Linköping University | Department of Electrical Engineering
Master’s thesis, 30 ECTS | Automatic Control
2021 | LiTH-ISY-EX--21/5437--SE

# Jointly Ego Motion and Road Geometry Estimation for Advanced Driver Assistance Systems

*Gemensam uppskattning av ego-rörelse och väggeometri för avancerade förarassistanssystem*

**Jawaria Asghar**

Supervisor : Shamisa Shoja
Examiner : Fredrik Gustafsson

External supervisor : Hossein Nemati, Mikhail Ivanov

LINKÖPING
UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

## Upphovsrätt

## Copyright

## Abstract

For several years, there has been a remarkable increase in efforts to develop an autonomous car. Autonomous car systems combine various techniques of recognizing the environment with the help of the sensors and could drastically bring down the number of accidents on road by removing human conduct errors related to driver inattention and poor driving choices.

In this research thesis, an algorithm for jointly ego-vehicle motion and road geometry estimation for Advanced Driver Assistance Systems (ADAS) is developed. The measurements are obtained from the inertial sensors, wheel speed sensors, steering wheel angle sensor, and camera. An Unscented Kalman Filter (UKF) is used for estimating the states of the non-linear system because UKF estimates the state in a simplified way without using complex computations. The proposed algorithm has been tested on a winding and straight road. The robustness and functioning of our algorithm have been demonstrated by conducting experiments involving the addition of noise to the measurements, reducing the process noise covariance matrix, and increasing the measurement noise covariance matrix and through these tests, we gained more trust in the working of our tracker. For evaluation, each estimated parameter has been compared with the reference signal which shows that the estimated signal matches the reference signal very well in both scenarios. We also compared our joint algorithm with individual ego-vehicle and road geometry algorithms. The results clearly show that better estimates are obtained from our algorithm when estimated jointly instead of estimating separately.

*key words*: autonomous car, ego-vehicle, road geometry, estimation, Advanced Driver Assistance Systems, sensor fusion, Unscented Kalman Filter.

# Acknowledgments

First of all, I want to thank Allah who gave me the courage, strength, and faith to carry out this thesis. Further, I would like to express my gratitude to the industrial supervisor Hossein Nemati at Huawei Gothenburg Research Center for giving me the opportunity to learn, expand my skills and capabilities by working on a challenging and dynamic project. I would like to extend my appreciation and gratitude to Mikhail Ivanov, the industry supervisor whose guidance, professional expertise, thoughtful comments, and recommendations have been invaluable throughout this thesis work.

Moreover, I would like to thank my thesis examiner Fredrik Gustafsson Linköping University for mentoring, directing, and taking the time to evaluate this thesis. I further extend my gratitude to my supervisor Shamisa Shoja for her valuable comments and timely feedback which helped me in producing good quality content.

Finally, I would like to express sincere thanks to my family and friends for their unconditional strength, support, and encouragement throughout my thesis work.

*Linköping, August 2021*
*Jawaria Asghar*

# Abbreviations

The list of abbreviations/acronyms used in the thesis are listed below:

**ADAS** Advance Driving Assistance Systems

**JERGE** Jointly Ego Motion and Road Geometry Estimation

**IMU** Inertial Measurement Unit

**CoG** Centre of Gravity

**KF** Kalman Filter

**EKF** Extended Kalman Filter

**UKF** Unscented Kalman Filter

**UT** Unscented Transformation

**RMSE** Root Mean Square Error

# Contents

# List of Figures

# List of Tables

# 1 Introduction

In this chapter, the research objectives, the problem of the study, investigative approaches, and the structuring of the thesis are presented. This chapter offers a brief introduction to the general context of autonomous vehicles to bring the topic around the motivations of the thesis.

## 1.1 Huawei Technologies Sweden AB

This thesis has being carried out at Huawei Research Center Gothenburg with an automotive research team in developing paths for Advanced Driver Assistance Systems.

## 1.2 Background

Advanced Driver Assistance Systems (ADAS) is a rapidly increasing automotive sector and a great topic of interest for researchers. ADAS seeks to increase global safety and increase responsiveness while driving. The integration of ADAS into vehicles is a solution, for now, more affordable than the advent of autonomous vehicles. Several studies show that they have a significant impact on reduction of the number of accidents [22].

