning, or eye rubbing.

Fatigued driving is frequently compared to intoxicated driving. Fairclough and Graham investigated driver impairment symptoms between sleep deprived and alcohol intoxicated subjects in simulation [112]. They observed that fatigued driving is as dangerous as intoxicated driving. In the study, 64 male subjects were divided evenly into 4 subject groups: control, partial sleep deprivation (PartSD, 4 h sleep), full sleep deprivation (FullSD, no sleep), and alcohol group (full sleep, blood alcohol level of 0.08 % to 0.1 %). Both FullSD and alcohol groups exhibited more lane crossings suggesting lower lateral control. The FullSD group distinguished themselves from the alcohol group by having more lane crossings as the study progressed suggesting a higher fatigue rate. Furthermore, the FullSD group demonstrated a higher headway suggesting judgement to pre-emptively compensate for impaired longitudinal control. Subjectively, the FullSD group experienced increased temporal and physical demand compared to other groups. Alcohol was observed to reduce the frustration level. Task-relevant thought was higher in the FullSD group compared to the control and alcohol groups suggesting high concentration requirements.

A variety of methods in the literature have been proposed to detect fatigued driving. Different sensory methods have been used for these systems such as cameras [113] [114] [115], EEG [116], electrocardiograms (ECG) [117], lane position sensors [118], lead vehicle distance sensor [119], and gyroscopes [40] [120]. Many methods in the literature assume that vehicle steering is measurable, typically through the OBD port or estimated by IMU sensors. For instance, the gyroscope have been demonstrated to be effective in estimating the steering wheel angle [5] [48] which is correlated to lane position [121] [4] [122].

These studies propose methods to predict driver fatigue through vehicle steering. Despite methods to estimate the steering angle through accelerometers and gyroscopes, it is difficult to distinguish between steering due to road curvature or driver related behaviours [38]. Furthermore, additional methods in correlating psychological states measured through EEG/ECG to inertial measurements are required to cross-validate these results.

#### Distracted Driving

Maintaining eye contact with the road is fundamental to safe driving. Inattentive or distracted driving occurs when the driving task is not the primary focus for a driver. Examples of distracted driving include eating, conversing with passengers, rubbernecking, phone usage, and map reading.

There are two main methods for detecting distracted driving. Since many distracted driving behaviours are characterized by the driver's visual inattention, distracted driving is frequently detected by using the driver's gaze by using cameras [123] [115]. A second method for detecting distracted driving uses inertial sensors to estimate vehicle kinematics to infer driver distraction [124]. When a driver is distracted, a common symptom is that they lose control of the car. This results in lane weaving and lane drifting.

These methods characterize distracted driving through lane changes and lane weaving. However, distractions are more complex than these simple observations. For instance, failure to follow road rules such as traffic control and right of way could occur when a driver is distracted. Methods need to be developed to identify distracted driving through more advanced characteristics and balance a level of privacy acceptable to drivers.

### 2.5.2 Driver Behaviour Analysis Methods

Simple thresholds are a basic method to create a decision boundary for a binary classification problem. In one study that focuses on smartphone sensors, Dai et al. threshold longitudinal and lateral accelerometer values for driver intoxication detection [67]. To validate the method, 72 sets of drunk driving related manoeuvres were tested including weaving, swerving, turning with a wide radius, and erratic speed changes. Lane maintenance issues were detected using a 5 s sliding window and thresholding min/max lateral acceleration values using $4\,\frac{\mathrm{m}}{\mathrm{s}^2}$. To detect speed control issues, different thresholds were used for acceleration and deceleration. These thresholds corresponded to $2.4\,\frac{\mathrm{m}}{\mathrm{s}^2}$ and $2.7\,\frac{\mathrm{m}}{\mathrm{s}^2}$ respectively. A hysteresis filter for 2 consecutive pattern matches was used to denoise the data. Despite low FP rates reported in this method, the set of drunk driving cues were limited and hard to distinguish from normal driving patterns (eg. weaving and lane changes have the same profile) [38] [125].

In a study by Bergasa et al., a method is proposed that detects inattentive and drowsy driving [28]. Inattentive driving is characterized through lane drifting and lane weaving behaviours. These behaviours needed to be distinguished from normal coasting, acceleration, and deceleration manoeuvres. The sensory platform uses a smartphone's rear camera, microphone, accelerometer, gyroscope, and the GPS. The smartphone was mounted in a vehicle with its 3 axes aligned to the car frame. Distracted driving was detected using the frequency of violated accelerometer and gyroscope thresholds corresponding to lateral accelerations and angular velocities. The accelerometer was sampled at 100 Hz and filtered using a Kalman filter. To ensure observability, a minimum acceleration threshold was used, a minimum time period was required, and minimum longitudinal velocity of $50\,\frac{\mathrm{km}}{\mathrm{h}}$. Thresholds were set to correspond to different intensities (low, medium, high). After an event was detected, a hysteresis period was used to denoise future detections. These measurements correspond to sudden acceleration/deceleration and high lateral accelerations. The method decouples the lateral acceleration due to the road curvature by using the GPS to calculate course road centripetal acceleration values. These values are subtracted from vehicle lateral accelerations to determine the vehicle centripetal accelerations.

In this inattentive and drowsy driving study, 12 participants (9 males and 3 females) each performed 2 sets of trials: controlled manoeuvre trips and naturalistic driving trips. In the controlled manoeuvre trips, the drivers were asked to perform lane drifting and lane weaving to simulate inattentive driving. Naturalistic driving trips were recorded without explicit driving directions. Sessions were 20 min in duration and were at different times of day (morning, afternoon, and night) over 20 driving sessions. In total, 480 min of driving data were collected on pre-defined highway routes. Drivers were asked to perform 2 lane drifting, 6 lane weaving, 4 sudden acceleration, 6 sudden brake, and 2 sudden turn manoeuvres. A total of 282 inattentive driving events were recorded, 245 from controlled experiments and 37 from naturalistic driving. A mean precision and recall of 82 % and 92 % respectively were achieved for all driving events. Lane drifting precision and recall were 89 % and 100 % respectively. Lane weaving precision and recall were 93 % and 96 % respectively.

Thresholds have also been used to characterize drowsy driving. In a study by Fukuda et al, a method is proposed to detect driver drowsiness using the steering angle of the car [126]. Steering adjustment points were used to characterized the steering angle with a resolution of 0.1° and sampling period of 128 ms. Steering Adjusted Interval (SAI) length and variability were both observed to decrease with increased vehicle speed and are consistent with other past works [4]. These profiles differed between drivers but the trend was consistent across all drivers. For preprocessing, steering interval measurements were normalized by converting measured SAI to relative values at 80 $\frac{\text{km}}{\text{h}}$. Furthermore due changing variability of the SAI length, the threshold to determine drowsiness was set as a function of vehicle speed.

The proposed method uses a recent average SAI and checks if it exceeds a normal driving SAI threshold to identify a drowsy driver. The threshold SAI is calculated to be the product of the learned normal driving SAI, correction coefficient for normal driving as a function of vehicle speed, and drowsiness coefficient as a function of each driver. To validate the proposed approach, 10 drivers were evaluated for 1 h while driving at approximately 80 $\frac{\text{km}}{\text{h}}$. The drivers subjectively reported their drowsiness as alert, light drowsy, or drowsy. $\alpha$-wave rates were also measured as it physiologically correlated with drowsiness. It was found that estimated light drowsiness had an accuracy of 76 % and 88 % when compared to $\alpha$-wave rates and self-reported drowsiness respectively.

### 2.5.3 Advanced Classifier Models

A number of studies apply advanced classifier techniques to identify aggressive drivers. In a study by Hong et al., a method is proposed to identify aggressive drivers using a naive Bayes classifier [17]. The sensory platform uses a smartphone, telematic device, and additional IMU sensors from a Wii remote. To characterize aggressive behaviour, manoeuvre recognition was used to segment driving situations. The study used start, stop, high speed, turn, before turn, and after turn to segment portions of the trip for analysis. It was observed that the car's maximum speed is not a major factor in distinguishing aggressive drivers. However, aggressive drivers tend to delay required speed changes resulting in more forceful changes when they do occur. As such, acceleration is a major factor in identifying aggressive drivers. Aggressive drivers tend to experience higher g-forces within the car in both lateral and longitudinal directions. This trend was expected for stopping and starting the vehicle, but the phenomenon also occurred during high speed travel

($> 50\,\frac{\text{km}}{\text{h}}$). Aggressive drivers were identified from prior traffic violations (violation-class) and questionnaire responses (questionnaire-class). Using all 3 sensor platforms resulted in a classification accuracy of 90.5 % for the violator aggressive-class and 81 % for the questionnaire aggressive-class.

NNs have also been used to analyze driver behaviours. In a paper by Sandberg and Wahde, a method was proposed to detect drowsy driving in simulation using NNs [127]. Base features were created by measuring speed, lateral position, steering angle, and yaw angle at 25 Hz. Taking the time derivative of each base feature generated an additional feature set. These features and their derivatives correspond to manoeuvre recognition features. Aggregate features included standard deviation of base features, lateral position mean squared error, lane exit percentage, Time-to-lane crossing, intermediate/large steering wheel reversals, and rapid steering wheel movement. Some of these measurements can be obtained using IMU sensors embedded on a smartphone. The neurons used sigmoid activation functions and the network trained using Particle Swarm Optimization (PSO).

By evaluating the discrimination power for individual features, it was observed that lateral position, lateral velocity, and yaw angle were the most discriminative features using a simple thresholding classifier to test classification performance. Using all the 15 indicators achieved the best performance at 75.5 % f1 score. It was observed that f1 scores ranged between 59 % to 89.2 % when validated on a per driver basis. The variance in f1 scores demonstrate that drowsy driving in some drivers may be more observable than others.

Cameras are one of the more robust sensors for distracted driving. In a study by You et al., a method was proposed to detect drowsy and distracted drivers using cameras, IMUs, and the GPS found on smartphones [128]. To provide simultaneous access to both front and rear cameras, an intermittent camera switching system was developed. The study detected drowsy driving, inattentive driving, tailgating, lane weaving/drifting, and careless lane changes. Manoeuvre detection using the IMU was employed to control cameras switching so that these behaviours were captured more accurately. Distracted driving was characterized when the driver was not facing forward for longer than 3 s during straight driving. Face segmentation used a Haar-like features was used to detect the face from the video and a decision-stump classifier was trained to label the face direction. Facing right precision and recall were both 68 %. Facing left precision and recall was 79 % and 88 % respectively. Confusion was attributed to camera switching and minor face turns. Distracted driving FN were caused by face direction errors, camera switching, and turn detection errors. Distracted FP were caused by turn detection errors and camera switching.

Complex architectures such as Vehicular ad hoc networks (VANETs) have also been designed for driver behaviour analysis. In a study by Al-Sultan et al., a VANET was proposed to detect driving behaviour [99]. A 5-layer architecture was developed to detect normal, fatigue, drunk, and reckless driving. Information variables were divided into 2 groups: first/second and fourth/fifth layers. Group 1 provided information affecting behaviour such as circadian rhythm (time and time zone) and environment (noise and temperature). Group 2 provided information specific to certain behaviours such as speed control, lane position, lateral/longitudinal acceleration, eye percent closure (PERCLOS), average eye closing speed (AECS), and blood alcohol level (BAL). These nodes correspond to manoeuvre detection allowing a separate layer to characterize driver behaviour. A third intermediate layer was the driver behaviour and connected the 2 groups.

Using these layers, a directed acyclic graph was developed with the previous timestep label connected to the third layer's input. The network's prior and transition probabilities were parameterized using past literature. The system was tested in simulation. Common effects were observed:

- Fatigue and bad environment implying fatigued driving
- Awake and good environment implying reckless driving
- Fatigued and good environment implying alcohol intoxication

### 2.5.4 Environmental Factors

Exogenous environmental factors also have an impact on the driving style during a trip. Time of day, road type, and weather each affect how the driver may react to road conditions. For instance, time of day affects visibility and driver alertness. Weather conditions include precipitation such as snow, rain, and ice. Different road types include highway, country, urban, and city roads. Each road type has different traffic volumes, traffic flow control methods, speed limits, and number of aggressive peer drivers.

The Federal Highway Administration (FHWA) is currently funding multiple efforts to demonstrate and bring about the benefits of UBI [21]. For instance, they are helping insurance companies investigate precisely how driving conditions (congestion, roadway type, weather, and night versus day) impact accident related insurance claims. These conditions are easily collectable through UBI telematic systems.

Methods for incorporating environmental conditions into driver behaviour analysis have also been developed in the literature. For instance, Castignani et al. include weather and time of day in their platform for driver behaviour profiling [29]. Weather conditions include normal, rain, storm, snow, and fog. Possible values for time of day are day or night. In another study, Hosseinioun et al. developed a method to detect and evaluate driving events using a smartphone [100]. A driving performance index score was proposed that incorporated speeding, phone usage, weather (rain/snow), time of day, driving duration, and driving breaks into the calculated metric.

Driver experience also has a role in these environmental factors. Konstantopoulos et al. investigated driver safety of drivers with different experience levels by tracking eye movements in simulation [129]. By studying eye movements of driving instructors and learner drivers, it was observed that experienced drivers had increased sampling rate, shorter processing time, and broader road scanning than learner drivers. Night and rain driving was considered high risk due to hampered visual search strategies during these conditions. Driving during night and rain conditions increases the risk of a crash, which may be related to visual search strategies that become more efficient with increased experience.

Different weather conditions also impact stopping distance. Riener and Ferscha investigated the impact of proactive braking on traffic flow in simulation [130]. Maximum deceleration values due to physical weather conditions impacted static friction braking and kinetic friction slippage. Tire and road qualities were also observed to affect braking distance.

Road types also affect driving behaviour. Araujo et al. developed a smartphone application to evaluate efficient driving patterns [131]. They defined 3 road types: urban, highway, and combined. Each road type generated different recommendations for optimal fuel consumption based on observed driving behaviour. Recommendations included analyzing idle times, gear shifts, acceleration, coasting, throttle aggression, and speeding.

A study by Paefgen et al. analyzes 3 road types: city, country, and highways [10]. Turns and braking were twice as common as accelerating events, suggesting the need for braking and turning to occur more frequently in real traffic situations. City environments had twice the number of events as compared to country roads. Despite high average speeds, event counts on highways were low. City traffic was found to require more acceleration, braking, and sharper turns. While braking and acceleration occur less frequently on country roads, turns were equally important due to increased rotary traffic outside the city.

In a study by Andrey et al., the impact of weather information on road safety is investigated [132]. Despite understanding the physics of road surface friction and driver visibility impact on safety, converting estimates of frictional change or impaired visibility into collision risk is much more difficult. They observed that collision and injury risk increase with higher precipitation, wind, and time of day.

There are many environmental factors that impact driving behaviour. Trip conditions such as weather and time of day can be easily measured through OBD or smartphone devices within the car. Different environmental conditions impact driver behaviour and aggressiveness by affecting the driver's response. By improving driver acceptance of UBI, future studies can collect more data to investigate accident rates and insurance claim data with respect to these environmental factors.

### 2.5.5 Summary

This section reviewed some of the methods for driver behaviour analysis. Many of these methods use manoeuvre detection techniques for their analysis. Improvements in manoeuvre detection increase the underlying performance of these methods. Improved manoeuvre detection developed in this thesis will enable performance improvements for driver behaviour analysis.

# Chapter 3

# Vehicle State Estimation

Measurements collected within the vehicle are observations of the vehicle's state. To analyze driver behaviours and manoeuvres generating these states, the vehicle's state must be estimated from observations. Vehicle state estimation also rejects measurement noise through filtering methods leading to more accurate state estimates. Vehicle models can also be used to generate simulation data for algorithm training.

In this section, a full state space model is presented using the Two Wheel Robot model described in Section 3.1 and the system update model described in Section 2.2.1. Noise models to simulate the IMU and Velocity sensors are investigated in Sections 3.2 and 3.3 respectively. An EKF is then formulated using the equations in Section 2.2.4. A carrot controller for simulated lane keeping is presented in Section 3.5. Accuracy results for a simulated trial are presented in 4.5. Finally, a method to generate training data for classifier training is presented in Section 3.10 using the simulated velocity controller presented in Section 3.9.

## 3.1 Vehicle Model

The Two Wheel Robot Model presented in Section 2.2.1 is used for the vehicle model. The state and control vectors are restated as follows:

$$X_t = \begin{bmatrix} x_{t-1} + v_t \cos\theta \Delta t + v_t \cos\theta \frac{(\Delta t)^2}{2} \\ y_{t-1} + v_t \sin\theta \Delta t + v_t \sin\theta \frac{(\Delta t)^2}{2} \\ \theta_{t-1} + \omega_t \Delta t \\ v_{t-1} + a_{t-1} \\ \omega_{t-1} + \alpha_{t-1} \\ \widetilde{\alpha}_t \\ \widetilde{a}_t \end{bmatrix} \tag{3.1}$$

Sensors placed within the car provide measurements from two sources: the GPS and the IMU. To emulate the measurement using a mobile device in the vehicle, sensor values will be simulated using the filtered state values from the state model described in Equation 3.1 and Gaussian noise will be added to the values to simulate sensor noise. The sensor noise is assumed to be zero mean Gaussian with the variance estimated as described in Section 3.2. The GPS sensors provide longitude and latitude values corresponding to $\bar{x}_t$ and $\bar{y}_t$ measurements. The GPS can also estimate the linear velocity ($\bar{v}_t$) using the Doppler effect of the GPS base frequencies. The gyroscope sensor measures the rotational velocity ($\bar{\omega}_t$) directly. The linear acceleration ($\bar{a}_t$) is obtained from the device's accelerometer. In the current model, the phone is assumed to be aligned relative to the car and located at its centre of gravity (CG).

The normal acceleration measurement ($\bar{a}_{n,t}$) can be derived in terms of the state as follows,

$$l = r\theta$$
$$v = r\omega$$

$$\frac{1}{r} = \frac{\omega}{v} \tag{3.2}$$

In this equation, $l$ is the linear arc distance, $r$ is the arc radius, and $\theta$ is the arc angle. After differentiating with respect to time, $v$ is the linear velocity and $\omega$ is the arc angular velocity.

However from centripetal force,:

$$a_n = \frac{v^2}{r}$$

Substituting Equation 3.2,

$$a_n = (v)^2 \tfrac{\omega}{v}$$

$$a_n = v\omega \tag{3.3}$$

The final measurement matrix is represented below,

$$Y_t = \begin{bmatrix} \bar{x}_t \\ \bar{y}_t \\ \bar{v}_t \\ \bar{\omega}_t \\ \bar{a}_{l,t} \\ \bar{a}_{n,t} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ v_t \\ \omega_t \\ a_t \\ v_t\omega_t \end{bmatrix} \tag{3.4}$$

## 3.2 IMU Noise Model

A measurement experiment was conducted to observe the sensor accuracy, noise distribution profile, and the distribution parameters for the simulation. The noise model experiment used a Samsung Galaxy S4 mobile phone running Android 4.4.2 indoors in the University of Waterloo campus. The mobile device was at rest with the screen of the phone pointing upward and the top of the phone pointing North. In this experiment, sensor values were obtained from the accelerometer, gyroscope, and magnetometer using the Android SDK. The values were logged using the ADT plugin for Eclipse.

The data collected and presented for this thesis is able to utilize GPS and GLONASS satellites. Within the data collection area in Waterloo, there were at least 11 GNSS satellites in view assuming a threshold of 30° angle of view. As such, it is reasonable to expect less than 19 m of error based on the findings presented in [1].

Smartphone manufacturers typically bulk calibrate the smartphone using preset values before sale to the consumer. The Android API has a proprietary method to calibrate the sensor and measurements were taken from this API. There were no further attempts to calibrate the sensors. Therefore it is possible that multiplicative errors exist in the noise model. In the histograms, biases due to uncalibrated sensors also exist. Additionally, it is known that sensors cannot be perfectly aligned orthogonally, however this source of error was neglected. These assumptions were made as it was only necessary to obtain simple baseline noise distributions for simulation.

To determine a proper noise model for the simulation, histograms of sensor values were required. The device was in an idle state pointing to North to ensure constant mean. This

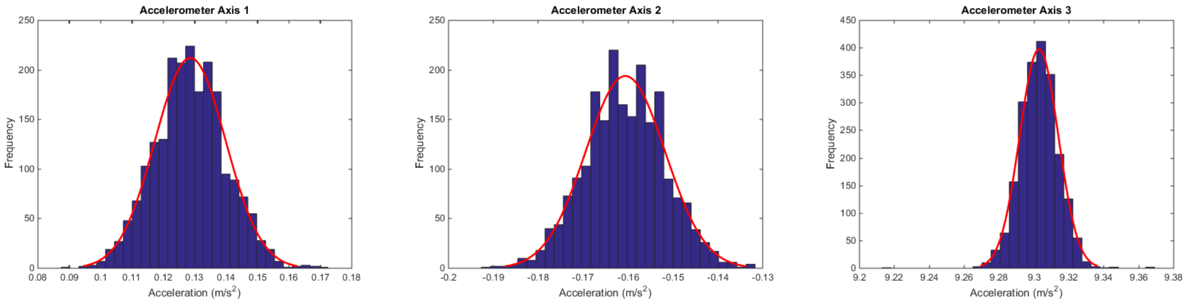


Figure 3.1: Histogram of 2116 Accelerometer measurements around the (from left to right): (a) x-axis, (b) y-axis, (c) z-axis

allows the variance of the sensor to be estimated using the data collected. To measure the distribution of sensor values, 3 trials were performed. Each trial collected data on separate sensors specifically the accelerometer, gyroscope, and magnetometer. Each trial was 20 seconds in duration and collected data at 100 Hz. This high measurement rate demonstrated that data could be collected at high frequency to help reject noise. The small duration also allows a constant bias assumption to be valid.