For reducing the number of accidents on the road, the ADAS systems are designed in a way to provide more innovative features to facilitate user driving. The level of security granted to these systems is extremely high so that they can be used with confidence. The most important task of autonomous vehicle systems with the help of sensors are collecting information about the environment, estimating their position in the environment, forecasting the movement of other objects, and sending the performed calculations to the system for controlling the vehicle [15]. The performance of these sensors ensures safety. The design phases of ADAS are impacted by the continuous improvement of these technologies. In this thesis, we studied an approach to develop a sensor fusion algorithm for combined estimation of ego-vehicle motion and the geometry of road.

## 1.3 Motivation

In ADAS, the sensor technology [3] is used for measuring the surroundings of a vehicle for estimating the states of a system. The sensors used for this purpose are camera, radar, lidar, and other inertial sensors. With the help of sensor measurements, the ego-vehicle motion can be detected to recognize the movement and heading of a vehicle. The detection and tracking can further be enhanced with the additional information of road geometry along with the ego-motion of vehicle [18]. There are many previous algorithms that estimates separately the ego-vehicle motion and the geometry of the road. However, they do not provide any promising results. Some studies, for example [18] and [9], also provide a combined estimation of the parameters but the approach used in these studies involves complex computations and methods. In this thesis, we have focused on estimating jointly:

- The ego-vehicle motion.
- The road geometry.

The motivation of this research study is to develop an algorithm for joint estimation of ego-motion of vehicle and road geometry for ADAS because an autonomous car works more effectively when it contains not only the information of the car's ego-motion but also have an additional knowledge of the road geometry which results in better performance and enhanced driving experience. The results obtained from our approach are very promising. The findings of this thesis may also aid the company's research on autonomous vehicles.
Figure 1.1 shows an autonomous car with sensor technology. The sensors which we include in the thesis are inertial sensors, steering wheel angle sensor, wheel speed sensors, and camera.

Figure 1.1: Autonomous Car with sensor technology.

## 1.4 Aim

Many researches related to ADAS still work on the development of a vehicle equipped with systems that do not require human participation in the driving process. The goals are to ensure maximum safety by minimizing accidents and optimization of transport costs by eliminating the driver from the vehicle.

In this thesis, we developed a sensor fusion algorithm for the estimation of ego-vehicle motion and road geometry. An Unscented Kalman Filter (UKF) is used for the purpose of sensor fusion and estimating the states. The sensor provides the measurements from the inertial sensors, wheel speed sensors, steering wheel angle sensor, and camera.

The purpose of this research study is to investigate the state-of-art sensor fusion algorithm to answer the following questions/objectives:

1. How to develop a sensor fusion algorithm for jointly ego-motion and road geometry estimation (JERGE)?
2. How to evaluate and analyze the results of the algorithm?

3. How the algorithm performs by computing Root Mean Square Error (RMSE) between the measured and estimated values?
4. How is the performance of the algorithm when compared with the baseline algorithm?
5. What suggestions can be proposed for further continuation of the research study?

## 1.5 Thesis Outline

This thesis consists of seven chapters including Introduction (chapter 1) defining the introduction to the research study, Background (chapter 2) represents the theory and background related to the research topic, Data Exploration and Experiments (chapter 3) discusses the data and experiments performed in the thesis, Method (chapter 4) discusses the methodology and algorithm implementation, Results and Discussion (chapter 5) presents the investigative results and discussion achieved from the proposed approach, Conclusion (chapter 6) discusses the interpretation drawn from the research study, and finally, Future work (chapter 7) proposed the future work for further continuation of the research study.

## 1.6 Delimitations

This thesis was examined under two scenarios, the first scenario contains 130 seconds information from a winding road, and the second scenario involves a 56 seconds drive on a straight road. The algorithm was working for the selected scenarios and due to time constraints, the analysis has not been tested for the other scenarios and is out of the scope of the thesis.

# 2 Background

The objective of this chapter is to introduce the notions, concepts, definitions, and general principles used in the approach.

## 2.1 Literature Survey

Recent years have been a period of intensive progress in the field of ADAS. This involves the development and improvement of available solutions in this area but adapting the systems remains the biggest challenge to the emerging solutions. The high dynamics of the sector require a flexible approach and a rigid framework. History shows that such attempts have already been made with the advent of the first vehicles operated by humans [21]. There are many previous researches done on the topic of ADAS for many years.