Figure 3.1 illustrates the first set of trials with the accelerometer. It was observed from the trials that the acceleration is approximately 9.3. Most of this bias acceleration can be attributed to Earth's gravity. The average standard deviation of all 3 axes was determined to be 0.010 45 $\frac{\mathrm{m}}{\mathrm{s}^2}$. Figure 3.2 illustrates the second set of trials with the gyroscope. It was observed that the rotation about the z-axis has a non-zero mean. Part of this bias can be attributed to Earth's rotation. The average standard deviation of all 3 axes was determined to be 0.003 883 $\frac{\mathrm{rad}}{\mathrm{s}}$. Figure 3.3 illustrates the third set of trials with the magnetometer. It was observed that the magnetometer was strongest on the y-axis. This can be attributed to the Earth's magnetic field. It was observed through the sparse distribution that the magnetometer precision is poor. Additionally in further trials, it was determined that the magnetometer is highly sensitive to the presence of other magnetic fields. Since the environment inside a vehicle acts as a Faraday's cage and is highly unpredictable, it was determined that the use of the magnetometer measurement would only lower the accuracy of drift corrections.

Further GNSS trials were performed outdoors within the University of Waterloo campus. With an elevation mask of 10 degrees in clear skies, the satellite predictor tool estimates that 15 GNSS satellites (GPS and GLONASS) are available for a GNSS device to utilize. Furthermore, Dilution of Precision (DOP) measurements are summarized in Table 3.1. GNSS measurements on the device are consistent with the environment estimates.

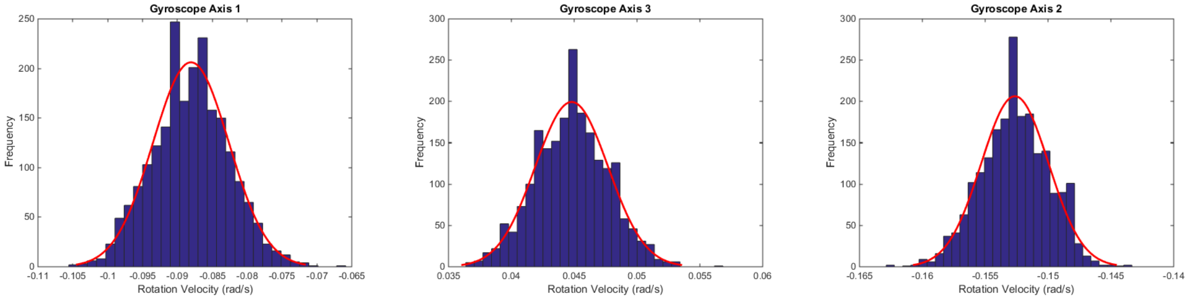


Figure 3.2: Histogram of 2127 Gyroscope measurements around the (from left to right): (a) x-axis, (b) y-axis, (c) z-axis

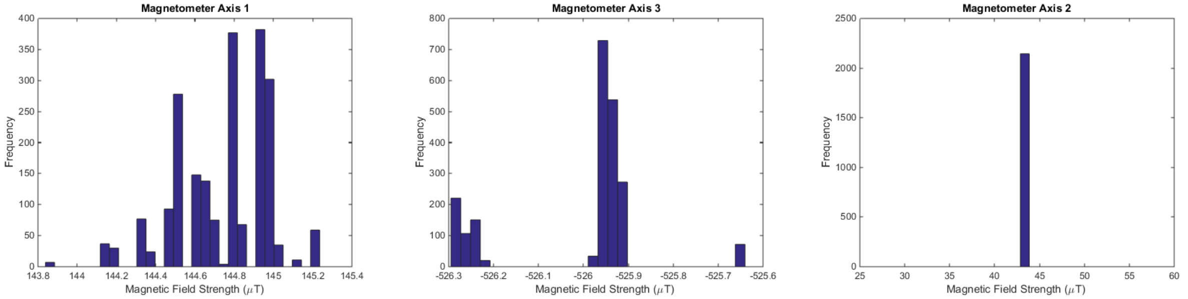


Figure 3.3: Histogram of 2145 Magnetometer measurements around the (from left to right): (a) x-axis, (b) y-axis, (c) z-axis

Table 3.1: DOP Estimates from GNSS Trial

| Measurement | Value |
|---|---|
| DOP | 1.7 |
| PDOP | 1.3 |
| VDOP | 1.1 |
| HDOP | 0.7 |

The position variance accuracy was measured to be accurate to ±4 m. For the purpose of the simulation, this accuracy is estimated to be the 95 % confidence interval of a Gaussian process leading to a standard deviation of the noise model to be 2 m. Using a similar argument from sources, the velocity standard deviation is estimated to be 0.0125 $\frac{\mathrm{m}}{\mathrm{s}}$.

Despite reading the gyroscope measurements from the Android API, the values were miscalibrated in a subset of the trials. Figure 3.4 illustrates that an internal calibration procedure is performed after data collection commences near timestep 20. During high angular velocity activity during this window, the angular velocity zeroed incorrectly and leads to miscalibrated gyro measurements as seen in Figure 3.5. A secondary calibration technique was employed to compare the difference in average angular velocity magnitude between timesteps $20 - 25$ and $40 - 50$. If the magnitude has a difference exceeding a threshold of 0.05, then the signal is re-zeroed in the later interval. This simple technique detects a miscalibration by the Android API at the beginning of the trip and corrects the bias for the remainder of the trip. Additional plots for more trips are included in Appendix B. On the other hand, the smartphone shifted from the original orientation mid-trip causing a shift in the gyro biases. These trips were excluded from the results.

## 3.3 Velocity Noise Model

The velocity sensor noise model was derived from data provided by Intelligent Mechatronics System (IMS). 10 participants collected the mobile phone data from 23 trips using Android or Apple devices. Within the dataset, 5 and 18 trips were collected on the Android and Apple platform respectively. Each trip contains 2 datasets: one from a telematic tracking device connected to the vehicle's Onboard Diagnostic (OBD) port and another from a Mobile Application (MA) running on the driver's mobile device. The OBD telematic device collects data from an internal Differential GPS (DGPS) and the vehicle's OBD port. The velocity measurements from the OBD data are collected from the vehicle's speedometer.

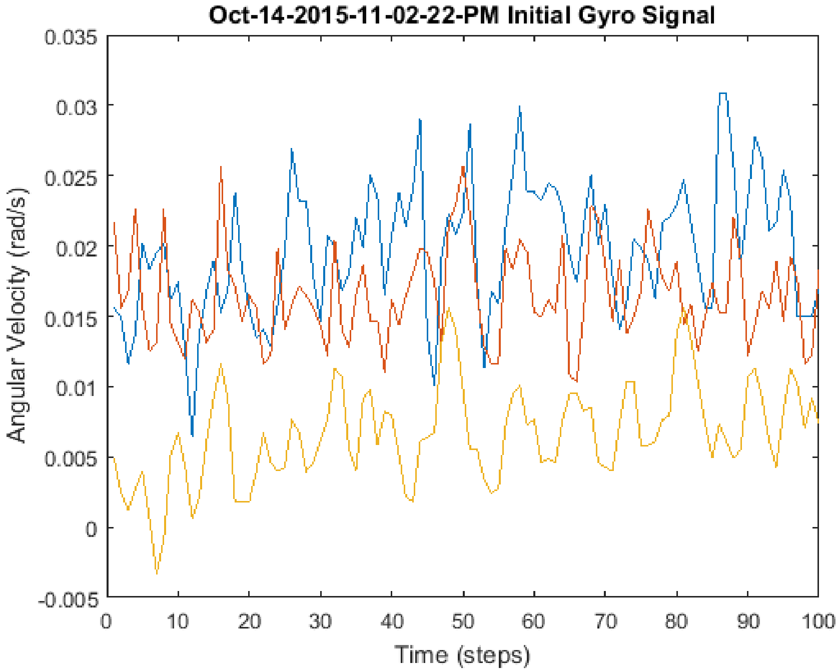


Figure 3.4: Calibrated gyro measurements from Android API

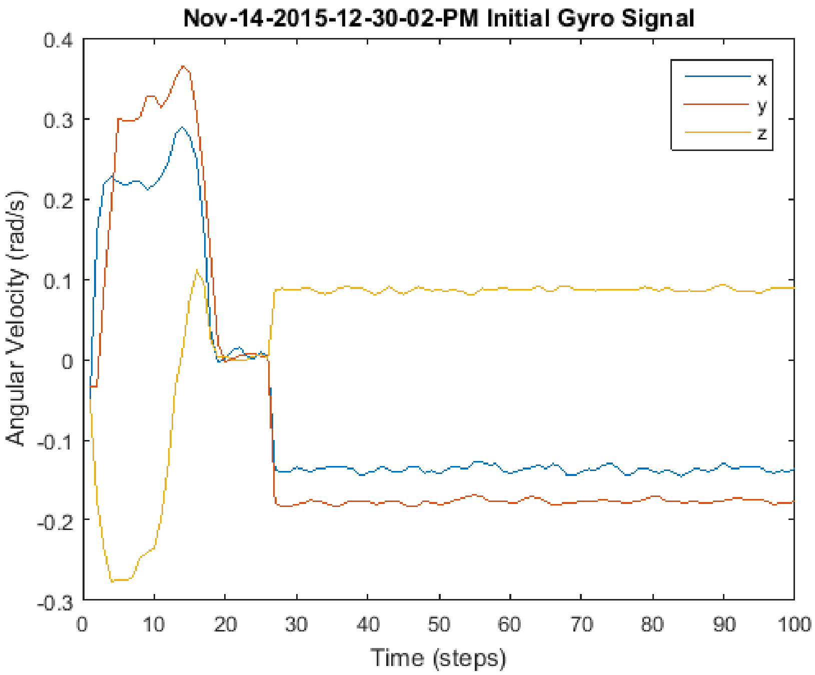


Figure 3.5: Biased gyro measurements from Android API

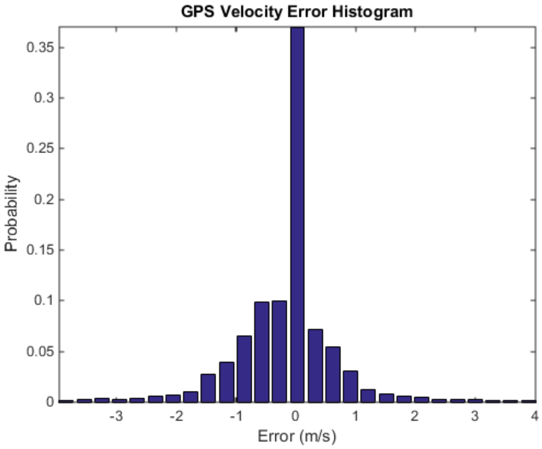


Figure 3.6: GPS Velocity Error Histogram between ±4 $\frac{\text{m}}{\text{s}}$

Sources of error in the speedometer include tire diameter variations from wear, temperature, pressure, vehicle load, and nominal tire size. Vehicle manufacturers typically calibrate speedometers to read high by an amount equal to the average error, to ensure that their speedometers never indicate a lower speed than the actual speed of the vehicle. This way, manufactures are not liable for drivers violating speed limits. Compared to past mechanical systems, digital systems will also have additional error when the final drive gear ratio is changed.

The OBD velocity values were collected from the vehicle's speedometer as integers in $\frac{\text{km}}{\text{h}}$ introducing precision error into the measurement. Figure 3.6 visualizes the probability density function of the MA GPS velocity errors between $\pm 4\,\frac{\text{m}}{\text{s}}$. The error distribution is not Gaussian as seen by the long and heavy tails that extend beyond the domain of the presented graph. The mean of the distribution is $-0.2247\,\frac{\text{m}}{\text{s}}$ with a variance of 0.8892.

## 3.4 2D Extended Kalman Filter

Sensor fusion for this non-linear system can be achieved using an EKF as described in Section 2.2.4. First the state estimate model and the measurement estimate model must be formulated while the same measurement model can be used to calculate the innovations at each time step.

The formulation is similar to Equation 3.1 except for $\omega_t$ and $a_t$. With the use of a mobile device, no information regarding the acceleration and steering can be measured directly. Therefore, the state estimate model in our methodology assumes that $\omega_t$ and $a_t$ are constant. That is,

$$\omega_t = \omega_{t-1} \tag{3.5}$$

$$a_t = a_{t-1} \tag{3.6}$$

Thus, the state estimate update is represented as,

$$f = \begin{bmatrix} x_t \\ y_t \\ \theta_t \\ v_t \\ \omega_t \\ a_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_{t-1}\cos\theta_{t-1}\Delta t + a_{t-1}\cos\theta_{t-1}\frac{(\Delta t)^2}{2} \\ y_{t-1} + v_{t-1}\sin\theta_{t-1}\Delta t + a_{t-1}\sin\theta_{t-1}\frac{(\Delta t)^2}{2} \\ \theta_{t-1} + \omega_{t-1}\Delta t \\ v_{t-1} + a_{t-1}\Delta t \\ \omega_{t-1} \\ a_{t-1} \end{bmatrix} \tag{3.7}$$

The Jacobians are required for the first order approximation in the calculation of the EKF. The Jacobians of the State Estimate Update (Equation 3.7) and Measurement Model (Equation 3.4) are calculated as follows.

$$J_f = \begin{bmatrix} 1 & 0 & (-v\Delta t\sin\theta - a\frac{(\Delta t)^2}{2}\sin\theta) & \Delta t\cos\theta & 0 & \frac{(\Delta t)^2}{2}\cos\theta \\ 0 & 1 & (v\Delta t\cos\theta + a\frac{(\Delta t)^2}{2}\cos\theta) & \Delta t\sin\theta & 0 & \frac{(\Delta t)^2}{2}\sin\theta \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

$$J_h = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \omega & v & 0 \end{bmatrix} \tag{3.9}$$

For the EKF, the process and measurement covariances are required to specify the noise models. The measurement covariances were assumed to be independent and calculated using the sensor noise variances. The process noise was set to an identity matrix and tuned to achieve close estimates to the ground truth without introducing large latencies in the estimates. The process and measurement covariances below were used in the state estimation,

$$R = K_{noise} \begin{bmatrix} 10^{-5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{-5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-9} & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 10^{-5} \end{bmatrix} \tag{3.10}$$

$$Q = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\omega^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_a^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_a^2 \end{bmatrix} \tag{3.11}$$

In Equation 3.10, $K_{noise}$ represents a constant that the velocity, gyroscope, and accelerometer noise is scaled with for testing robustness in simulation. These values were tuned using a tuning process that will be discussed in Section 3.6. The EKF also requires an initial estimate. During simulation, the state is initialized to the ground truth.

## 3.5 Carrot Controller

A controller was designed to compute a suitable control input to the plant to generate simulation data. The carrot controller is a simple control method for a two wheel robot [135], which can be extended to an Ackermann robot. Given a control signal vector with velocity and rotation components, the controller can effectively follow a line to the next waypoint. To simulate a lane following driver, the waypoints are defined by the ends of the lane the driver is following. This controller approach was used to simulate lane following and lane changes. Lanes were assumed to have a lane width of 4 m as observed in the Ontario Provincial Standard Specification in construction specifications for pavement repair [136], recent construction contracts awarded by the Ontario Ministry of Transportation in 2013 [137], and oversize guidelines posted by the Ontario Ministry of Transportation [138].

An illustration of the carrot controller is depicted in Figure 3.7. The controller was colloquially named from the traditional idea of a passenger riding an animal. The rider

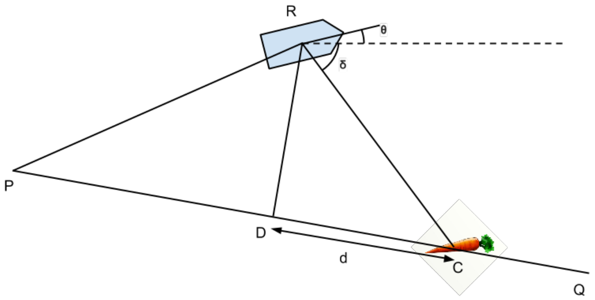


Figure 3.7: Carrot controller

would direct the animal's movements using a carrot-on-a-stick to change the desired heading allowing for the rotation of the animal. Changing the stick distance from the animal allows control of the animal's speed.

To formulate the control model, the 2 waypoints $P$ and $Q$ are defined. $O$ is the origin in a defined coordinate system, $P$ represents the start of the desired path, $Q$ represents the desired destination of the path, $R$ represents the position of the vehicle off-course from the path. Point $D$ represents the closest point on $PQ$ to $R$. The point $D$ is determined using,

$$OD = OP + Proj_{\hat{PQ}}(PR) \tag{3.12}$$

$$Proj_{\hat{PQ}}(PR) = (\hat{PQ} \cdot \vec{PR})\hat{PQ} \tag{3.13}$$

The carrot distance $d$ represents the distance on $PQ$ from $D$ where the carrot should be placed. A more aggressive control can be achieved by decreasing $d$. Cross-track error is a variation of the control method to determine the heading error. However, the carrot distance should be used to ensure the heading is always directed towards Q. To determine the heading error, $\theta_c$ represents the vehicle angular position and the heading error from the reference can be calculated using,

$$\delta_c = \arctan(RC_y, RC_x) \tag{3.14}$$

Finally, the angular velocity control signal can be determined using a simple first order controller.

$$\widetilde{\omega} = k_{rotation} * (\delta_c - \theta_c) \tag{3.15}$$

Since the system model uses a control vector with acceleration and rotation components, the carrot controller was slightly modified to interface with the acceleration control signal. Again a proportional controller was used.

$$\widetilde{a} = k_{acceleration} * (v_{ref} - v) \tag{3.16}$$

## 3.6 2D Simulation Results

Ideally in a linear system, a Kalman filter does not require tuning as the estimates are optimal. However noise from the process, system non-linearities, system model inaccuracies, and measurement noise errors each contribute to non-ideal values of process noise. Additionally, tuning the process noise allows the state estimates to appear smoother. This is particularly useful during segmentation or classification. As mentioned in Section 2.2.4, the process noise was manually tuned to the specified $Q$ matrix.

The tuning methodology involves iteratively changing process noise values on the diagonals and determining parameters that provide suitable state estimates that are close to the ground truth. For the simulation, $K_{noise} = 10$ was used to ensure that drift errors were significant enough for the EKF to correct and that the EKF was robust to higher magnitudes of noise. It is important to ensure the process noise is neither smaller nor larger than an optimal setting. If the process noise is smaller than an optimal value, inaccuracies of the model would be observed in the state estimate. The EKF would weigh the process state update more than measured values resulting in slow convergence to the measurement. This is especially observable in the $a$ and $\omega$ state estimates as the process model assumes values to be constant from the previous state, despite an exogenous control signal. If the process noise were larger than an optimal value, the measurement would be weighted more than required resulting in noisier estimates than if a lower value was used.

Since the model described in Section 2.2.4 assumes constant $a$ and $\omega$ from the previous state, it is critical that the state estimate for these state vector elements converges rapidly. Additionally, any state estimates that depend on the integration of these higher order states would suffer due to higher order integration drift for each level of integration in the system model. As such, tuning was performed sequentially with precedence for higher order states. The tuning was performed in the following order:

1. Linear Acceleration ($a_t$)
2. Angular Velocity ($\omega_t$)
3. Linear Velocity ($v_t$)
4. Angular Position ($\omega_t$)
5. X Position ($x_t$)
6. Y Position ($y_t$)

In order for the process noise to scale with the simulation noise constant, a scaling factor $K_{noise}$ was required to be included in the process noise covariance matrix. The tuned $Q$ matrix was arbitrarily divided by 10 in an attempt to cancel the original noise scaling and multiplied by the $K_{noise}$ variable. It was observed over many trials that the resultant process noise covariance matrix robustly performed well for $K_{noise} \in [1, 1000]$.

Results across different trials using the presented methodology are robust in performance. As an illustration, the filter's result from 1 randomly selected trial is presented in this section. The baselines used for comparison in this analysis are the open loop (OL) values model calculated using velocity and acceleration sensor values through dead reckoning. Both baselines use an OL model for the angular position from gyroscope sensor values. This particular trial used $K_{noise} = 10$ to demonstrate measurement robustness and allow integrative drift from baseline methods to be observable. The simulation was set to calculate state updates at a frequency of 100 Hz. Simulated measurements, including the GPS, were obtained at a rate of 100 Hz as well.

In the simulation, the vehicle's initial state is at the origin heading Eastbound at 5 $\frac{\mathrm{m}}{\mathrm{s}}$ resulting in the state vector $X_0 = [0\ 0\ 0\ 5\ 0\ 0]^T$. Between $t \in [0, 1]$, the vehicle maintains it's heading and velocity. During $t \in [1, 4]$, the vehicle turns at $\frac{\pi}{6} \frac{\mathrm{rad}}{\mathrm{s}}$. At $t \in [4, 6]$, the vehicle maintains the same heading while changing acceleration instantaneously from $a = 0.5 \frac{\mathrm{m}}{\mathrm{s}^2}$ to $a = 1 \frac{\mathrm{m}}{\mathrm{s}^2}$ linearly to simulate acceleration after a turn. At $t \in [6, 15]$, the vehicle uses the Carrot Controller algorithm described in Section 3.5 to enter the passing lane at $x = 14.5$. In this time range, the acceleration was set to a constant 1 $\frac{\mathrm{m}}{\mathrm{s}^2}$. Between $t \in [15, 34]$, the vehicle again uses the Carrot Controller to perform a lane change to the slow lane at $x = 18.5$.

Figure 3.8 provides an overview to illustrate the trajectory of the vehicle's manoeuvres. All estimation methods and the position sensor values have a ticker trajectory with markers appearing at a frequency of 2 Hz. It was observed that OL methods result in considerable

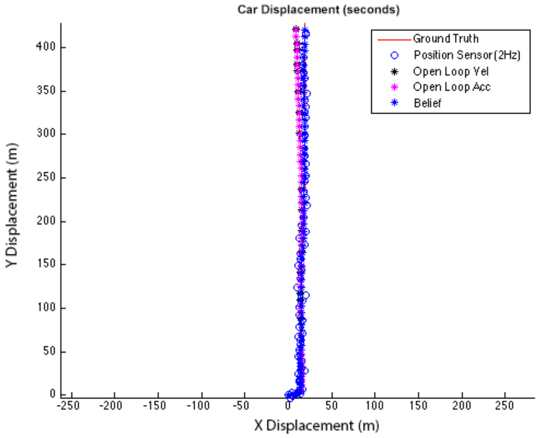


Figure 3.8: Overview of car displacement

drift of the vehicle's state estimate from the ground truth, especially after the lane change manoeuvre. The belief label represents the state estimate from the EKF.

Additional trials were performed to observe the effects when the $K_{noise}$ differed between simulation parameters and the EKF tuning covariances. It was found that differing $K_{noise}$ values showed poor filter tuning which resulted in either slow error convergence with weak covariances or more drift and noise with stronger covariances.

The performance of the linear acceleration state estimation using the EKF is presented in Figure 3.9. It was observed that the state estimate is noisy around constant acceleration. However when acceleration changes instantaneously and linearly, the estimates follow closely and accurately. Figure 3.10 depicts the error associated with the estimates. Spikes were observed in the error graph. Specifically the downward first spike at $t = 400$ is a result of an instantaneous change in acceleration due to the acceleration after the turn. In the range $t \in [4, 6]$, it was observed that the estimated acceleration undershoots the ground truth. This phenomenon is due to the constant acceleration assumption in the state estimation update. The upward second spike at $t = 600$ is due to instantaneous change to 1 $\frac{m}{s^2}$. The error quickly converges to zero after this change. Finally, the upward third spike is associated with the instantaneous change in acceleration at $t = 1500$.

The performance of the angular velocity state estimation using the proposed method is presented in Figure 3.11. It was observed that the state estimate is noisy around constant angular velocity. Despite instantaneous changes in angular velocity during the turn at $t \in [1, 3]$ and complex changes around $t \in [15, 20]$, the state estimates converge quickly to the ground truth after each manoeuvre. Each instantaneous change can be observed in the

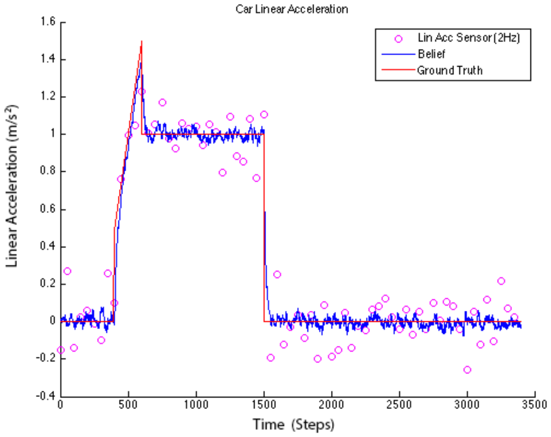


Figure 3.9: Performance comparison of linear acceleration state estimates

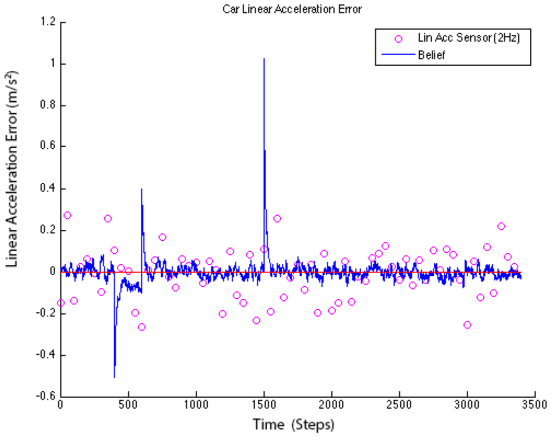


Figure 3.10: Error comparison of linear acceleration state estimates

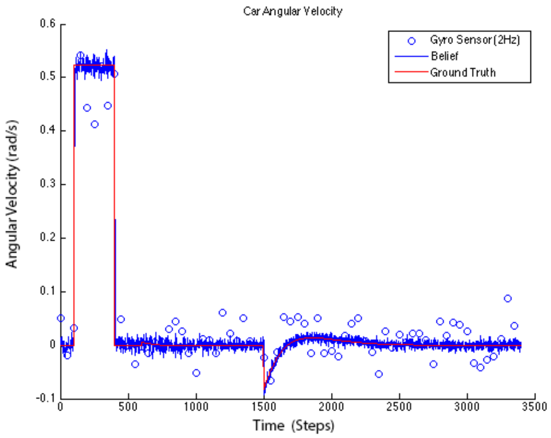


Figure 3.11: Performance comparison of angular velocity state estimates

error graph depicted in Figure 3.12.

The performance of the linear velocity state estimation using the proposed method and OL methods are captured in Figure 3.13. Comparing the state estimates from the EKF and OL acceleration state estimation methods, it was observed that the velocity estimate errors are small in magnitude for both methods. In the simulated noise model, the IMU noise parameters are much lower than the GPS parameters. These parameters result in GPS noise diluting the filter's accuracy in the short term resulting in the OL acceleration estimation being more accurate than the EKF state estimate for most of the observed segment. However as the simulation progresses, the OL acceleration method exhibits considerable drift after $t = 2500$.

Figure 3.14 compares the angular position state estimation performance of the proposed methodology and the OL Gyroscope model. State estimation drift from the OL method can be observed as early as $t = 100$. Large angular position errors are detrimental as heading errors introduce non-linear errors that are difficult to correct. These non-linear errors are difficult to robustly design for in linearized models such as the OL method. It is important to observe that angular position estimate errors converge to zero despite the lack of an angular position measurement.

Figure 3.15 contrasts the two OL methods obtained from the velocity and acceleration measurement with the EKF position state estimation in the x-axis. It was observed that drift errors considerably affect the OL state estimates after $t = 1300$ whereas the proposed methodology is robust to integration drift when estimating position. On the other hand, Figure 3.16 captures drift error as early as $t = 200$ during the turn manoeuvre.

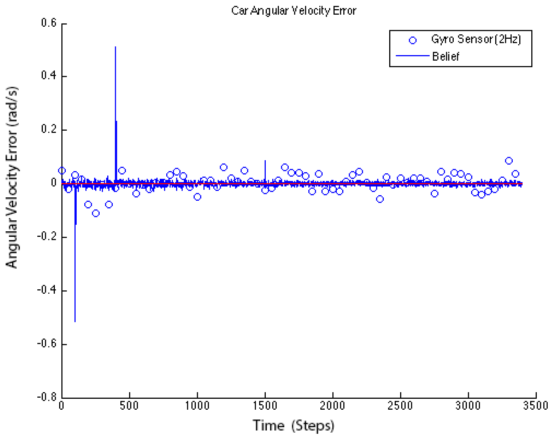


Figure 3.12: Error comparison of angular velocity state estimates

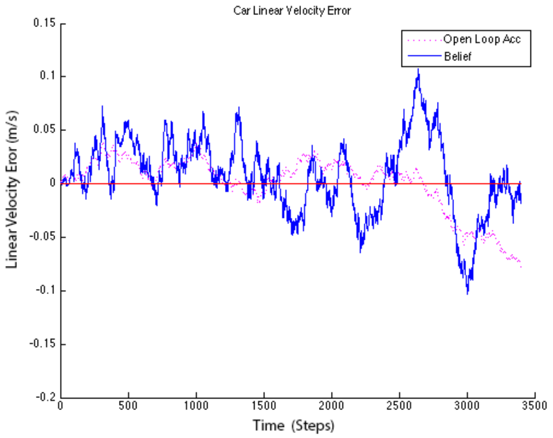


Figure 3.13: Error comparison of velocity state estimates

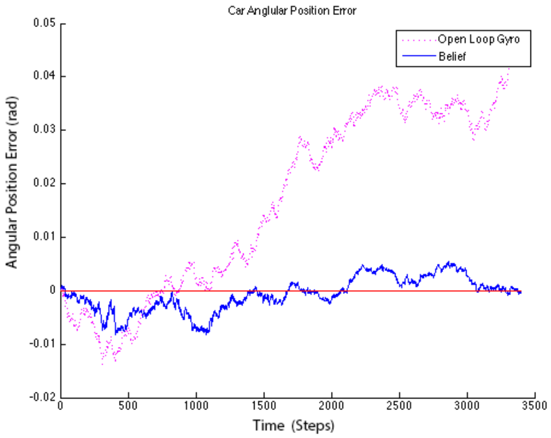


Figure 3.14: Error comparison of angular position state estimates

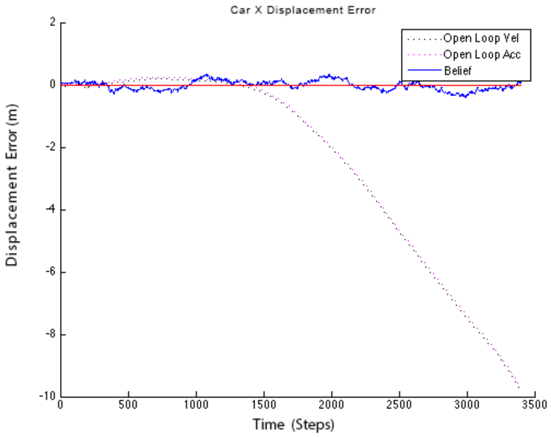


Figure 3.15: Error comparison of X displacement estimates

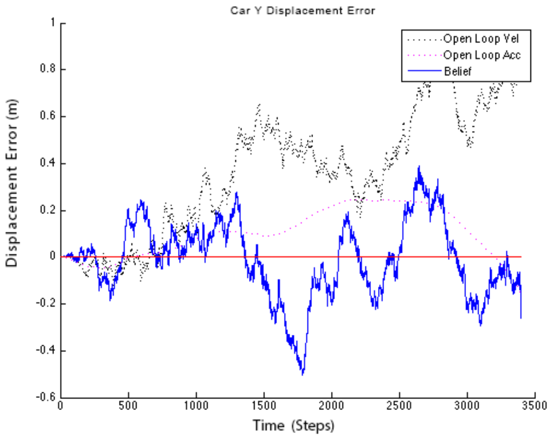


Figure 3.16: Error comparison of Y displacements estimates

## 3.7 2D Non-Simulation Results

To validate the EKF on non-simulation data, the state estimator was applied on the collected trip data. As an illustration, the filter's result from 1 exemplar trip is presented in this section.

The baselines used for comparison in this analysis are the sensor measurements calculated by interpolating measured sensor values. The state estimator was modified to mask observation model rows when measurements are not available. The GPS angular position sensor is less accurate at low velocities due to decreased signal-to-noise ratio (SNR) when the GPS differentiates consecutive position measurements. Additional logic was used to ignore angular position measurements at low velocities. The tuning parameters $R$ and $Q$ used for these results corresponded to their definitions in Equations 3.10 and 3.11 respectively. Since the measurement noise profiles are known, the noise multiplier was set as $K_{noise} = 1$. The accelerometer measurement was omitted from these experiments as it was found that road slant negatively affected the performance of the model.

Figures 3.17 and 3.18 illustrate the filter's performance on the X and Y positions respectively. It can be observed that positioning errors are high near regions when the car decelerates to a stop and when the car accelerates from rest. The car's measured velocity is inaccurate leading to higher position errors. Outside these regions, the position errors are relatively low indicating the position is accurate when the car is moving.

Figure 3.19 depicts the filter's performance on the angular position state. Initial mea-

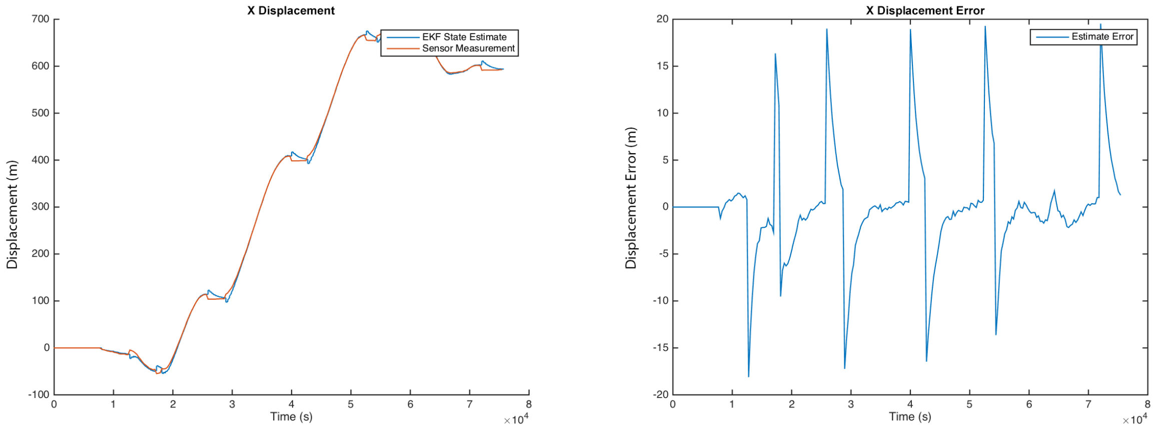


Figure 3.17: X Displacement state for the trial

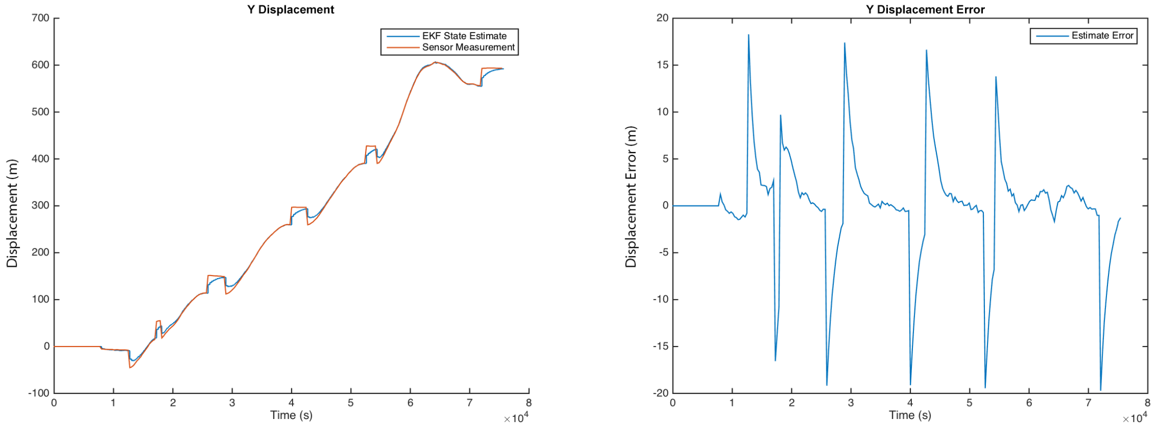


Figure 3.18: Y Displacement state for the trial

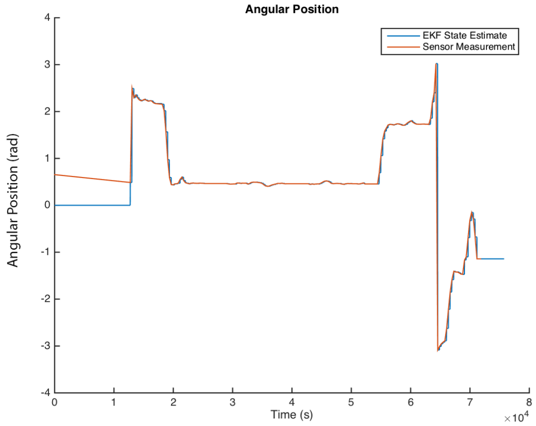


Figure 3.19: Angular Position state for the trial

surements from the GPS are incorrect and interpolated during the GPS fix leading to a poor ground truth reference at the beginning of the trip. The EKF state estimate and sensor measurements overlap for the majority of the trip indicating the filter performs accurately. Due to differing and irregularity update rates between the sensor and the filter, an error plot was not included in these results.

Figure 3.20 shows the filter's performance on the linear velocity state. Low linear velocity errors indicate the state estimator is performing well. Again, initial measurements during the initial GPS fix affects a poor ground truth reference at the beginning of the trip. It was observed that there were high errors before and after the car was stopped. The errors were more pronounced when the car was decelerating to a stop, corresponding to higher position errors as well in the same region. The state estimator believes that the car is decelerating, but does not have embedded logic to realize that the car has stopped decelerating and cannot have negative velocity. The state estimator continues to maintain a belief that the vehicle is still decelerating into the negative regions leading to large linear velocity errors. Due to uncertainty in the covariance matrix, the state estimator is able to quickly converge to the measured velocity.

Figure 3.21 illustrates the state estimator's performance on the linear acceleration state. Since the accelerometer measurement was omitted from these trials, there is no ground truth reference for comparison. In these results, it can be observed that the accelerometer closely corresponds the linear velocity derivative in the estimated state. After the car accelerates from a stop, the acceleration estimate is positive whereas the linear acceleration estimate is negative prior to a stop. During the regions when the car is stopped, the

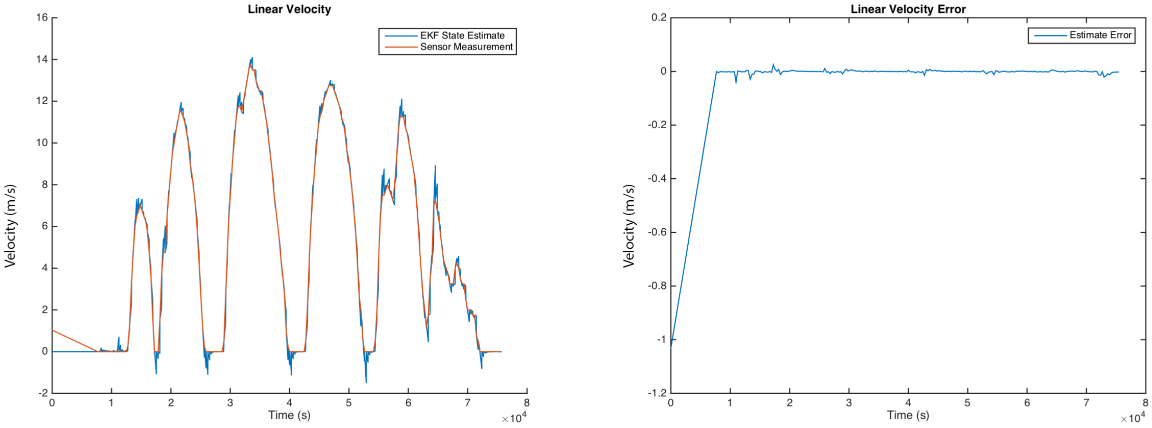


Figure 3.20: Linear Velocity state for the trial

linear acceleration is a positive acceleration corresponding to the filter correcting it's state estimate based on position and velocity between the belief and the GPS measurement.

## 3.8 3D Simulation Results

A simulation was created to estimate the pitch, roll, and yaw offsets of the phone when the car was at rest. The state estimates for pitch, roll, and yaw are detailed in Figures 3.22, 3.23, and 3.24 respectively. As seen in the pitch and roll plots, the states can be estimated without error. However, the yaw offset cannot be estimated and was found to be unobservable when the car was at rest.