This literature survey collects a few of the algorithms used for road geometry and ego-vehicle motion estimation. In [5], a dynamic model was proposed in the context of autonomous vehicles. The algorithm describes the model for ego-vehicle, the road, as well as the leading vehicle with the sensor information from radar, camera, and IMU for computing road geometry estimation and tracking of the vehicle. An Extended Kalman Filter was utilized in this approach for the state estimation. The authors of the article [9], developed a novel road model to describe the road information with high accuracy using a Bayesian fusion system. The sensors used in this approach were radar, camera, IMU, and wheel speed sensors. In this system, the surrounding information was fused from different objects like lane markings, leading vehicles, and barriers for estimating the road geometry ahead of the vehicle. The writers of [14] described a vision-based approach for the estimation of motion of ego-vehicle with a series of stereo images. For estimation in a dynamic environment, an iterative Sigma Point Kalman Filter was utilized along with RANSAC-based outlier rejection scheme capable of estimating frame-to-frame motion.

In [20], a detailed survey on lane-marking detection was performed and a video-based lane estimation and tracking (VioLET) system was proposed using a steerable filter which gives an effective technique to detect and track the lane in various weather and road conditions. The analysis provides a good overview of the efforts delivered by the automotive community and automobile in recent years. In [17], an approach was proposed by using a single-track model for estimating road geometry and tracking of the vehicle by using a new and improved model for road and host vehicles. An EKF was used for the non-linear model to estimate the state. In [12], a filtering algorithm was developed for estimating the road shape with the host vehicle provided with vehicle internal sensors, a radar, and a camera. These sensors utilize the observations of the lane-marking, leading vehicle direction, and reflections of roadside objects. In this approach, a road model based on clothoid was derived with a Bayesian fusion framework for estimating the geometry of the road. [30] used a monocular camera mounted on the ego vehicle for the detection of moving objects and motion of the ego-vehicle. This approach was able to estimate correctly ego-vehicle motion in various conditions e.g. cars, objects, and the camera itself in a moving state.

In [7], the information of the road surface was performed by using the image processing algorithm in which the estimates being carried out using a planar regression, and the optimization was performed using the squares technique. However, the robustness of such an approach is not guaranteed in the case of a complex scene, which contains many pixels that do not belong on the road surface. Another method proposed by [2],[1] demonstrated the 3D reconstruction and the relative location of the vehicle about its lane. The parameters of a 3D surface integrating the horizontal and vertical profiles were estimated from the track edges.

In this thesis, we focused on evaluating an approach for estimating the combined ego-motion and road geometry by using UKF. The UKF estimates the state in a simplified way without using the complex computation of Jacobian. The details of this method are discussed in chapter 4.

## 2.2 Sensor Fusion

Sensor fusion is a process of combining the data from various sensors for state estimation [10]. The sensor fusion algorithms make it possible to robustify the measurements by cross-checking information delivered by the various sensors while taking into account their uncertainties. The sensors associated with data fusion then allow the perception of vehicle motion and road geometry estimation. It is also possible with the help of sensor fusion to remove one of the sensors and still perform the same tasks [16]. The block diagram representing the sensor fusion framework for the approach can be seen in Figure 2.1.

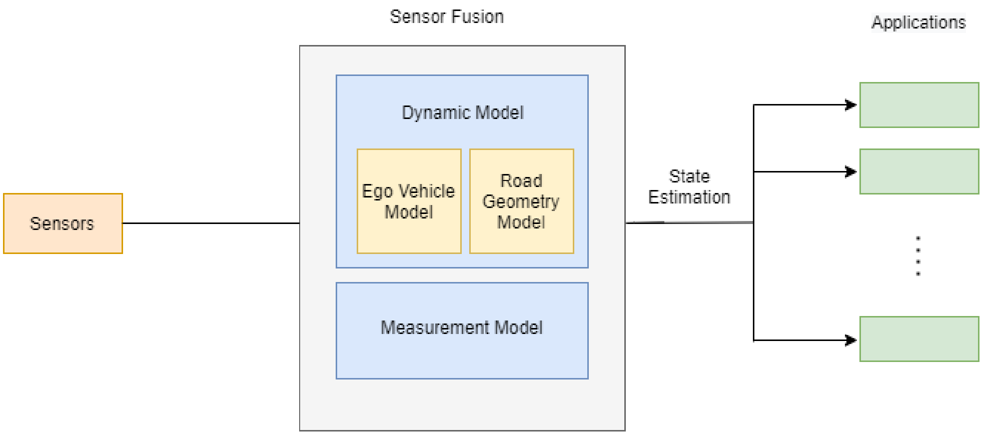


Figure 2.1: Sensor Fusion framework for the system [18].