## 3.9 Velocity Controller

The state formulation defines the control signal as an acceleration input. However, the decision made by the driver is related to velocity. A proportional controller is used to simulate driver acceleration and braking with a given desired velocity from the data generation process as shown in Equation 3.17. The acceleration ($a$) is calculated as the product of the controller gain $k$ and the difference between the desired velocity ($v_{ref}$) and the current velocity ($v_{l,t}$).

$$a = k * (v_{ref} - v_{l,t}) \tag{3.17}$$

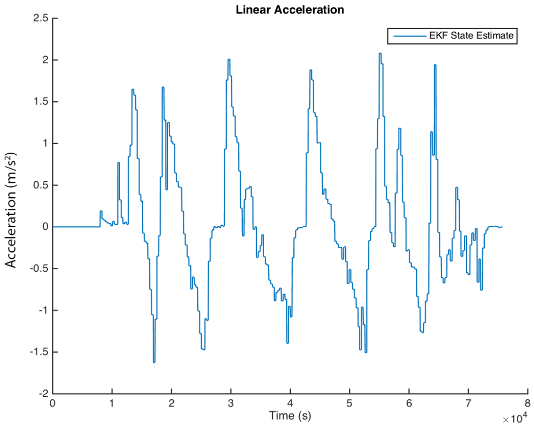


Figure 3.21: Linear Acceleration state for the trial

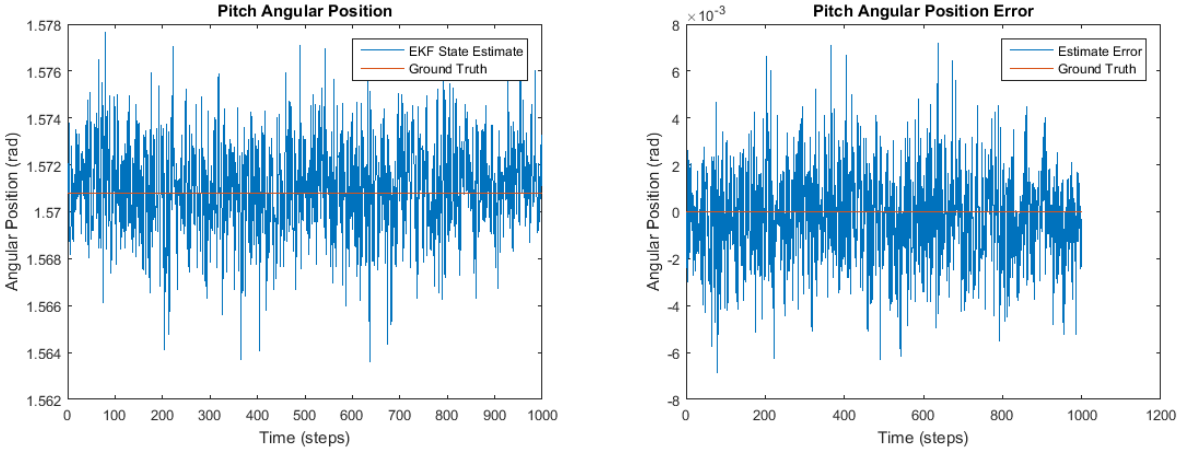


Figure 3.22: Pitch Angular Position state for the trial

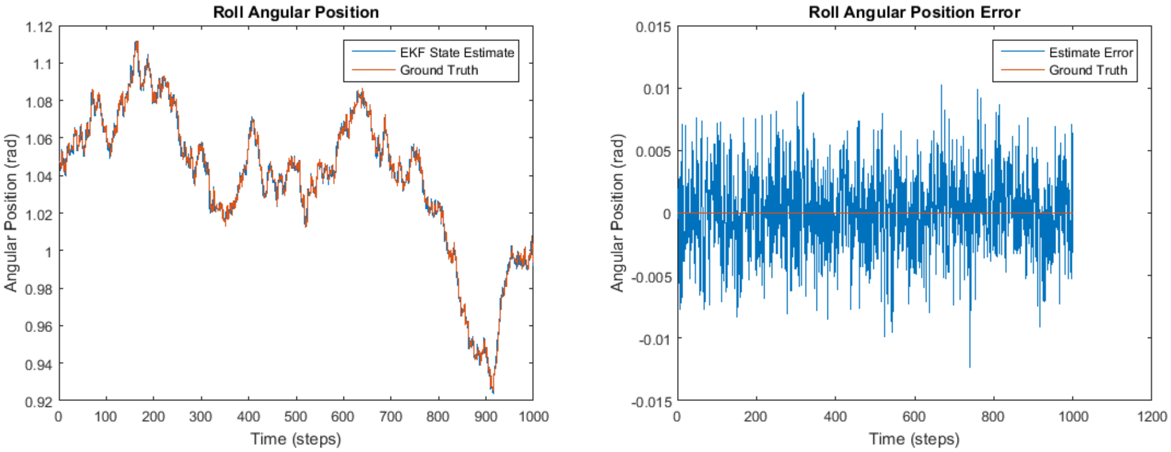


Figure 3.23: Roll Angular Position state for the trial

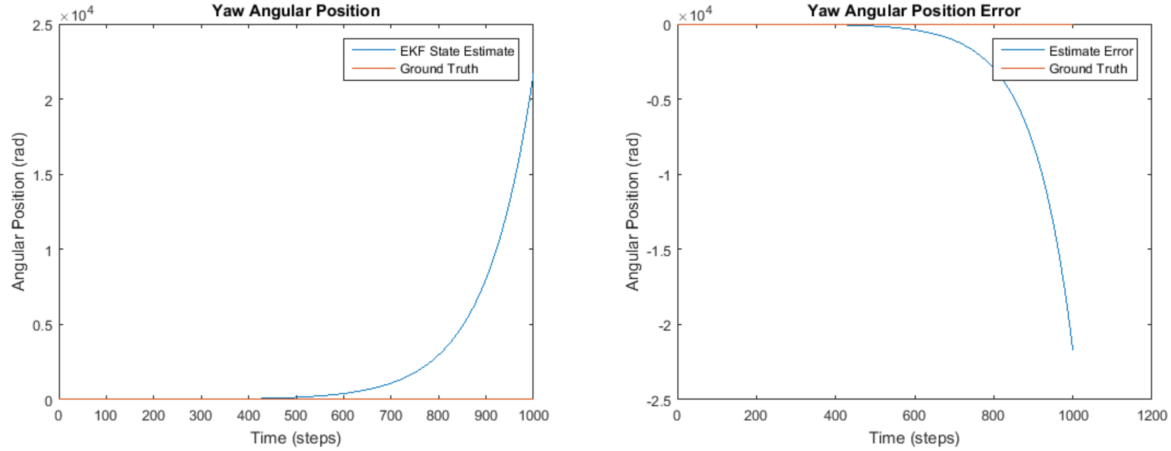


Figure 3.24: Yaw Angular Position state for the trial

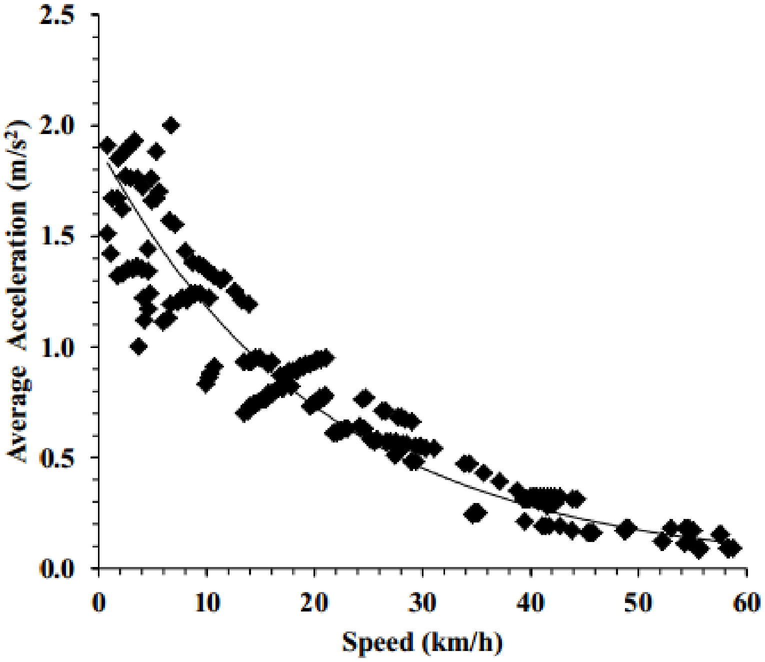


Figure 3.25: Acceleration profile for CS on four-lane divided highway [6]

In simulation, k has different values when accelerating and braking as average acceleration values are different. The acceleration profile is the line of best fit when vehicle acceleration is plotted against different speeds as illustrated in Figure 3.25. One paper from European Transport [6] investigated the acceleration profile on the highway. It is easier to obtain such a profile on the highway as traffic conditions have less of an impact on the acceleration profile. However, it is difficult to obtain such a profile within city limits. The acceleration profile presented in Figure 3.25 was used in this simulation and linearized as $k = 0.1$ (a negative linear relationship) when accelerating instead of an exponentially decreasing profile as described in [6]. On the other hand, an article from Popular Science [140] reported stopping distances result in a deceleration of $-9.81\,\frac{\mathrm{m}}{\mathrm{s}^2}$. These values provide an upper bound for aggressive braking. To simulate non-aggressive braking, a $k = 0.3$ was used during braking.

Table 3.2: Randomized Gaussian PDF parameters

| Random Variable | Mean ($\mu$, $\frac{\text{rad}}{\text{s}}$) | Std Dev ($\sigma$, $\frac{\text{rad}}{\text{s}}$) |
|---|---|---|
| PDF Amplitude | 0.15 | 0.07 |
| PDF Standard Deviation | 0.75 | 0.3 |

## 3.10 Training Data Generation

The simulation emulates a trial of a vehicle in normal traffic conditions for 1500 s (25 min) with a model update rate of 100 Hz totalling 150,000 data points per trial. In a normal driving scenario on the road, the driver's main focus is controlling the steering and acceleration. When driving in a single lane, the steering should centre the car in the lane. However, the acceleration control is situation dependent as the driver needs to decide whether to increase velocity towards the speed limit, control velocity to follow the car in front, or slow to a stop for an intersection or turn. To emulate the transition between the various manoeuvres, the driver's decisions are modelled using a Markov Decision Process (MDP). The speed limit was set to about 15 $\frac{\text{m}}{\text{s}}$ which is equivalent to 54 $\frac{\text{km}}{\text{h}}$. This speed was selected as it was close to Ontario's standard speed limit of 50 $\frac{\text{km}}{\text{h}}$ if no speed limits are posted [139] with an added 10 % tolerance for added speed. The resulting MDP is illustrated in Figure 3.27. This technique to generate the ground truth targets for classifier training has an advantage over training with real data as the ground truth of the manoeuvre is known during the simulation and therefore no manual labelling is required.

Once the manoeuvre is selected, it should be emulated by the car's simulation model. A proportional controller was developed (Equation 3.18) to control the car velocity during lane keeping manoeuvres. The acceleration ($a$) is calculated as the product of the controller gain $k$ and the difference between the desired velocity ($v_{ref}$) and the current velocity ($v$). Past works characterizing lane changes [4] [121] and intersection turns [5] were used to generate open loop signals for respective manoeuvres as seen in Figures 3.28. Control parameters for lane keeping velocities ($v_{ref}$), lane change times ($t_{lane}$), intersection turn times ($t_{turn}$) were randomized over a Gaussian distribution tabulated in Table 3.3. Gaussian noise was added to simulate sensor noise. Road curvature to replicate curvy roads observed in trips was simulated by adding random Gaussian PDFs to non-stop portions of the signal. The simulation updated the model and generated observations at 100 Hz.

A sample of the signal addition is illustrated in Figure 3.26. Centres for the PDF were randomly selected in the angular velocity signal. The standard deviation and amplitudes are then randomized according to the parameters presented in Table 3.2.

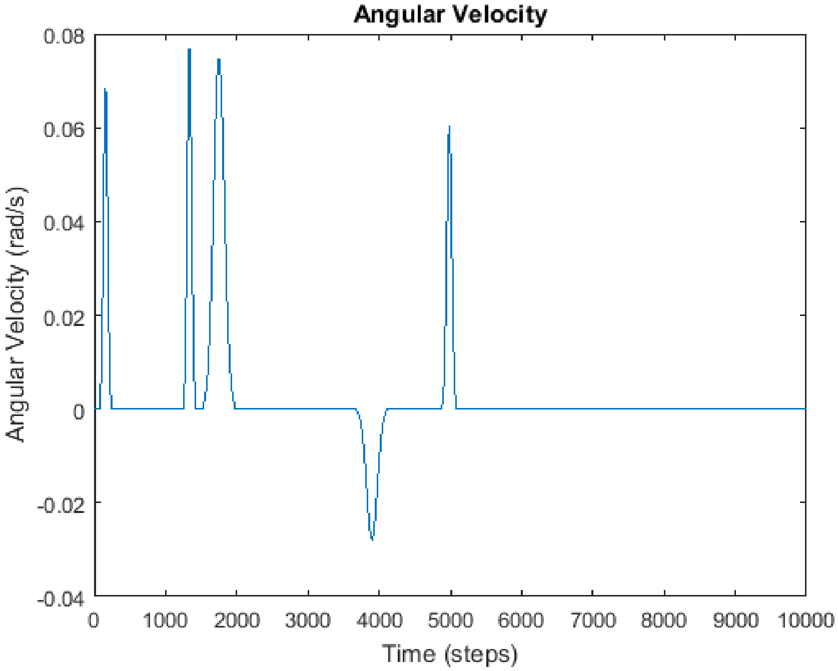


Figure 3.26: Example of Gaussian PDF that is added to the angular velocity signal to simulate road curvature

Table 3.3: Randomized simulation parameters

| Random Variable | Mean ($\mu$) | Std Dev ($\sigma$) |
|---|---|---|
| Speed Limit | 15 | 2 |
| Turn/Lane Change Acceleration | 0.5 | 0.25 |
| Lane Change Velocity | 5.7 | 0.4 |
| Intersection Turn Velocity | 5.7 | 0.4 |

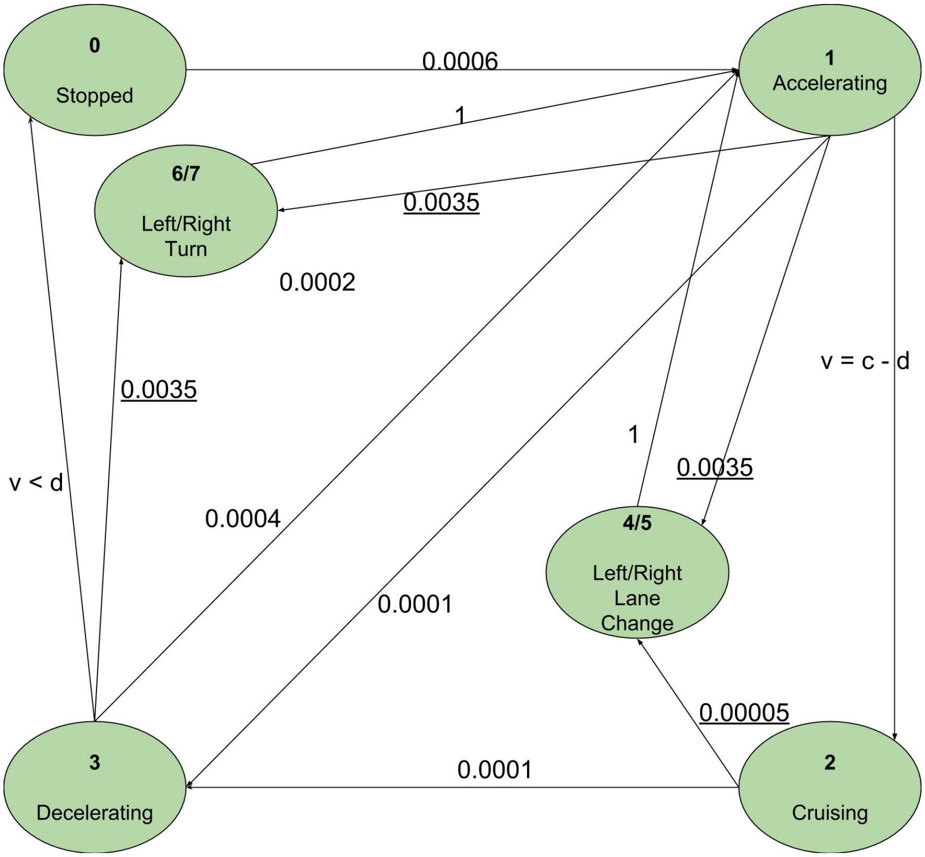


Figure 3.27: Markov Decision Process to simulate driver decisions

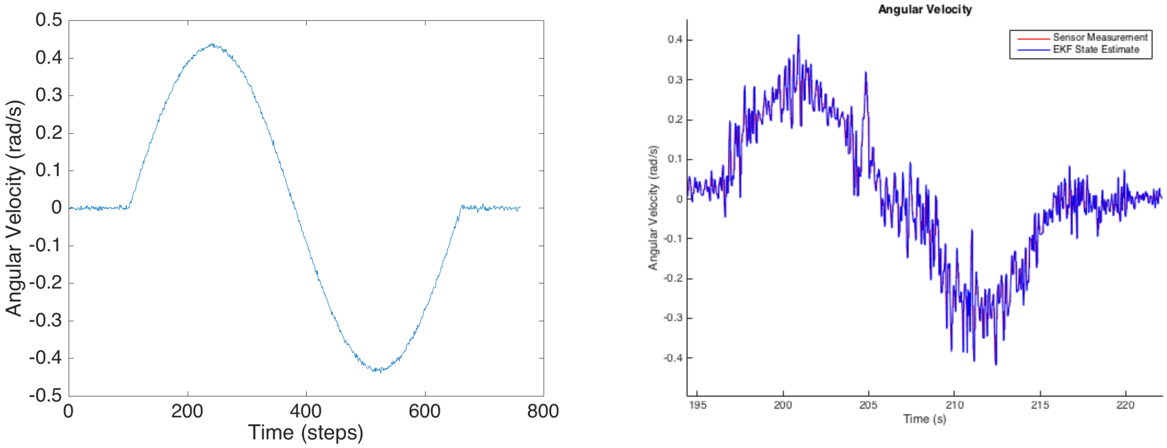


Figure 3.28: Simulated (left) and Observed (right) Lane Change Manoeuvre

$$a = k_v * (v_{ref} - v) \tag{3.18}$$

## 3.11 Observed Manoeuvres

A few manoeuvres have been captured during the trips and are plotted in this section to identify their characteristics. Figure 3.30 illustrates an acceleration and deceleration manoeuvre when the vehicle is coming to and from a stopped state. Figure 3.28 illustrates the sinusoidal angular velocity pattern observed during a lane change.

## 3.12 Summary

In this section, a method for state estimation and simulation data generation was described. The state estimation method demonstrated low errors to the ground truth. Furthermore, the proposed data generation method simulated vehicle kinematics providing ground truth states and measurement data. The next section will apply the simulator and state filter to train a manoeuvre segmentation classifier.

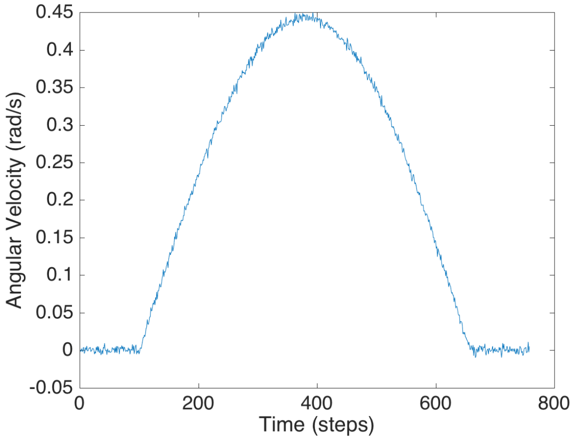


Figure 3.29: Simulated Intersection Turn Manoeuvre

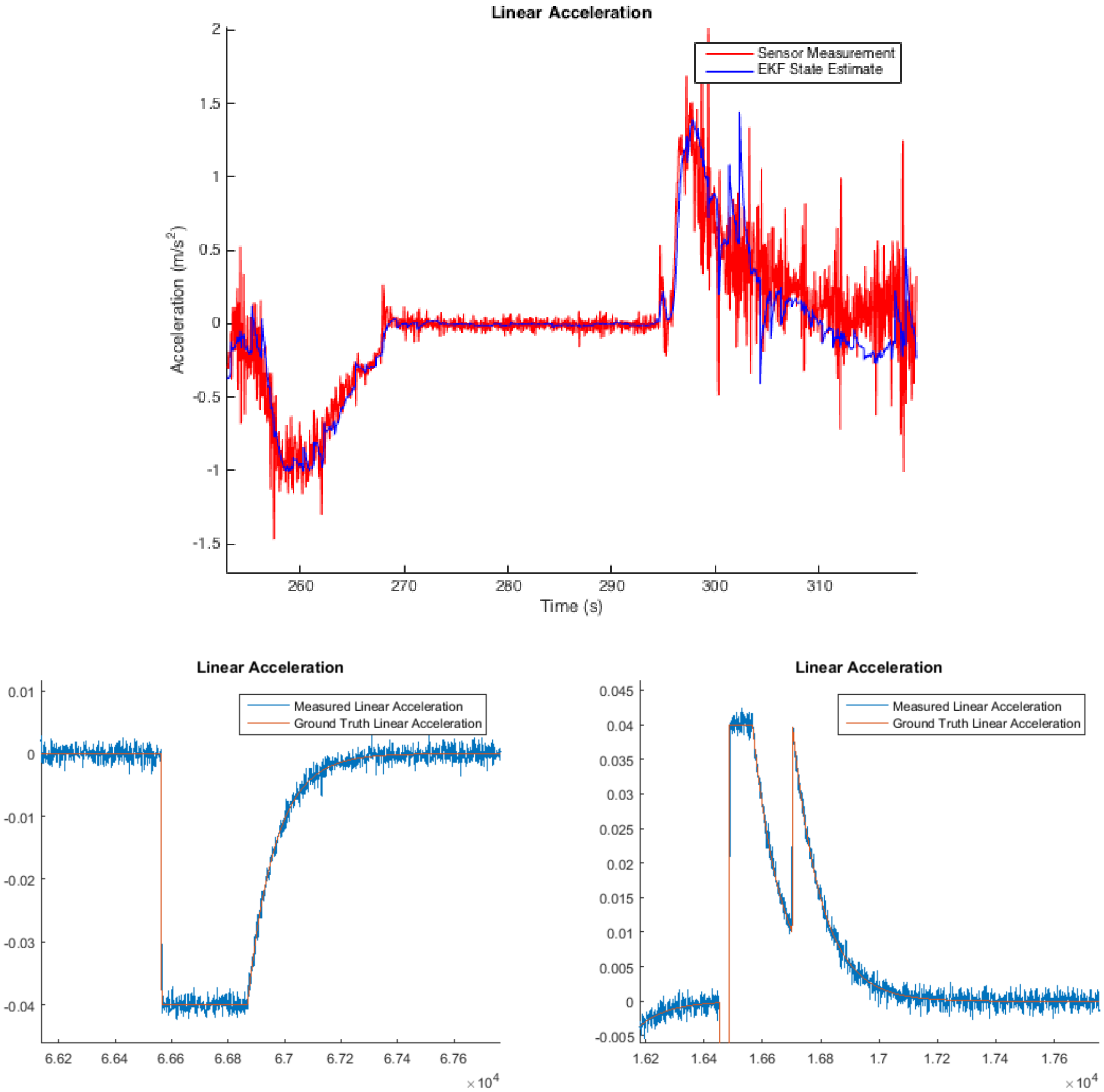


Figure 3.30: Observed Linear Acceleration Near Stops (top), Simulated Linear Deceleration (left), Simulated Linear Acceleration (right)

# Chapter 4

# Manoeuvre Segmentation

In this chapter, a method is proposed to use a simulated car kinematic model as training data for a classifier. A classifier-based approach is proposed for driving manoeuvre recognition from mobile phone data. While a telematic device can connect to the car directly to collect information relating to automobile control (eg. steering, accelerator position, and brake position), it is difficult to install for the average consumer and costly to distribute for companies tracking driver behaviour. Instead, a mobile device can be used to collect kinematic data as it is ubiquitous for an average consumer and contains sensors to accurately estimate a vehicle's trajectory [53].