The description of the block diagram is as follows:

- Dynamic model containing the dynamics of the road geometry and ego-vehicle to structure the process model.
- Measurement model describing the relation of measurements with the states.

- Inertial sensors, wheel speed sensors, steering wheel angle sensor, and camera sensor providing the measurements and input signals for estimating the road geometry and ego-vehicle motion.
- UKF for computing estimates based on the information of measurements.

The measurements are collected by the sensor fusion framework and fuses them for producing the state estimate.

## 2.3 Kalman Filter

Kalman filter is a recursive filter that allows estimating the state of a linear system from the information of input parameters and actual measurements. The Kalman filter is a treatment tool used in a wide range of technological fields such as signal processing, automation, radar, and communication systems [11].

In this thesis, we have a nonlinear and non-Gaussian case because our model is in the form of equation 2.3, 2.4 in which the estimates of the states are computed from the measurement information. For a system that also counts on the nonlinearity then the state transition and measurement function become non-linear. There is no closed-form solution for a nonlinear, and non-Gaussian case, so it is required to make some approximations [18] to resolve the issue of non-linearity. The most commonly used approximations are provided by the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) [31]. In this thesis, we have utilized UKF.

## 2.4 Dynamic Model

The sensor fusion framework utilizes the dynamic model for ego-vehicle and road geometry estimation. The details of parameters with their definitions are given in Table 2.1:

### a. Single-Track Model

The single-track model is utilized for the motion of the ego vehicle. It is also known as the bicycle model [24]. In a single- track model, the wheels at the rear and front axle are exhibited as single components instead of four wheels and contains lateral, longitudinal, and yaw motion of the vehicle. Figure 2.2, describes the parameters and the geometry for the bicycle model.

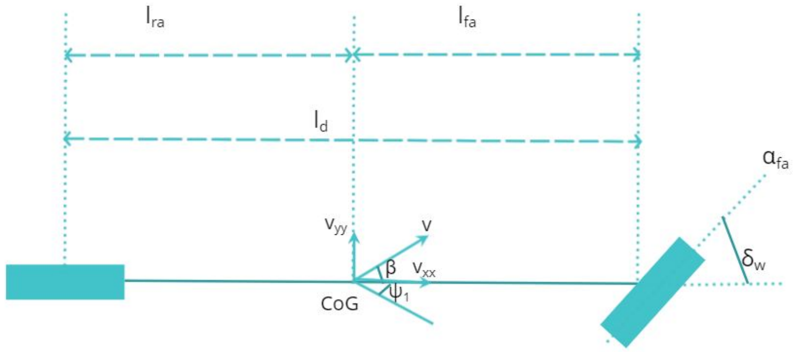


Figure 2.2: Geometry of the single-track model

Table 2.1: Parameters and Definition

| Parameters | Definition |
|---|---|
| $m$ | mass of vehicle (kg) |
| $l_{\mathrm{fa}}, l_{\mathrm{ra}}$ | Distance between the front and rear axle and the center of vehicle gravity respectively (m), respectively |
| $C_{\alpha \mathrm{fa}}, C_{\alpha \mathrm{ra}}$ | Cornering stiffness at front and real axle respectively (N/rad) |
| $\alpha_{\mathrm{fa}}, \alpha_{\mathrm{ra}}$ | slip angle at front and rear axle respectively (rad) |
| $I$ | vehicle's moment of inertia about the z-axis (kg$m^2$) |
| $\beta_1$ | float angle (rad) |
| $c$ | curvature of road ($m^{-1}$) |
| $w_2$ | road width (m) |
| $\delta$ | angle between lane and ego-vehicle velocity vector (rad/s) |
| $l_d$ | distance between front and rear wheel center (m) |
| $l_l, l_r$ | distance between mid-lane to the left and right side of the road respectively (m) |
| $\dot{\psi}_1$ | yaw rate (rad/s) |
| $a_{\mathrm{yy}}$ | lateral acceleration (m/s) |
| $\delta_2$ | angle between longitudinal axis of ego-vehicle and tangent of the road curvature (rad) |
| $v_{\mathrm{xx}}$ | longitudinal velocity (m/s) |
| $v_{\mathrm{yy}}$ | lateral velocity (m/s) |
| $\dot{v_{\mathrm{xx}}}$ | longitudinal acceleration (m/s$^2$) |
| $\delta_{\mathrm{w}}$ | front wheel angle (rad) |