As described in the previous chapter, a simulation is used to generate classifier training data. Since only simulation data is used, there is no need to collect training data and training data can be generated with minimal computation. To input features for classifier training a sliding window is applied on the simulated sensors and PCA is used for dimensionality reduction. The simulation ground truth states provided supervised training targets to train an SVM. Measurements were performed on a smartphone to collect test data. A novel method applying PCA to the gyroscope signal was used to calibrate the gyroscope's rotation matrix. Our approach allows phone orientation detection in the vehicle.

The remainder of the chapter is organized as follows: A novel method to estimate the phone rotation matrix is described in Section 4.1. The feature selection and reduction processes are described in Section 4.2. Classifier training is highlighted in Section 4.3. Data collection of actual driving sessions is described in Section 4.4 to validate the proposed methodology. The classification results are presented in Section 4.5.

Table 4.1: Principle Component to axis mapping

| PC (#) | Axis | Description |
|---|---|---|
| 1 | z | Yaw |
| 2 | x | Pitch |
| 3 | y | Roll |

## 4.1 Rotation Matrix Estimation

The aim of this research is to identify driver manoeuvres from a mobile device, which cannot be assumed to be located with a fixed and known orientation relative to the vehicle. A novel technique was developed to extract the rotation matrix between the mobile device and the vehicle. PCA was applied to the 3-axis gyroscope signal to calculate the directions of maximum variance in the phone frame.

It was observed that the PCs corresponded to axes describing the motion of the vehicle in the car frame as shown in Table 4.1. For all the trips analyzed, the highest direction of variance, i.e. PC1, corresponded to the yaw axis of the car frame, which was the axis describing turning actions. For the detection of turning manoeuvres, the angular velocity about the yaw axis of the vehicle frame is of interest.

Since PCA produces an orthonormal rotation matrix, the rotation matrix can be directly used as the phone to car rotation matrix. The PCA algorithm produces an ambiguity in terms of direction of the rotation frame [142]. To resolve the ambiguity of the z-axis direction, one accelerometer reading was used to determine the dominant direction corresponding to gravity.

To demonstrate the proposed algorithm, measurements were obtained from a smartphone wedged at known Euler angle offsets in the vehicle during driving. Lane changes, intersection turns, and highway driving were included in this dataset. The phone was placed in a standard flat position to observe if any significant error was introduced using this method for 7 trips. An additional 5 trips with non-zero Euler angles were used to test the effectiveness of this procedure. For the second set of trips, the car frame yaw signals were calculated using the proposed PCA rotation matrix and the traditional accelerometer calibration approach. The performance for an exemplar trip is presented in Figure 4.1 and the full results are detailed in Appendix A. RMS error between the PCA-estimated gyro signal about the vehicle yaw axis and the accelerometer based calibration for all trips was found to be 0.0026 $\frac{\text{rad}}{\text{s}}$ with a SNR of 1538.1. The error statistics are tabulated in Table 4.2. The low error achieved demonstrates this method generalizes across different trips.

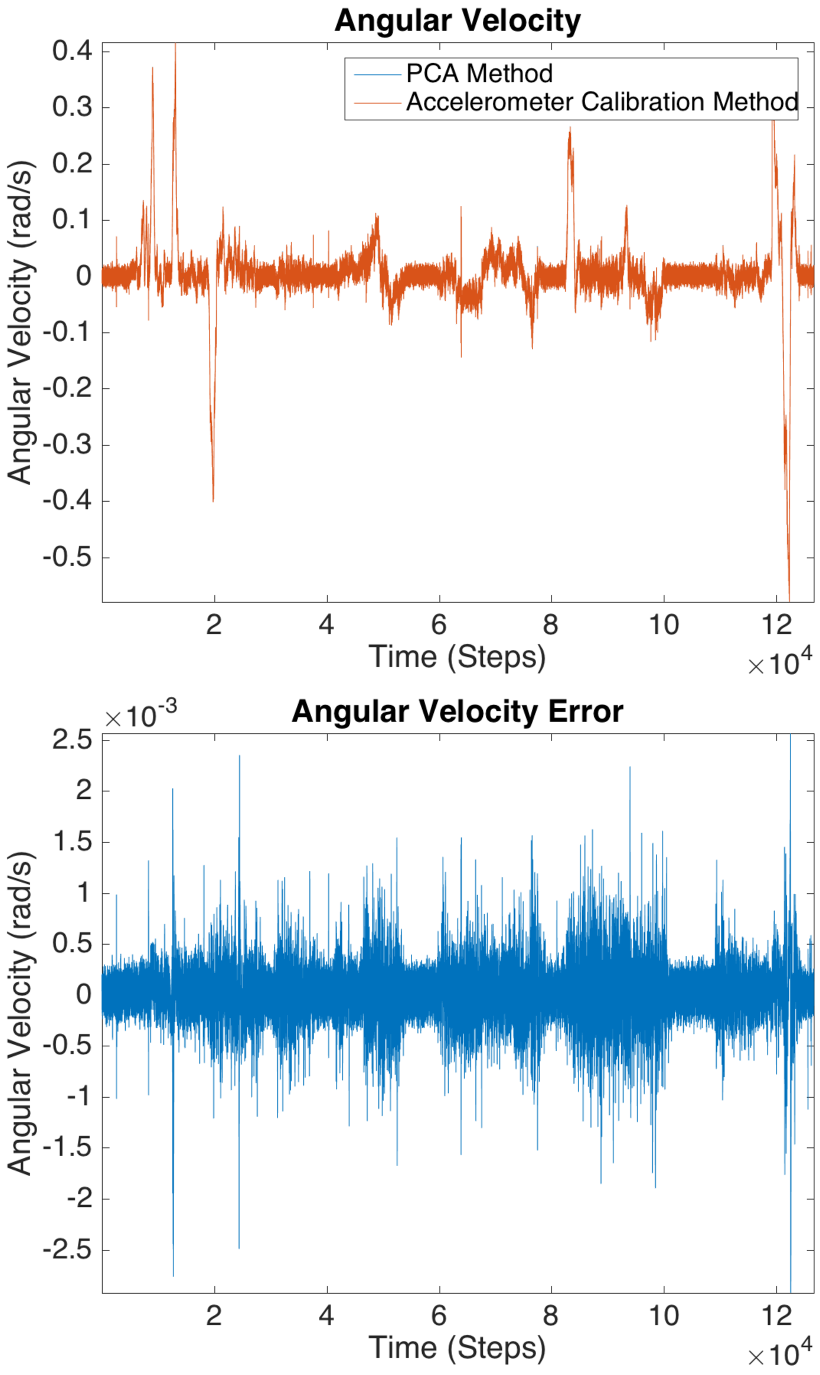


Figure 4.1: Performance (top) and error (bottom) between the PCA rotated gyro signal and the acc rotated gyro signal

Table 4.2: PCA Rotation Matrix Estimation Error Statistics

| | Mean | Min | Max |
|---|---|---|---|
| RMS ($\frac{\text{rad}}{\text{s}}$) | 0.0906 | 0.0373 | 0.1466 |
| RMS Error ($\frac{\text{rad}}{\text{s}}$) | 0.0026 | 0.0017 | 0.0040 |
| SNR | 1.5381e+03 | 219.0401 | 3.1921e+03 |

## 4.2 Feature Selection

To build the feature vector for training and classification, the velocity and angular velocity signals were selected as relevant features that could help to classify the manoeuvres. A sliding window was used on the Gyro PCA signal described in Section 4.1 and the vehicle forward linear velocity signal. A window size variable $r$ was used to denote the window size in time steps as seen in Equations 4.1 and 4.2. To ensure the values from both sensor types are of comparable magnitudes during the variance computation, the data from each sensor were scaled to the maximum measurement value of the entire simulation. The velocity and angular velocity windows are then concatenated and the feature vector for time step $t$ is presented in Equation 4.3.

$$V_t = \begin{bmatrix} v_{t-r} & \dots & v_t & \dots & v_{r+t} \end{bmatrix}^T \tag{4.1}$$

$$\dot{\Theta}_t \begin{bmatrix} \dot{\theta}_{t-r} & \dots & \dot{\theta}_t & \dots & \dot{\theta}_{t+r} \end{bmatrix}^T \tag{4.2}$$

$$F = \begin{bmatrix} V_t / v_{max} & \dot{\Theta}_t / \dot{\theta}_{max} \end{bmatrix}^T \tag{4.3}$$

A sliding window has several advantages over a single measurement. Temporal context of the signal is observable before and after the current time point. More data allows to detect situational context such as lane changes vs road curvature as investigated in [5]. The disadvantage of a sliding window is the high dimensionality and highly correlated features involved with this technique. PCA aims to reduce the dimensionality of these highly correlated features by finding the directions of maximum variance.

The 5th eigenvalue in the scree plot presented in Figure 4.2 was observed to be an inflection point and was used to truncate the signal to the top 5 PCs. The explained variances for the first 5 PCs are 79.95 %, 7.311 %, 6.006 %, 3.967 %, and 1.597 %. The total explained variance of the first 5 PCs is 98.831 %.

The first 5 PC are presented in Figure 4.3. PC1 selects the velocity features, effectively taking the mean of the velocity. PC2 selects the derivative of the velocity in the left half of

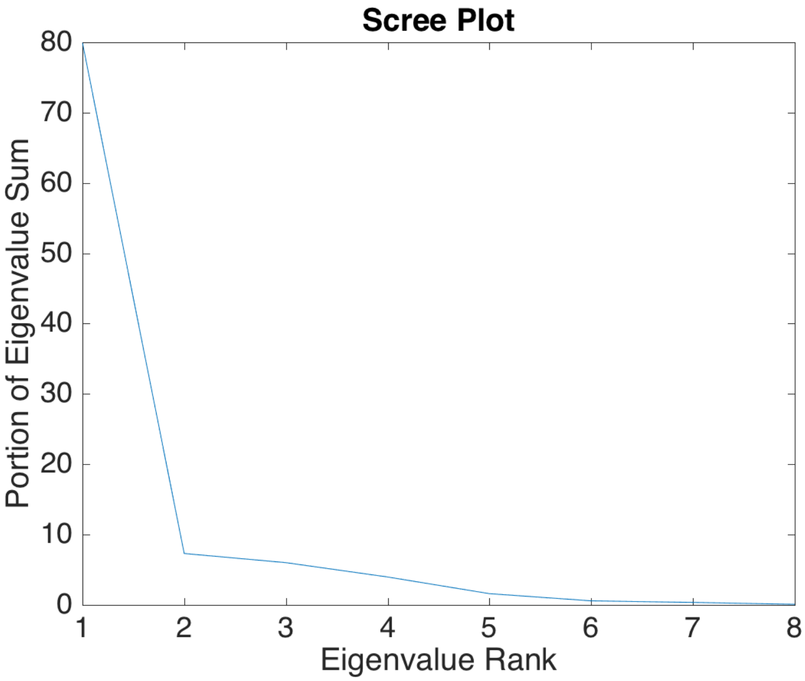


Figure 4.2: Scree plot of feature vector

the features. On the right half, it estimates the average of the angular velocity and a peak at the centre of the right half to represent peaks in the angular velocity signal. PC3-PC5 are higher order derivatives of both velocity and angular velocity signals.

## 4.3 Training

The Libsvm extension for Matlab [143] is used for SVM classifier training. A grid search was used to find the optimal SVM training parameters. SHARCNET[1], was used to train SVMs using parameters linearized by the grid search's bounds. C-Support Vector Classification (C-SVC) was used as the training SVM type and a Radial Basis Function (RBF) kernel was used. Cost, $\gamma$, and sliding window size were varied as tabulated in Table 4.3.

[1] University of Waterloo is a member of SHARCNET (www.sharcnet.ca), a consortium of universities and colleges operating a network of high-performance compute clusters in south western, central and northern Ontario.

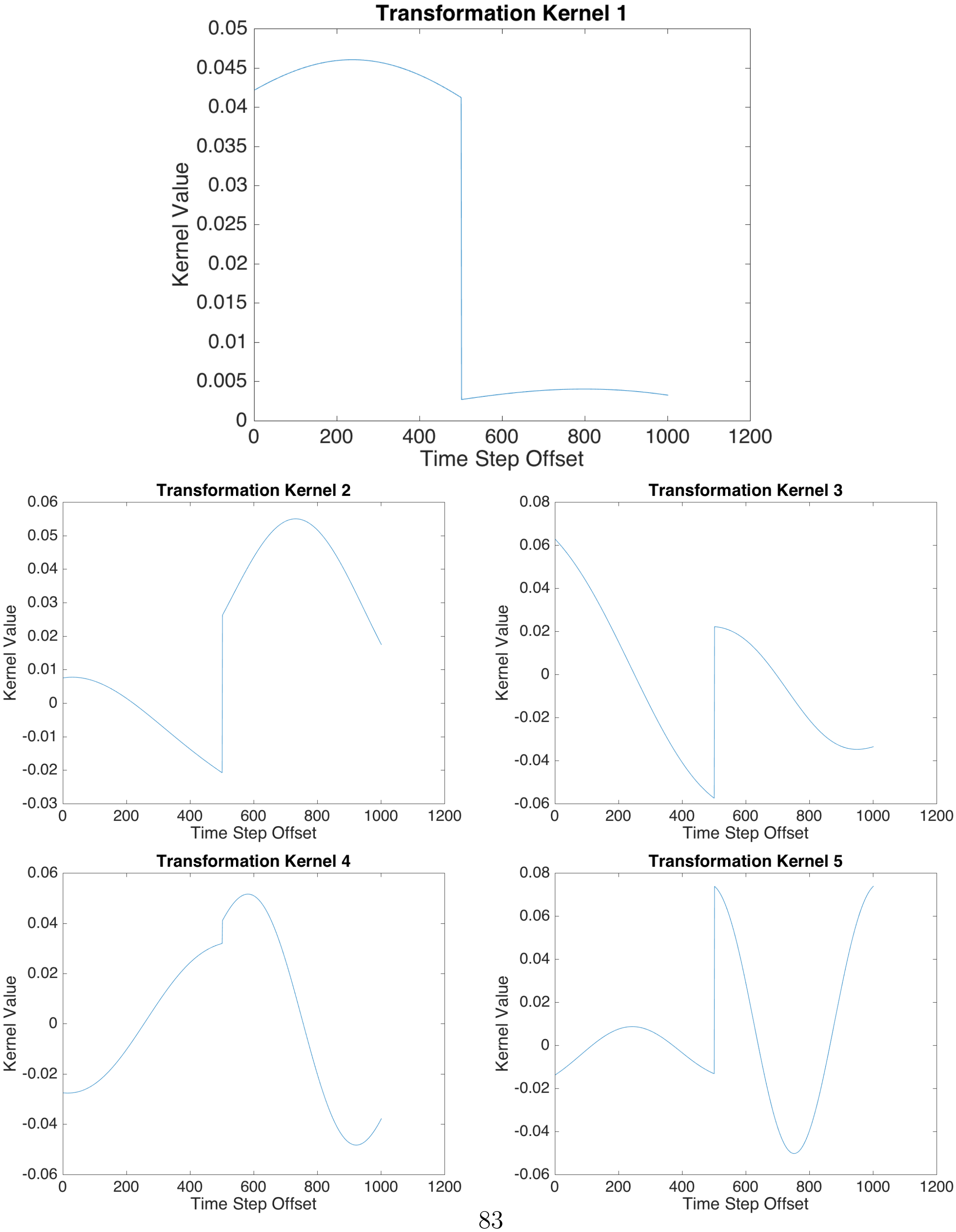


Figure 4.3: Top 5 principle components of feature vector for a 5 second window. The first 500 features correspond to normalized linear velocity and the second 500 features correspond to normalized angular velocity.

Table 4.3: Parameter and respective grid search

| Parameter | Min | Max | Subdivisions | Subdivision Scale |
|---|---|---|---|---|
| Cost | $10^{-7}$ | 1 | 5 | Log |
| $\gamma$ | $10^{-5}$ | 1 | 10 | Log |
| Sliding Window Size | 0.5 | 5 | 5 | Linear |

Table 4.4: Measurement from the CarRecorder mobile app

| Measurement | Sensor | Max Update Rate (Hz) |
|---|---|---|
| Latitude | GPS | 1 |
| Longitude | GPS | 1 |
| Speed | GPS | 1 |
| Heading | GPS | 1 |
| Acceleration | Accelerometer | 100 |
| Angular Velocity | Gyroscope | 100 |
| Magnetic field | Magnetometer | 100 |

## 4.4 Test Data Collection

The proposed approach is tested with data collected from driving experiments. To collect data during driving, the Android app described in Section 3.2 was used to collect GPS and IMU data for the presented results in this thesis. The measurement types are listed in Table 4.4.

In this trial, 12 trips were analyzed with one driver using one vehicle. The trips include straight and curved roads, busy and light traffic, city and suburban driving conditions. The total length of all trips combined was 53 minutes. The phone was placed in the vehicle cupholder during the data collection. The car started to operate at least 5 seconds after a GPS fix ($<$ 10m accuracy) was obtained for each trip. Since the reported update rates are the sensor's maximum rates, sensor updates vary and are not at a constant interval. To maintain a constant measurement interval and consistency of sample rates during training, the measurements were re-sampled at 100 Hz and linearly interpolated.

## 4.5 Manoeuvre Classification Results

The classifiers were trained using the methodology described in Section 4.3 and were tested on the datasets collected in Section 4.4. Table 4.5 tabulates classifiers with the top

Table 4.5: Turn and Straight Manoeuvre Classification Performance. Here, the top results from the grid search are presented in descending $F_1$ score. The grid search parameters are SVM cost ($C$), SVM RBF parameter ($\gamma$), and window size ($r$). Performance metrics are average precision ($P$), average recall ($R$), average balanced accuracy ($BA$), and average $F_1$ score ($F_1$).

| $F_1$ | BA | P | R | C | $\gamma$ | r (s) |
|---|---|---|---|---|---|---|
| 0.8194 | 0.8874 | 0.8158 | 0.8279 | 1 | 0.021544 | 1.5 |
| 0.8018 | 0.8655 | 0.8417 | 0.7904 | 0.017783 | 0.077426 | 2.5 |
| 0.7867 | 0.8671 | 0.8072 | 0.7978 | 1 | 0.00046416 | 2.5 |
| 0.7854 | 0.8624 | 0.7958 | 0.7913 | 1 | 0.021544 | 2 |
| 0.7850 | 0.8651 | 0.8028 | 0.7940 | 1 | 0.27826 | 1 |
| 0.7758 | 0.8566 | 0.8176 | 0.7799 | 1 | 0.0016681 | 3.5 |
| 0.7758 | 0.8505 | 0.8301 | 0.7691 | 1 | 0.0016681 | 5 |
| 0.7746 | 0.8493 | 0.8322 | 0.7674 | 1 | 0.0059948 | 5 |
| 0.7744 | 0.8575 | 0.7937 | 0.7824 | 0.017783 | 0.27826 | 1 |
| 0.7742 | 0.8550 | 0.8160 | 0.7764 | 1 | 0.077426 | 2 |

10 $F_1$ scores including parameters and performance. The performance metrics in the table are a mean of the metric aggregate for all manoeuvres.

The top performing classifiers had cost parameters between 0.017783 and 1 with a mean of 0.8035. The classifiers had $\gamma$ parameters in the magnitudes $10^{-3}$ to $10^{-1}$ with a mean of 0.0764. It was observed that the top classifiers generally worked better when the range was smaller than 2.5 s. The window size of the top classifiers had a mean of 2.6 s and a median of 2.25 s.

Furthermore, it was observed that the addition of the Gaussian noise to simulate road curvature improved the top classification. In an alternate set of training parameters performed, $F_1$ scores achieved increased from 0.79 to 0.8194.

Our proposed method achieves an average $F_1$ score of 0.8194 across all manoeuvres. To analyze the results of the unbalanced testing dataset, the average balanced accuracy was also calculated and achieved 0.8874. Finally, precision and recall performance was 0.8158 and 0.8279 respectively. These results demonstrate the proposed approach is able to achieve good results.

The confusion matrix for the top classifier is illustrated in Table 4.6. In the table, the rows are the ground truth and the columns are the classes predicted by the classifier. Recall rates for turns were found 0.8739 for left turns and 0.8517 for right turns. These results

Table 4.6: Confusion Matrix of the optimal classifier

| | Stop | Acc | Dec | L Turn | R Turn |
|---|---|---|---|---|---|
| Stop | 0.9995 | 0.0002 | 0 | 0 | 0.0003 |
| Acc | 0.0803 | 0.6651 | 0.1868 | 0.0193 | 0.0486 |
| Dec | 0.0639 | 0.1405 | 0.7494 | 0.0328 | 0.0133 |
| L Turn | 0.0369 | 0.0422 | 0.0388 | 0.8739 | 0.0083 |
| R Turn | 0.0362 | 0.0415 | 0.0700 | 0.0006 | 0.8517 |

achieve similar performance when compared with [96] and [38], which use non-simulation data to train their classifiers. Despite obtaining a lower recall rate compared to [5], their study may not generalize well as only 2 routes were used, which may have led to overfitted time and signal thresholds. In this study, each of our tests used different routes providing results for more generalized naturalistic driving.

It was observed that the classification performance for each manoeuvre was sensitive to the $\gamma$ parameter. From the trained classifiers, it was observed that top straight manoeuvre classifiers used a larger $\gamma$ $(10^{-3} - 1)$ and a larger window $(1 - 2.5$ seconds). Intersection turns were observed to perform better with a lower $\gamma$ $(10^{-5} - 10^{-3})$ and a smaller window $(0.5 - 2$ seconds).

The different $\gamma$ value leads to the classifier performing well on one type of manoeuvre but not the other. This trend as well as the range of $\gamma$ parameters in the overall top performers indicate the performance metric is prioritizing straight manoeuvres over turn manoeuvres. This explains the confusion between intersection turn and straight manoeuvres.

A second observation is that acceleration and deceleration are confused with stop manoeuvres. When reviewing the labelled data, the confusion with these labels were found to be near transitions between accelerating, decelerating, and when the car shifts forward near an intersection.

Finally, it was observed that left turns consistently achieved better classification performance than right turns. Despite higher angular velocity amplitudes, left turn manoeuvres tend to take longer to execute allowing them to be more temporally distinguishable.

## 4.6 Summary

In this section, the manoeuvre segmentation training and validation process was described. A novel technique for rotation matrix estimation demonstrated the yaw gyro is extracted by applying PCA on the 3-axis gyroscope signal. Linear and angular velocity states were selected as features and PCA was used for feature reduction by truncating using the top 5 PCs. The classifier training on simulated data and data collection process for validation was described. The smartphone was used to collect measurement data in the car. The classification result was presented and grid search was employed for parameter optimization.

# Chapter 5

# Conclusions and Future Work

## 5.1 Conclusions

This thesis proposes a vehicle state estimation, phone rotation estimation, and manoeuvre segmentation classifier for analyzing driving behaviour using smartphone sensors. An EKF was proposed for vehicle state estimation using measurements from smartphone sensors and the accuracy was validated in simulation. A method for rotation matrix estimation was proposed using PCA on the gyro signal. Each PC corresponded to a separate axis in the vehicle reference frame. The method was further validated on driving sessions by placing the phone in different known orientations. A manoeuvre segmentation classifier was proposed using multi-class SVMs. Validation was performed using data collected from driving sessions.

### 5.1.1 Vehicle State Estimation

With recent technological advances in GNSS and mobile device technologies, more accurate navigation and state estimation can be achieved. In Chapter 3, a vehicle state estimation approach was proposed using inexpensive sensors. It was demonstrated that low-cost state estimation is achievable and the methodology was further verified in simulation to ensure robustness and reliability. This approach is advantageous as it provides a non-intrusive state estimate technique and leverages the availability of mobile devices, while reducing sensitivity to sensor noise. The estimation approach allowed for the noise model to be adjusted by using a noise multiplier on a subset of sensors. A magnitude increase in the standard deviation of the sensor noise model demonstrated the robustness of the approach. The magnetometer was investigated as a potential angular position sensor. However the mobile device magnetometer sensor proved to be imprecise and susceptible to electrical noise and bias.

### 5.1.2 Manoeuvre Classification

In Chapter 4, a method was proposed to use a 2D car kinematics simulation as training data for a manoeuvre classifier. Using this data, a sliding window and PCA was used to obtain input features for classifier training. The simulation ground truth states provided supervised training targets to train an SVM, removing the need for costly training data collection and tedious and error prone manual labelling. A grid search was employed for parameter tuning. Arbitrary phone orientations were accounted for by rotating the gyroscopic signals into PCs corresponding to the principal directions of turning.

The scree plot indicated the top 5 PCs should be used when using PCA for a 5 s window. The classifier was found to have a 0.8158 average precision rate and a 0.8279 average recall rate across all manoeuvres resulting in an average $F_1$ score of 0.8194 on the dataset. Furthermore, promising accuracy rates were achieved when classifying multi-class manoeuvres using only simulated data for training. Simple features such as velocity and angular velocity are useful for identifying driving manoeuvres. These results improves manoeuvre segmentation techniques to support driver behaviour analysis.

## 5.2 Future Work

### 5.2.1 Vehicle State Estimation

There are a few limitations to the proposed methodology. To account for sensor intermittency, a family of measurement update equations need to be created for each sensor instead of using one large update matrix for all sensors. This can be achieved by masking rows in the EKF measurement model pertaining to the available measurements at a given time.

Since the desired application requires the use of a mobile phone to obtain measurements, another limitation of the methodology is the assumption that the sensors are at the car's CM in a reference frame aligned with the car. If the device is rotated or translated by the user, the measurements would not correspond to the measurement model proposed.

Furthermore, the problem is a 3 dimensional problem, which was simplified to a 2 dimensional model. In reality, states such as elevation changes and rotations about the pitch and roll axes would reflect the measurements.

To correct for these phone rotations, measurements need to be relative to the phone's coordinate frame. This correctional rotation is achieved by re-estimating the phone rotation matrix or incorporating rotation angle offsets to the model's states. Positional offset can be corrected by either assuming negligible measurement model errors due to sensor proximity to the CM or by again incorporating these offsets into the model.

Finally, a Two Wheel Robot model was used for the proposed methodology's model. To fully model rotational kinematics, a bicycle (described in Section 2.2.1) or Ackermann robot model would be a more accurate representation. If errors are demonstrated to be negligible, then the simpler Two Wheel Robot model can be used.

There are a few aspects of the methodology that should be explored as well. It was assumed that the measurement noise were fully independent. However, certain noise sources are covariate such as acceleration on velocity and velocity on position. On the other hand, the process noise can be better estimated or automatically tuned. A more accurate lane following and lane changing model should be investigated to better represent the driver behaviour in simulation. Finally, position data implies that geolocation data are available. A Bayesian search approach using map matching can help determine probable speeds and angular positions that the vehicle is maintaining to reduce noise.

This thesis uses a simulation model of driver behaviour to train the classifier. Classification errors for real trip data may originate either from the simulation model's accuracy or from the feature selection and classification approach. A more detailed evaluation of the simulation would measure the error contributed by each step in the proposed method. The simulation model accuracy can be evaluated by comparing sensory data during manoeuvres between the simulation and trips. Comparing characteristics such as radius of curvature for turns and velocity profiles during manoeuvres can indicate the simulation's accuracy. Furthermore, the classifier performance can be compared when training with trip and simulated data.

### 5.2.2 Manoeuvre Classification

There are limitations to using the PCA method to estimate the directions of turning for the gyroscope. The proposed method allows the mobile phone to be placed in an arbitrary position and orientation with respect to the vehicle, but assumes that the phone remains in a constant pose with respect to the vehicle for the duration of the trip. If the phone moves during the trip, the principal directions of turning would need to be re-estimated.

A few aspects of the proposed algorithm can be improved. The simulator may increase transition probabilities between manoeuvres to simulate velocity peaks and generate training data for manoeuvres that occur in quick succession. Trip data may also be used to train the classifiers as a baseline comparison. The method should also be more extensively tested with data from multiple drivers with different driving styles.

Libsvm contains additional parameters to tune the classifier training. For instance, varying the weighted costs of each class when training the SVM in the Libsvm package may improve performance and mitigate the unbalanced data problem mentioned in Section 4.5. Additional features can also be added to evaluate if they improve on these results. Features such as radius of curvature can be calculated [96] and input as a feature to the SVM. Another potentially useful feature is the magnetometer sensor which could be combined with velocity and gyroscope similar to [38].

Highway and non-highway driving behaviours are different according to [17]. The work investigates driver behaviour classification at different speeds. Training a highway vs. non-highway classifier may support better distinguishing between these two scenarios. As mentioned in their work, on- and off-ramp behaviour should also be analyzed.

Finally, the feasibility of using hierarchical classifiers or ensemble methods can be investigated. Hierarchical classifiers may account for any behavioural changes between scenarios. Furthermore, ensemble methods may be tuned for specific manoeuvres allowing certain classifiers to specialize in a subset of manoeuvres (i.e. turning or straight manoeuvres).

### 5.2.3 Power and Memory

The results reported in this thesis were collected and processed offline to leverage higher computational power. Thus, the power and memory requirements to implement the proposed method on a smartphone were not analyzed. For classification labelling with kernel SVMs, the computational complexity is linear with respect to the number of support vectors. Tuning the slack variable decreases the number of support vectors and will also decrease the computational requirements respectively. Further analysis should be performed to determine how the proposed approach could be implemented on the smartphone and the corresponding power and memory requirements.

# Bibliography

[1] A. M. Dujon, R. T. Lindstrom, and G. C. Hays, "The accuracy of Fastloc-GPS locations and implications for animal tracking," *Methods in Ecology and Evolution*, vol. 5, no. 11, pp. 1162–1169, 2014.

[2] S. L. Waslander, "ME 597 Lecture Set 3 Motion Modelling," 2013.

[3] S. Rezaei and R. Sengupta, "Kalman filter-based integration of DGPS and vehicle sensors for localization," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 6, pp. 1080–1088, 2007.

[4] W. van Winsum, D. de Waard, and K. A. Brookhuis, "Lane change manoeuvres and safety margins," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 2, no. 3, pp. 139–149, sep 1999. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S136984789900011X

[5] D. Chen, K.-t. Cho, S. Han, Z. Jin, and K. G. Shin, "Invisible Sensing of Vehicle Steering with Smartphones," *Mobile Systems, Application and Services*, pp. 1–13, 2015.

[6] A. Mehar, S. Chandra, and S. Velmurugan, "Speed and Acceleration Characteristics of Different Types of Vehicles on Multi-Lane Highways," *European Transport International Journal of Transport Economics, Engineering and Law*, vol. 1, no. 55, pp. 1–12, 2013.

[7] A. R. Jiménez, F. Seco, C. Prieto, and J. Guevara, "A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU," *WISP 2009 - 6th IEEE International Symposium on Intelligent Signal Processing - Proceedings*, pp. 37–42, 2009.

[8] D. Tedaldi, A. Pretto, and E. Menegatti, “A Robust and Easy to Implement Method for IMU Calibration without External Equipments,” in *IEEE Internation Conference on Robotics and Automation (ICRA)*, Hong Kong, China, jun 2014.

[9] W. T. Fong, S. K. Ong, and A. Y. C. Nee, “Methods for in-field user calibration of an inertial measurement unit without external equipment,” *Measurement Science and Technology*, vol. 19, no. 8, pp. 1–11, 2008. [Online]. Available: http://stacks.iop.org/0957-0233/19/i=8/a=085202

[10] J. Paefgen, F. Kehr, Y. Zhai, and F. Michahelles, “Driving Behavior Analysis with Smartphones : Insights from a Controlled Field Study,” 2012.

[11] Transport Canada, “Road Safety in Canada,” 2011.

[12] “Canadian Motor Vehicle Traffic Collision Statistics 2012,” Transport Canada, Tech. Rep., 2014.

[13] A. Cohen and L. Einav, “The Effects of Mandatory Seat Belt Laws on Driving Behavior and Traffic Fatalities,” *Review of Economics and Statistics*, vol. 85, no. 4, pp. 828–843, 2003. [Online]. Available: http://www.mitpressjournals.org/doi/abs/10.1162/003465303772815754

[14] D. Glassbrenner, “Estimating the lives saved by safety belts and air bags,” in *18th International Technical Conference on the Enhanced Safety of Vehicles*, Nagoya, 2003, pp. 1–5. [Online]. Available: http://scholar.google.com/scholar?hl=en{&}btnG=Search{&}q=intitle:Estimating+the+lives+saved+by+safety+belts+and+air+bags{#}0

[15] N. Brundin and G. Samuelsson, “The Future of Active Safety,” Arthur D Little, Tech. Rep., 2014.

[16] H. Eren and S. Makinist, “Estimating driving behavior by a smartphone,” *... Vehicles Symposium (IV), ...*, no. 254, pp. 234–239, 2012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6232298

[17] J.-H. Hong, B. Margines, and A. K. Dey, “A Smartphone-based Sensing Platform to Model Aggressive Driving Behaviors,” in *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI ’14. New York, NY, USA: ACM, 2014, pp. 4047–4056.

[18] S. Sarker and J. Wells, "Understanding mobile handheld device use and adoption," *Communications of the ACM*, vol. 46, no. 12, pp. 35–40, 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=953484

[19] P. Händel, J. Ohlsson, and M. Ohlsson, "Smartphone Based Measurement Systems for Road Vehicle Traffic Monitoring and Usage Based Insurance," *IEEE Systems Journal*, vol. 8, no. 4, pp. 1238–1248, 2013. [Online]. Available: http://www.diva-portal.org/smash/record.jsf?pid=diva2:665980

[20] P. Desyllas and M. Sako, "Profiting from business model innovation: Evidence from Pay-As-You-Drive auto insurance," *Research Policy*, vol. 42, no. 1, pp. 101–116, feb 2013. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S004873331200145X

[21] D. Karapiperis, B. Birnbaum, A. Brande, S. Castagna, A. Greenberg, R. Harbage, and A. Obersteadt, "Usage-Based Insurance and Vehicle Telematics : Insurance Market and Regulatory Implications Dimitris Karapipe ris and," Center for Insurance Policy and Research, Tech. Rep., 2015.

[22] Canadian Council of Motor Transport Administrators, "Speed and Intersection Safety management (SISM)," Canadian Council of Motor Transport Administrators Standing Committee on Road Safety Research and Policies, Tech. Rep. May, 2010.

[23] Organisation for Economic Co-operation and Development and O. F. O. R. Economic, "Speed Management," Organisation for Economic Co-operation and Development, Tech. Rep., 2006.

[24] S. Arvidsson, "Reducing asymmetric information with usage-based automobile insurance," *Swedish National Road & Transport Research Institute (VTI)*, 2010. [Online]. Available: http://core.kmi.open.ac.uk/download/pdf/6633086.pdfhttp://econpapers.repec.org/paper/hhsvtiwps/2010{_}5f002.htm

[25] C. Troncoso, G. Danezis, E. Kosta, J. Balasch, and B. Preneel, "PriPAYD: Privacy-friendly pay-as-you-drive insurance," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 5, pp. 742–755, 2011.

[26] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate GPS trajectories," *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, no. c, pp. 352–361, 2009. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1653771.1653820

[27] J. Gomez-Gil, R. Ruiz-Gonzalez, S. Alonso-Garcia, and F. J. Gomez-Gil, "A Kalman filter implementation for precision improvement in Low-Cost GPS positioning of tractors," *Sensors (Switzerland)*, vol. 13, no. 11, pp. 15 307–15 323, 2013.

[28] L. M. Bergasa, D. Almeria, J. Almazan, J. J. Yebes, and R. Arroyo, "DriveSafe: An app for alerting inattentive drivers and scoring driving behaviors," *IEEE Intelligent Vehicles Symposium, Proceedings*, no. Iv, pp. 240–245, 2014.

[29] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Driver Behavior Profiling Using Smartphones : A Low-Cost Platform for Driver Monitoring," *IEEE Intelligent Transportation Systems Magazine*, no. January, pp. 91–102, 2015.

[30] W. Elleuch, A. Wali, and A. M. Alimi, "Collection and exploration of GPS based vehicle traces database," in *2015 4th IEEE International Conference on Advanced Logistics and Transport, IEEE ICALT 2015*, 2015, pp. 275–280.

[31] (2013, sep) Selective Availability. [Online]. Available: http://www.gps.gov/systems/gps/modernization/sa/

[32] United Nations Committee on the Peaceful Uses of Outer Space, "Activities carried out in 2013 in the framework of the workplan of the International Committee on Global Navigation Satellite Systems," United Nations Office for Outer Space Affairs, Tech. Rep. December, 2013.

[33] A. K. Gupta, R. D. Koilpillai, and S. S. S. Evani, "A Kalman Filtering Approach for Integrating MEMS Based INS and GPS for Land Vehicle Application," in *Proceedings of National Conference on Communications*, Kanpur, India, jan 2007, pp. 1–5.

[34] C. Suliman, C. Cruceru, and F. Moldoveanu, "Mobile Robot Position Estimation Using the Kalman Filter," *Scientific Bulletin of the Petru Maior University of Tirgu Mures*, vol. 6, no. 23, pp. 75–78, 2009.

[35] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *International Symposium of Aerospace/Defense Sensing, Simulation. and Controls*, vol. 3, no. 26, 1997, pp. 182–193.

[36] Y. Morales and T. Tsubouchi, "DGPS, RTK-GPS and StarFire DGPS Performance Under Tree Shading Environments," in *2007 IEEE International Conference on Integration Technology*. Ieee, mar 2007, pp. 519–524. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4290370

[37] M. Matosevic, Z. Salcic, and S. Berber, “A Comparison of Accuracy Using a GPS and a Low-Cost DGPS,” *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 5, pp. 1677–1683, 2006.

[38] D. a. Johnson and M. M. Trivedi, “Driving style recognition using a smartphone as a sensor platform,” *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 1609–1615, 2011.

[39] M. Van Ly, S. Martin, and M. M. Trivedi, “Driver classification and driving style recognition using inertial sensors,” *IEEE Intelligent Vehicles Symposium, Proceedings*, no. 4, pp. 1040–1045, 2013.

[40] S. A. Lawoyin, D.-y. Fei, and O. Bai, “A Novel Application of Inertial Measurement Units ( IMUs ) as Vehicular Technologies for Drowsy Driving Detection via Steering Wheel Movement,” no. December, pp. 166–177, 2014.

[41] O. Hallingstad, O. Hegrenaes, and E. Berglund, “Model-Aided Inertial Navigation for Underwater Vehicles,” in *International Conference on Robotics and Automation*, vol. 36, no. 2, Pasadena, California, USA, 2011, pp. 316–337.

[42] E. Foxlin, “Pedestrian Tracking with Shoe-Mounted Inertial Sensors,” *Computer Graphics and Applications, IEEE*, no. December, pp. 38–46, 2005. [Online]. Available: http://www.journals.cambridge.org/abstract{_}S0373463305003164http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=1528431

[43] A. Leardini, G. Lullini, S. Giannini, L. Berti, M. Ortolani, and P. Caravaggi, “Validation of the angular measurements of a new inertial-measurement-unit based rehabilitation system : comparison with state-of-the-art gait analysis,” *Journal of Neuroengineering and Rehabilitation*, vol. 11, no. 136, pp. 1–7, 2014.

[44] G. Ligorio and A. M. Sabatini, “Extended Kalman Filter-Based Methods for Pose Estimation Using Visual, Inertial and Magnetic Sensors: Comparative Analysis and Performance Evaluation,” *Sensors*, vol. 13, no. 2, pp. 1919–1941, 2013.

[45] J. Maye, R. Triebel, L. Spinello, and R. Siegwart, “Bayesian On-line Learning of Driving Behaviors,” in *2011 IEEE International Conference on Robotics and Automation*. Ieee, may 2011, pp. 4341–4346. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5980414

[46] Y. Michalevsky, D. Boneh, and G. Nakibly, “Gyrophone: Recognizing Speech from Gyroscope Signals,” *23rd USENIX Security Symposium (USENIX Security*

*14)*, pp. 1053–1067, 2014. [Online]. Available: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/michalevsky

[47] A. Sathyanarayana, S. O. Sadjadi, and J. H. L. Hansen, "Leveraging sensor information from portable devices towards automatic driving maneuver recognition," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 660–665, 2012.

[48] C. Hatipoglu, Ü. . Özguner, and K. a. Redmill, "Automated lane change controller design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 1, pp. 13–22, 2003.