The vehicle's path of movement in the longitudinal direction with an angle $\beta_1$ is known as float angle and can be expressed as:

$$\tan \beta_1 = \frac{v_{\mathrm{yy}}}{v_{\mathrm{xx}}} \tag{2.1}$$

where $v_{\mathrm{yy}}$ and $v_{\mathrm{xx}}$ are the lateral and longitudinal velocity components of the vehicle. The vehicle's velocity vector $\boldsymbol{v}$ together with the yaw rate and float angle are attached at the origin also known as CoG. Other important parameters of the bicycle model are the slip angle $\boldsymbol{\alpha}_{\mathrm{i}}$ and the cornering stiffness $\boldsymbol{C}_{\boldsymbol{\alpha}\mathrm{i}}$ which contains a lot of information on the dynamics of the vehicle [8]. The slip angle represents the angle between the actual wheel's direction and the direction in which the wheel points and the cornering behavior of the tire are described by cornering stiffness. For small values of slip angle, the tire's lateral frictional force can be assumed to proportional to the slip angle [19] as:

$$\boldsymbol{F}_{\mathrm{i}} = \boldsymbol{C}_{\boldsymbol{\alpha}\mathrm{i}} \boldsymbol{\alpha}_{\mathrm{i}} \tag{2.2}$$

### b. Road Model

The dynamic model describing the road geometry consists of the curvature, road width, and the angle between lane and velocity vector of ego-vehicle. The most important part of the road geometry is curvature. The road geometry is linked to the dynamics of the vehicles. If the driver holds the wheel in the center, the vehicle follows a straight line. If the driver applies a steering angle, the vehicle turns in a circle of radius $\boldsymbol{R}$. When going from a straight line to curvature with radius $\boldsymbol{R}$, the driver changes the steering wheel angle from zero to some small value. The road curvature is divided into infinitely small segments represented as $\boldsymbol{dr_d}$ and $\boldsymbol{dr}$ represents the driven path on the road. In figure 2.3, the parameters and geometry for the road are described.

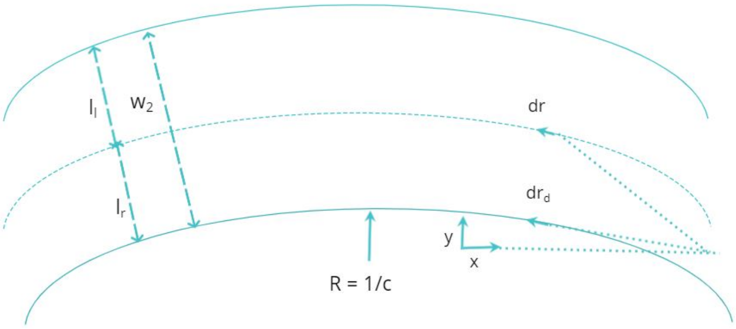


Figure 2.3: Geometry of the road

### State-Space Model

Our state-space model contains the ego-vehicle and road geometry dynamics together with the measurement relations. The state vector $\boldsymbol{x}$, input signal $\boldsymbol{u}$, and process noise $\boldsymbol{v}$ which is Gaussian can be expressed as [18]

$$x_{\mathrm{k+1}} = f(x_{\mathrm{k}}, u_{\mathrm{k}}) + v_{\mathrm{k}} \qquad x_{\mathrm{k}} \in \mathcal{R}^{N_{\mathrm{x}}}, u_{\mathrm{k}} \in \mathcal{R}^{N_{\mathrm{u}}} \tag{2.3}$$