[49] G. Reymond, a. Kemeny, J. Droulez, and a. Berthoz, "Role of lateral acceleration in curve driving: driver model and experiments on a real vehicle and a driving simulator." *Human factors*, vol. 43, no. 3, pp. 483–495, 2001.

[50] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous Helicopter Aerobatics through Apprenticeship Learning," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010. [Online]. Available: http://ijr.sagepub.com/content/29/13/1608$\delimiter"026E30F$nhttp://ijr.sagepub.com/content/29/13/1608.full.pdf

[51] H. Liu and G. Pang, "Evaluation of RTK-GPS Performance with Low-cost Single-frequency GPS Receivers," in *Proceedings of IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, Tokyo, Japan, oct 1999, pp. 435–440.

[52] I. Mohamad, M. Ali, and M. Ismail, "Availability, reliability and accuracy of GPS signal in Bandar Baru Bangi for the determination of vehicle position and speed," in *2009 International Conference on Space Science and Communication*, no. October, 2009, pp. 224–229. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=5352632

[53] J. Almazan, L. M. Bergasa, J. J. Yebes, R. Barea, and R. Arroyo, "Full auto-calibration of a smartphone on board a vehicle using IMU and GPS embedded sensors," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 1374–1380, 2013.

[54] W. Chen, Z. Li, M. Yu, and Y. Chen, "Effects of Sensor Errors on the Performance of Map Matching," *Journal of Navigation*, vol. 58, no. 2, pp. 273–282, may 2005. [Online]. Available: http://www.journals.cambridge.org/abstract{_}S0373463305003164

[55] D. A. Pomerleau, "ALVINN , an autonomous land vehicle in a neural network," 1989.

[56] R. Ishizaki, M. Morimoto, and K. Fujii, “An evaluation method of driving behavior by in-vehicle data camera,” *International Conference on Emerging Trends in Engineering and Technology, ICETET*, vol. 2, pp. 293–297, 2012.

[57] F. Wei, W. Guo, X. Liu, C. Liang, and T. Feng, “Left-Turning Vehicle Trajectory Modeling and Guide Line Setting at the Intersection,” vol. 2014, 2014.

[58] P. Liu, A. Kurt, and U. Ozguner, “Trajectory Prediction of a Lane Changing Vehicle Based on Driver Behavior Estimation and Classification,” in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. Ieee, oct 2014, pp. 942–947. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6957810

[59] R. K. Satzoda, S. Martin, P. Gunaratne, and M. M. Trivedi, “Towards automated drive analysis: A multimodal synergistic approach,” *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, no. Itsc, pp. 1912–1916, 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6728508

[60] V. Neale, T. Dingus, and S. Klauer, “An overview of the 100-car naturalistic study and findings,” in *19th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, Washington DC, USA, 2005. [Online]. Available: http://www.nhtsa.gov/DOT/NHTSA/NRD/Multimedia/PDFs/CrashAvoidance/2005/100Car{_}ESV05summary.pdf

[61] J. McClafferty and J. Hankey, “100-Car Reanalysis: Summary of Primary and Secondary Driver Characteristics,” Tech. Rep., 2010.

[62] S. Klauer, T. a. Dingus, V. L. Neale, J. Sudweeks, and D. Ramsey, “The Impact of Driver Inattention On Near Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study Data,” *Analysis*, no. April, p. 226, 2006.

[63] T. E. Boyce and E. S. Geller, “An instrumented vehicle assessment of problem behavior and driving style: Do younger males really take more risks?” *Accident Analysis and Prevention*, vol. 34, no. 1, pp. 51–64, 2002.

[64] D. Jin, K. Myung, K. Kim, K. Soo, H. G. Park, and M. Hyung, “Localization System of Autonomous Vehicle via Kalman Filtering Dong Jin Kim\ Myung Kuk Kim\ Kil Soo,” no. I, pp. 934–937, 2011.

[65] J. Engelbrecht, M. J. Booysen, and G.-J. Van Rooyen, "Recognition of driving manoeuvres using smartphone-based inertial and GPS measurement," no. May, 2014.

[66] F. J. Bruwer and M. J. Booysen, "Comparison of GPS and MEMS Support for Smartphone-Based Driver Behavior Monitoring," pp. 434–441, 2015.

[67] J. Dai, J. Teng, X. Bai, Z. Shen, and D. Xuan, "Mobile phone based drunk driving detection," *Pervasive Computing Technologies for Healthcare PervasiveHealth 2010 4th International Conference onNO PERMISSIONS*, pp. 1–8, 2010. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=5482295

[68] C. Barthold, K. Pathapati Subbu, and R. Dantu, "Evaluation of gyroscope-embedded mobile phones," *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, pp. 1632–1638, 2011.

[69] R. Bhoraskar, N. Vankadhara, B. Raman, and P. Kulkarni, "Wolverine: Traffic and road condition estimation using smartphone sensors," in *2012 4th International Conference on Communication Systems and Networks, COMSNETS 2012*, 2012.

[70] P. Mohan, R. Ramjee, and V. N. Padmanabhan, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," *6th ACM conference on Embedded network sensor systems (SenSys '08)*, pp. 323–336, 2008. [Online]. Available: http://research.microsoft.com/pubs/78568/Nericell-Sensys2008.pdf

[71] B. Khaleghi, A. El-ghazal, A. R. Hilal, J. Toonstra, W. B. Miners, and O. A. Basir, "Opportunistic Calibration of a Smartphone Orientation in a Vehicle," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2015.

[72] L. Liu, C. Karatas, H. Li, S. Tan, M. Gruteser, J. Yang, Y. Chen, and R. P. Martin, "Toward Detection of Unsafe Driving with Wearables," *Proceedings of the 2015 workshop on Wearable Systems and Applications - WearSys '15*, pp. 27–32, 2015. [Online]. Available: http://dl.acm.org/citation.cfm?id=2753509.2753518

[73] G. Bishop and G. Welch, "An introduction to the kalman filter," pp. 1–16, 2001. [Online]. Available: http://old.shahed.ac.ir/references/kalman{_}filter{_}notes.pdf

[74] R. Dorobantu and B. Zebhauser, "Field Evaluation of a Low-Cost Strapdown IMU by means GPS," 1999.

[75] R. Bellman, *Dynamic Programming*, 1st ed. Princeton, NJ, USA: Princeton University Press, 1957. [Online]. Available: http://books.google.com/books?id=fyVtp3EMxasC{&}pg=PR5{&}dq=dynamic+programming+richard+e+bellman{&}client=firefox-a{#}v=onepage{&}q=dynamicprogrammingrichardebellman{&}f=falsehttp://www.saylor.org/site/wp-content/uploads/2011/06/Dynamic-Programming.pdf

[76] C. Freeman, D. Kulic, and O. Basir, "An evaluation of classifier-specific filter measure performance for feature selection," *Pattern Recognition*, vol. 48, no. 5, pp. 1812–1826, 2015.

[77] J. Shlens, "A Tutorial on Principal Component Analysis," *ArXiv*, pp. 1–13, 2014.

[78] J. P. Lewis, "Three Derivations of Principal Component Analysis," Tech. Rep.

[79] L. I. Smith, "A tutorial on Principal Components Analysis Introduction," *Statistics*, vol. 51, p. 52, 2002. [Online]. Available: http://www.mendeley.com/research/computational-genome-analysis-an-introduction-statistics-for-biology-and-health/

[80] O. Sutton, "Introduction to k Nearest Neighbour Classification and Condensed Nearest Neighbour Data Reduction," Tech. Rep., 2012.

[81] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, vol. 20, no. 3, sep 1995, pp. 273–297. [Online]. Available: http://link.springer.com/10.1007/BF00994018

[82] D. Mining, K. Discovery, and C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 43, no. 2, pp. 121–167, 1998.

[83] K. P. Bennett and E. J. Bredensteiner, "Duality and geometry in SVM classifiers," in *In Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, 2000, pp. 57–64. [Online]. Available: http://pdf.aminer.org/000/334/459/duality{_}and{_}geometry{_}in{_}svm{_}classifiers.pdf

[84] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[85] B. Hssina, A. Merbouha, H. Ezzikouri, and M. Erritali, "A comparative study of decision tree ID3 and C4.5," *International Journal of Advanced Computer Science and Applications*, no. 2, pp. 13–19, 2014.

[86] A. Jain, H. S. Koppula, B. Raghavan, and A. Saxena, "Know Before You Do : Anticipating Maneuvers via Learning Temporal Driving Models," *arXiv2015*, 2015.

[87] J. McCall, D. Wipf, M. Trivedi, and B. Rao, "Lane Change Intent Analysis Using Robust Operators and Sparse Bayesian Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 431–440, sep 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4298904

[88] A. Doshi and M. M. Trivedi, "Examining the impact of driving style on the predictability and responsiveness of the driver: Real-world and simulator analysis," *IEEE Intelligent Vehicles Symposium, Proceedings*, pp. 232–237, 2010.

[89] A. Houenou and P. Bonnifait, "Vehicle trajectory prediction based on motion model and maneuver recognition," *Intelligent Robots and . . .* , no. 61161130528, pp. 4363–4369, 2013. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6696982

[90] A. Barth and U. Franke, "Tracking oncoming and turning vehicles at intersections," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 861–868, 2010.

[91] T. Sato and M. Akamatsu, "Modeling and prediction of driver preparations for making a right turn based on vehicle velocity and traffic conditions while approaching an intersection," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 11, no. 4, pp. 242–258, jul 2008. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1369847807000733

[92] K. Fitzpatrick and W. Schneider, "Turn Speeds and Crashes within Right-Turn Lanes," vol. 7, no. 2, 2005. [Online]. Available: http://d2dtl5nnlpfr0r.cloudfront.net/tti.tamu.edu/documents/0-4365-4.pdf

[93] X. Yan, E. Radwan, and D. Guo, "Effects of major-road vehicle speed and driver age and gender on left-turn gap acceptance," *Accident Analysis and Prevention*, vol. 39, no. 4, pp. 843–852, 2007.

[94] J. E. Naranjo, C. González, R. García, and T. De Pedro, "Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 438–450, 2008.

[95] R. Schubert, K. Schulze, and G. Wanielik, “Situation assessment for automatic lane-change maneuvers,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 607–616, 2010.

[96] V. Di Lecce and M. Calabrese, “NN-based measurements for driving pattern classification,” *2009 IEEE Intrumentation and Measurement Technology Conference*, no. May, pp. 259–264, may 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5168455

[97] P. Phondeenana, N. Noomwong, A. Prof, and S. Chantranuwathana, “Driving Maneuver Detection System based on GPS Data Driving Maneuver Detection System based on GPS Data,” no. SEPTEMBER, 2013.

[98] M. R. Savino, “Standardized Names and Definitions for Driving Performance Measures,” Ph.D. dissertation, 2009.

[99] S. Al-Sultan, A. H. Al-Bayatti, and H. Zedan, “Context-aware driver behavior detection system in intelligent transportation systems,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4264–4275, 2013.

[100] S. V. Hosseinioun, H. Al-Osman, and A. E. Saddik, “Employing Sensors and Services Fusion to Detect and Assess Driving Events,” *2015 IEEE International Symposium on Multimedia (ISM)*, pp. 395–398, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7442366

[101] O. O. Alharaki, “The Integration of GPS Navigator Device with Vehicles Tracking System for Rental Cars Firms,” vol. 8, no. 6, pp. 1–5, 2010.

[102] I.-k. Hong, K.-h. Lee, and W.-s. Lee, “Development of a Driving Simulator for Virtual Experience and Training of Drunk Driving,” *Advances in Transportation Studies*, pp. 1–13, 2011. [Online]. Available: http://host.uniroma3.it/r...specialissue2011.htm

[103] K. Murata, E. Fujita, S. Kojima, S. Maeda, Y. Ogura, T. Kamei, T. Tsuji, S. Kaneko, M. Yoshizumi, and N. Suzuki, “Noninvasive biological sensor system for detection of drunk driving,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 1, pp. 19–25, 2011.

[104] E. Malar, M. Gauthaam, and D. Chakravarthy, “A Novel Approach for the Detection of Drunken Driving using the Power Spectral Density Analysis of EEG,” *International Journal of Computer Applications*, vol. 21, no. 7, pp. 10–14, 2011. [Online]. Available: http://www.ijcaonline.org/volume21/number7/pxc3873436.pdf

[105] B. R. Chang, H. F. Tsai, and C. P. Young, "Intelligent data fusion system for predicting vehicle collision warning using vision/GPS sensing," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2439–2450, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2009.07.036

[106] N. H. Thien and T. Muntsinger, "Horizontal Gaze Nystagmus Detection in Automotive Vehicles," Carnegie Mellon University, Tech. Rep.

[107] D. E. Marple-Horvat, H. L. Cooper, S. L. Gilbey, J. C. Watson, N. Mehta, D. Kaur-Mann, M. Wilson, and D. Keil, "Alcohol badly affects eye movements linked to steering, providing for automatic in-car detection of drink driving." *Neuropsychopharmacology : official publication of the American College of Neuropsychopharmacology*, vol. 33, no. 4, pp. 849–858, 2008.

[108] A. Aljaafreh, N. Alshabatat, and M. S. Najim Al-Din, "Driving style recognition using fuzzy logic," *2012 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2012*, pp. 460–463, 2012.

[109] V. Vaitkus, P. Lengvenis, and G. Zylius, "Driving style classification using long-term accelerometer information," in *Methods and Models in Automation and Robotics (MMAR), 2014 19th International Conference On*, 2014, pp. 641–644.

[110] T. Pholprasit, W. Choochaiwattana, and C. Saiprasert, "A comparison of driving behaviour prediction algorithm using multi-sensory data on a smartphone," *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2015 - Proceedings*, 2015.

[111] C. D. Agostino, A. Saidi, G. Scouarnec, L. Chen, and S. Member, "Learning-Based Driving Events Recognition and Its Application to Digital Roads," pp. 1–12, 2015.

[112] S. H. Fairclough and R. Graham, "Impairment of driving performance caused by sleep deprivation or alcohol: a comparative study." *Human factors*, vol. 41, no. 1, pp. 118–128, 1999. [Online]. Available: http://hfs.sagepub.com/cgi/doi/10.1518/001872099779577336

[113] X. Fan, B. C. Yin, and Y. F. Sun, "Yawning detection for monitoring driver fatigue," *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, ICMLC 2007*, vol. 2, no. August, pp. 664–668, 2007.

[114] J. He, S. Roberson, B. Fields, J. Peng, S. Cielocha, and J. Coltea, "Fatigue Detection using Smartphones," *Journal of Ergonomics*, vol. 03, no. 03, pp. 1–7, 2013. [Online]. Available: http://www.omicsgroup.org/journals/fatigue-detection-using-smartphones-2165-7556.1000120.php?aid=21520

[115] Z. Zhu and Q. Ji, "Real time and non-intrusive driver fatigue monitoring," *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749)*, vol. 53, no. 4, pp. 657–662, 2004. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1398979

[116] C. T. Lin, R. C. Wu, S. F. Liang, W. H. Chao, Y. J. Chen, and T. P. Jung, "EEG-based drowsiness estimation for safety driving using independent component analysis," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 12, pp. 2726–2738, 2005.

[117] G. Li and W.-Y. Chung, "Detection of driver drowsiness using wavelet analysis of heart rate variability and a support vector machine classifier." *Sensors (Basel, Switzerland)*, vol. 13, no. 12, pp. 16 494–511, 2013. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3892817{&}tool=pmcentrez{&}rendertype=abstract

[118] R. R. Knipling and W. W. Wierwille, "Vehicle-Based Drowsy Driver Detection : Current Status and Future Prospects," in *PROCEEDINGS OF THE 1994 ANNUAL MEETING OF IVHS AMERICA*, 1994, pp. 245–256. [Online]. Available: http://www.itsdocs.fhwa.dot.gov/JPODOCS/REPTS{_}TE/7068.pdf

[119] Y. Kim, Y. Kim, and M. Hahn, "Detecting driver fatigue based on the driver's response pattern and the front view environment of an automobile," *Proceedings of the 2nd International Symposium on Universal Communication, ISUC 2008*, pp. 237–240, 2008.

[120] J. Ma, Z. Wang, X. Lin, P. Wang, and D. Chu, "Vehicle Steering Angle Estimation Using Off-the-shelf Sensor Technologies," no. 2011, pp. 1736–1742, 2013.

[121] H. M. Mandalia and D. D. Salvucci, "Using Support Vector Machines for Lane-Change Detection," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, no. 22, pp. 1965–1969, sep 2005. [Online]. Available: http://pro.sagepub.com/lookup/doi/10.1177/154193120504902217

[122] D. D. Salvucci and A. Liu, “The time course of a lane change: Driver control and eye-movement behavior,” *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 5, no. 2, pp. 123–132, 2002.

[123] M. C. Chuang, R. Bala, E. A. Bernal, P. Paul, and A. Burry, “Estimating gaze direction of vehicle drivers using a smartphone camera,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 165–170, 2014.

[124] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin, “Sensing vehicle dynamics for determining driver phone use,” *Proceeding of the 11th annual international conference on Mobile systems, applications, and services - MobiSys ’13*, p. 41, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2462456.2464447

[125] N. Kalra and D. Bansal, “Analyzing Driver Behavior using Smartphone Sensors : A Survey,” *International Journal of Electronic and Electrical Engineering.*, vol. 7, no. 7, pp. 697–702, 2014.

[126] J. Fukuda, E. Akutsu, and K. Aoki, “An estimation of driver’s drowsiness level using interval of steering adjustment for lane keeping,” *JSAE Review*, vol. 16, no. 2, pp. 197–199, 1995.

[127] D. Sandberg and M. Wahde, “Particle swarm optimization of feedforward neural networks for the detection of drowsy driving,” in *International Joint Conference on Neural Networks (IJCNN*, 2008, pp. 788–793.

[128] C.-w. You, N. D. Lane, F. Chen, R. Wang, Z. Chen, T. J. Bao, M. Montes-de oca, Y. Cheng, M. Lin, L. Torresani, and A. T. Campbell, “CarSafe App: Alerting Drowsy and Distracted Drivers using Dual Cameras on Smartphones Categories and Subject Descriptors,” *Mobisys’13*, pp. 1–14, 2012. [Online]. Available: http://mclab.citi.sinica.edu.tw/cwyou/papers/carsafe{_}mobisys{_}2013.pdf

[129] P. Konstantopoulos, P. Chapman, and D. Crundall, “Driver’s visual attention as a function of driving experience and visibility. Using a driving simulator to explore drivers’ eye movements in day, night and rain driving,” *Accident Analysis and Prevention*, vol. 42, no. 3, pp. 827–834, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.aap.2009.09.022

[130] A. Riener and A. Ferscha, “Effect of proactive braking on traffic flow and road throughput,” *Proceedings - IEEE International Symposium on Distributed Simulation and Real-Time Applications, DS-RT*, pp. 157–164, 2009.

[131] R. Araújo, Â. Igreja, R. De Castro, and R. E. Araújo, “Driving coach: A smartphone application to evaluate driving efficient patterns,” *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 1, no. 1, pp. 1005–1010, 2012.

[132] J. Andrey, B. Mills, and J. Vandermolen, “Weather information and road safety,” *Institute for Catastrophic Loss Reduction, Toronto, Ontario, Canada*, no. 15, 2001. [Online]. Available: http://www.iclr.org/images/Weather{_}information{_}and{_}road{_}safety.pdf

[133] Society of Automotive Engineers, “Electric Speedometer Specification - On Road,” aug 2011. [Online]. Available: http://standards.sae.org/j1226{_}198302/

[134] Government Printing Office, “ELECTRONIC CODE OF FEDERAL REGULATIONS: Speedometer,” aug 2005. [Online]. Available: http://tinyurl.com/m6kjv6z

[135] S. L. Waslander, “ME 597 Assignment 1,” 2013.

[136] Ontario Provincial Standard Specification, “Construction Specification for Repairing Rigid Pavement with Precast Concrete Slabs,” Ontario Provincial Standard Specification, Tech. Rep., 2008.

[137] Ontario Ministry of Transportation. (2014, may) A Guide To Oversize/overweight Vehicles And Loads In Ontario. [Online]. Available: http://www.mto.gov.on.ca/english/trucks/oversize/guide.shtml

[138] ——. (2014) Construction Reports: Northwestern. [Online]. Available: http://www.cdn.mto.gov.on.ca/english/traveller/trip/construction{_}reports-northwestern.shtml

[139] ——. (2010, jun) Driving The Speed Limit. [Online]. Available: http://www.mto.gov.on.ca/english/safety/topics/speed.shtml

[140] R. C. Nicklin, “Kinematics of Tailgating,” *AAPT The Physics Teacher*, vol. 35, no. 2, pp. 78–79, 1997. [Online]. Available: http://scitation.aip.org/content/aapt/journal/tpt/35/2/10.1119/1.2344598

[141] T. G. Dietterich, “Machine Learning for Sequential Data: A Review,” Tech. Rep., 2002.

[142] M. Tipping and C. Bishop, “Probabilistic principal component analysis,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999. [Online]. Available: http://onlinelibrary.wiley.com/doi/10.1111/1467-9868.00196/abstract

[143] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1961199

# APPENDICES

# Appendix A

# Rotation Matrix Estimation

In this chapter, the rotation matrix estimation using the PCA method from Section 4.1 are presented for each trip. The left plot shows the angular velocity about the Z-axis using the PCA method (red) and gravity vector (blue). The centre plot shows the angular velocity error. The right plot illustrates the angular velocity MSE. The date of the trips are labelled in the captions.

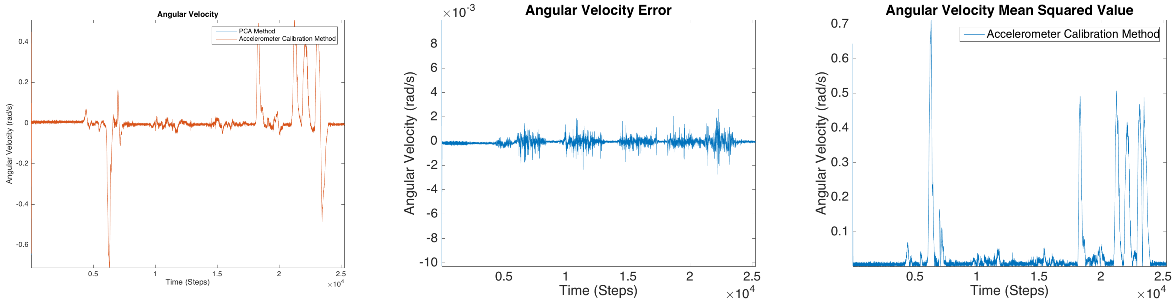


Figure A.1: Aug-7-2015-11-19-20-AM

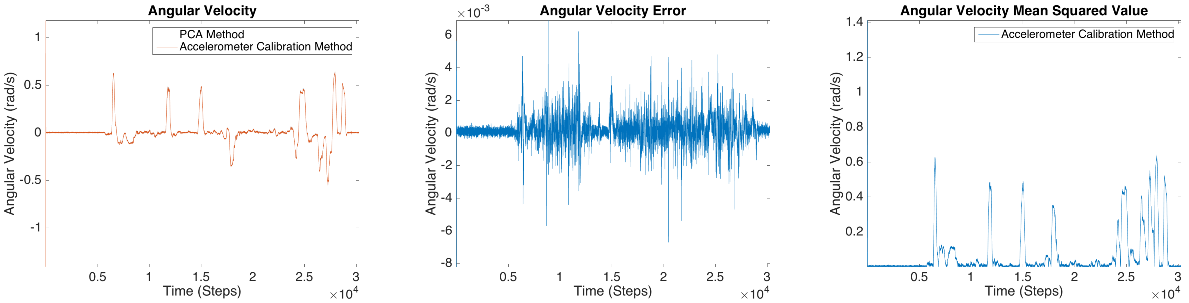


Figure A.2: Sep-15-2015-9-11-04-AM

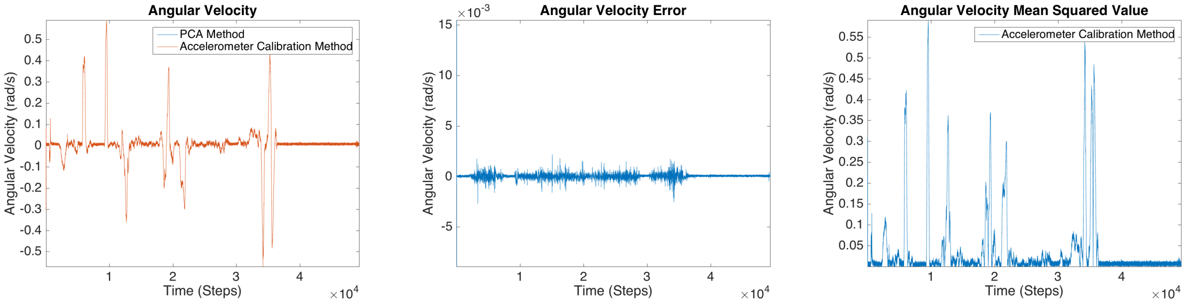


Figure A.3: Sep-16-2015-9-17-33-AM

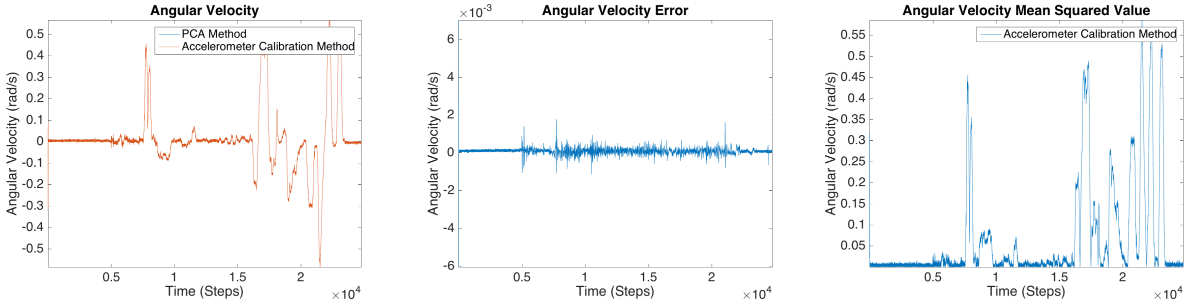


Figure A.4: Sep-16-2015-9-42-41-AM

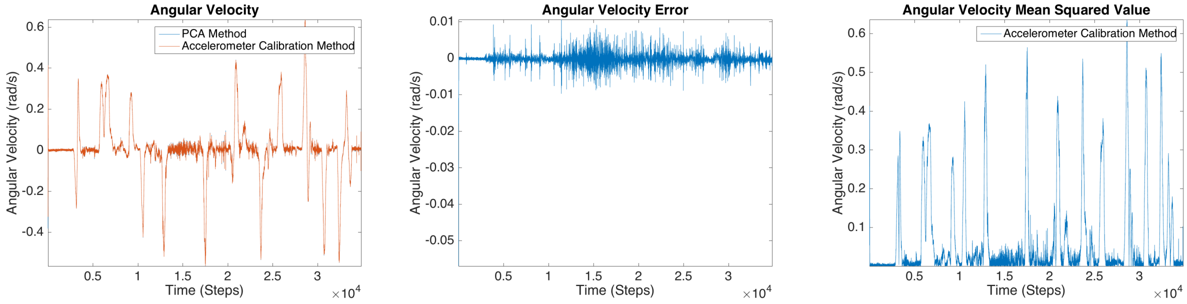


Figure A.5: Oct-13-2015-3-08-48-PM

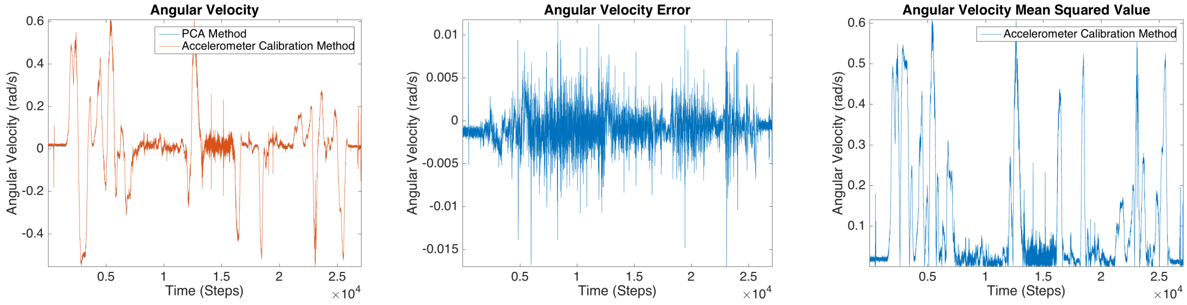


Figure A.6: Oct-14-2015-11-02-22-PM

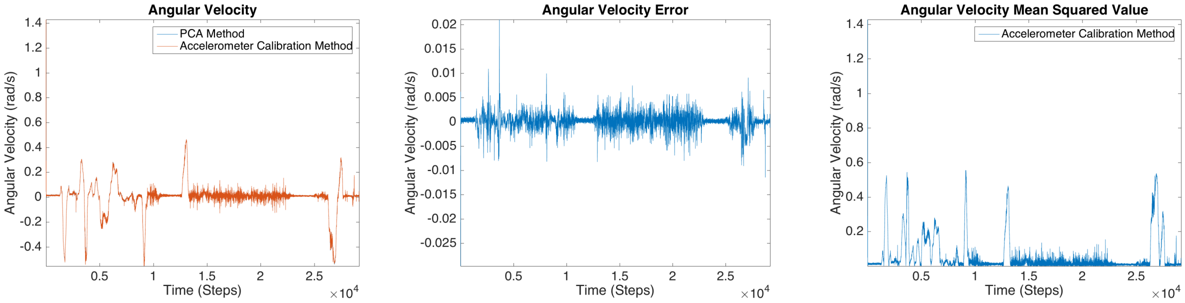


Figure A.7: Oct-15-2015-10-00-39-AM

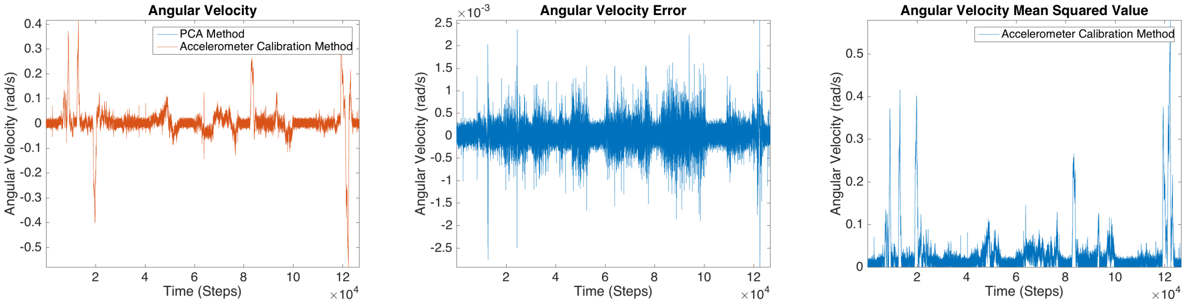


Figure A.8: Oct-15-2015-9-54-02-PM

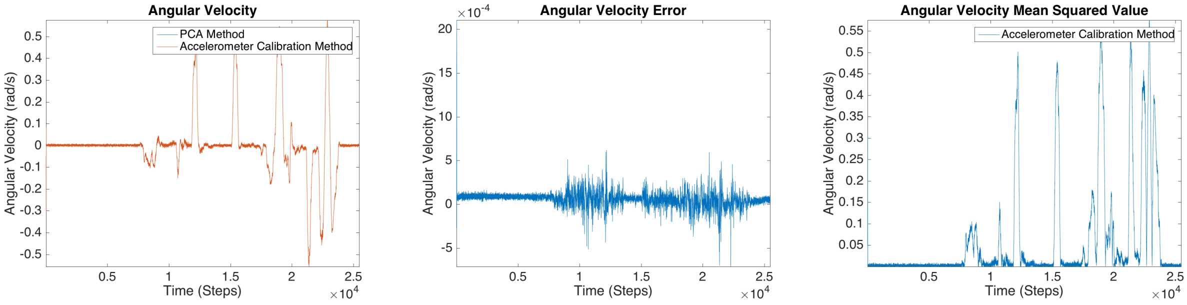


Figure A.9: Nov-12-2015-11-30-00-AM

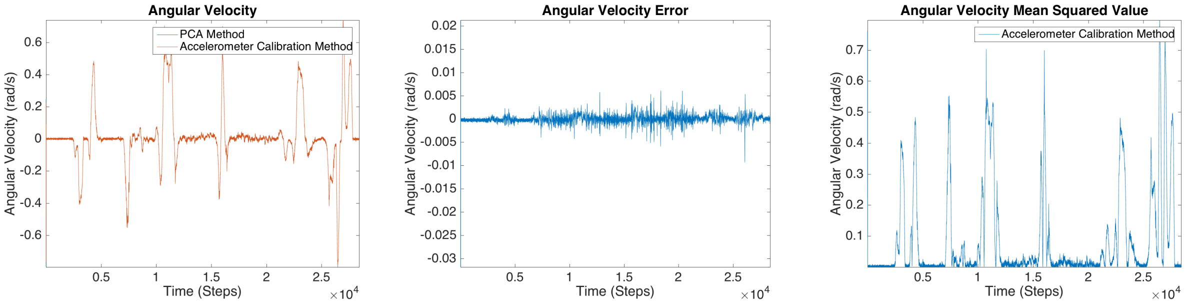


Figure A.10: Nov-12-2015-12-39-49-PM

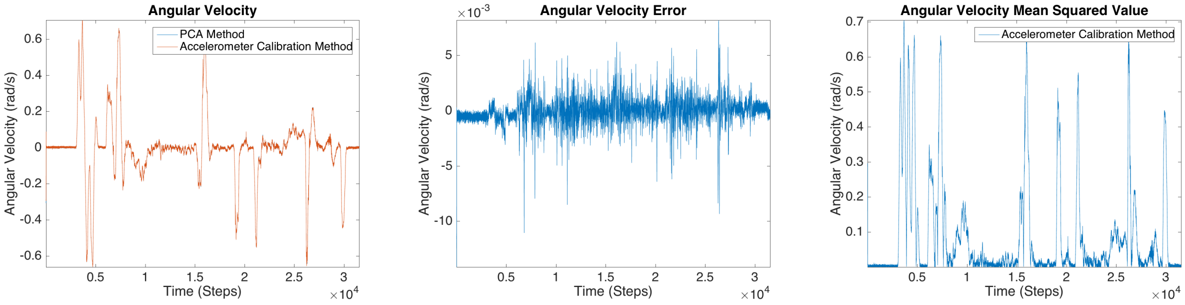


Figure A.11: Nov-12-2015-5-15-59-PM

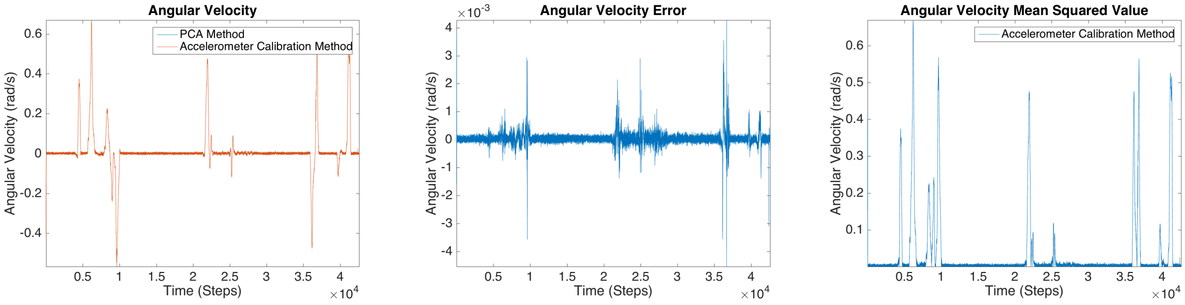


Figure A.12: Nov-14-2015-12-30-02-PM

# Appendix B

# Gyroscope Bias Issue

In this chapter, a subset of gyroscope signals are presented. Figure B.1 illustrates the gyro signal properly calibrated by the Android API. Figure B.2 illustrates the gyro signal miscalibrated by the Android API.

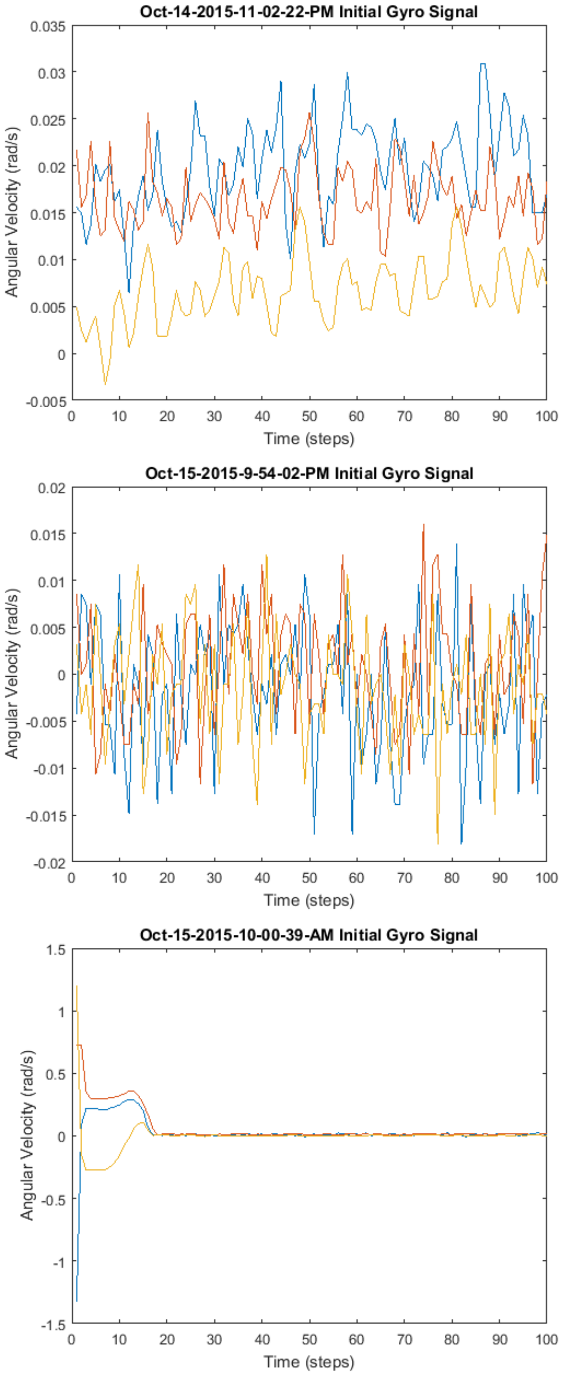


Figure B.1: Calibrated gyro measurements from Android API

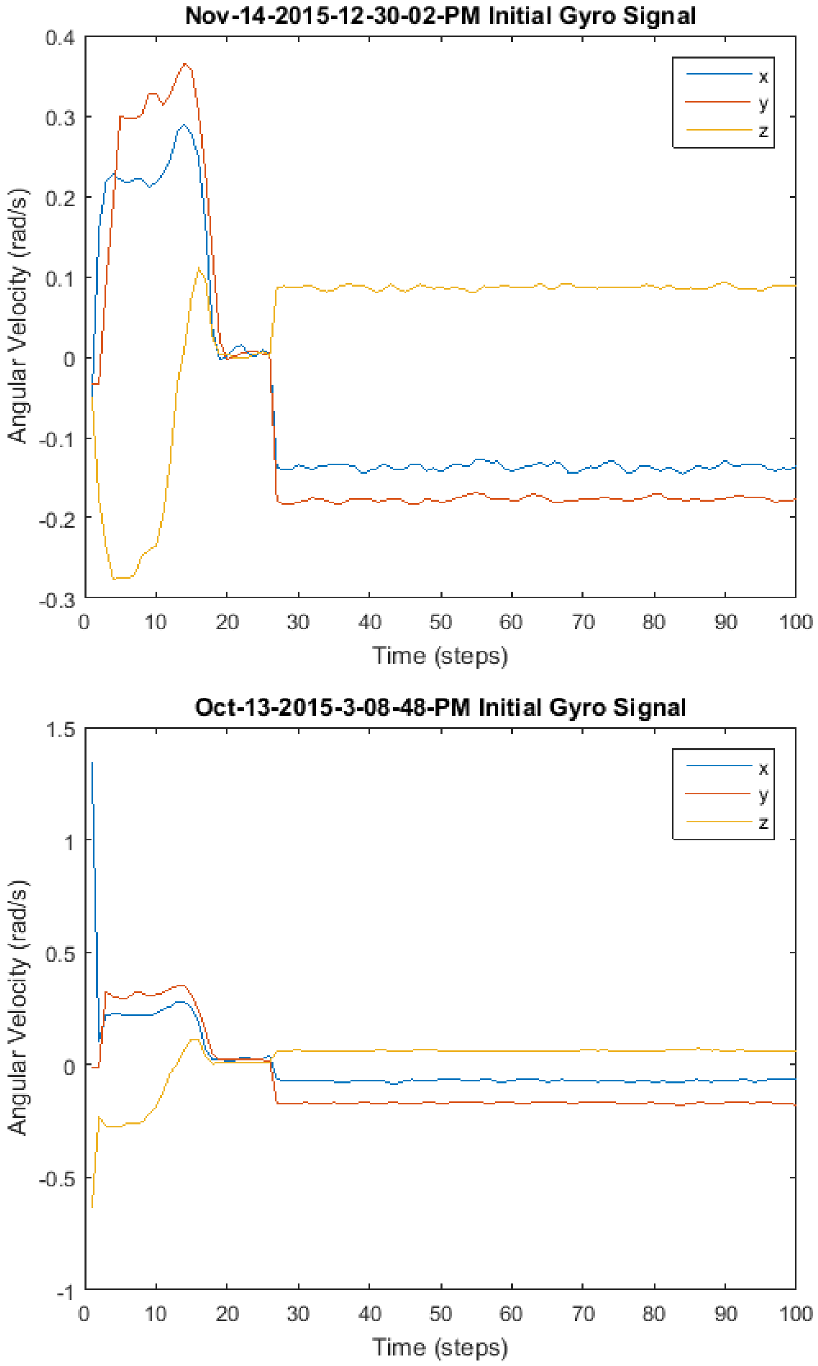


Figure B.2: Biased gyro measurements from Android API