where $\boldsymbol{f}$ is the state transition function which describes the state evolution to time. The measurements $\boldsymbol{y}$, and measurement noise $\boldsymbol{e}$ which is Gaussian can be expressed as

$$y_{\mathrm{k}} = h(x_{\mathrm{k}}, u_{\mathrm{k}}) + e_{\mathrm{k}} \qquad y_{\mathrm{k}} \in \mathcal{R}^{N_{\mathrm{y}}} \tag{2.4}$$

where $\boldsymbol{h}$ is the measurement function which represents the relation of measurements to the state. The state vector consisting of the ego-vehicle dynamics is represented as $\boldsymbol{x_1}$ and we have

$$x_1 = [\dot{\psi}_1 \quad \beta_1 \quad l_1]^T \tag{2.5}$$

The state vector for the road geometry is denoted as $\boldsymbol{x_2}$ and is given by

$$x_2 = [c \quad \delta \quad w_2]^T \tag{2.6}$$

The complete state vector consists of the two parts for ego-vehicle motion and road geometry and is specified by

$$x = [\dot{\psi}_1 \quad \beta_1 \quad l_1 \quad c \quad \delta \quad w_2]^T \tag{2.7}$$

The input signal $u_{\mathrm{k}}$ contains the front wheel angle $\delta_{\mathrm{w}}$, the longitudinal acceleration $\dot{v_{\mathrm{xx}}}$ and velocity $v_{\mathrm{xx}}$ and is given by

$$u_{\mathrm{k}} = [\delta_{\mathrm{w}} \quad v_{\mathrm{xx}} \quad \dot{v_{\mathrm{xx}}}]^T \tag{2.8}$$

Our state-space model is in discrete form, in driving the evolution of states over time continuous-time differential equations are obtained. One of the simplest ways of converting the differential equations into discrete form can be performed by using the method of standard forward Euler [18] which approximates continuous differential equation relative to

$$x_{\mathrm{t+1}} = x_{\mathrm{t}} + Tg(x_{\mathrm{t}}, u_{\mathrm{t}}) \overset{\Delta}{=} f_{\mathrm{t}}(x_{\mathrm{t}}, u_{\mathrm{t}}) \tag{2.9}$$

where the time sample is denoted by T. The measurement model utilized in this thesis is in discrete form already. The evolution of float angle $\beta_1$ and yaw rate $\dot{\psi}_1$ over the time is represented as [18]:

$$\ddot{\psi}_1 = \beta_1 \frac{-C_{\mathrm{fa}} l_{\mathrm{fa}} \cos\delta_w + C_{\mathrm{ra}} l_{\mathrm{ra}}}{I} - \dot{\psi}_1 \frac{C_{\mathrm{fa}} l_{\mathrm{fa}}^2 \cos\delta_w + C_{\mathrm{ra}} l_{\mathrm{ra}}^2}{I v_{\mathrm{xx}}} + \frac{C_{\mathrm{fa}} l_{\mathrm{fa}} \tan\delta_w}{I} \tag{2.10}$$

$$\dot{\beta}_1 = -\beta_1 \frac{C_{\mathrm{fa}} \cos\delta_w + C_{\mathrm{ra}} + \dot{v_{\mathrm{xx}}} m}{m v_{\mathrm{xx}}} - \dot{\psi}_1 \left(1 + \frac{C_{\mathrm{fa}} l_{\mathrm{fa}} \cos\delta_w - C_{\mathrm{ra}} l_{\mathrm{ra}}}{m v_{\mathrm{xx}}^2}\right) + \frac{C_{\mathrm{fa}} \sin\delta_w}{m v_{\mathrm{xx}}} \tag{2.11}$$

The road curvature $c$ in this model is described at ego-vehicle and represents the driven curvature by the vehicle. The change of $c$ to time is described as

$$\begin{aligned}
\dot{c} = {} & \frac{1}{(Im^2 v_{\mathrm{xx}})^4} (C_{\mathrm{ra}}^2 (I + l_{\mathrm{ra}}^2 m)(-\dot{\psi}_1 l_{\mathrm{ra}} + \beta_1 v_{\mathrm{xx}}) + C_{\mathrm{fa}}^2 (I + l_{\mathrm{fa}}^2 m)(\dot{\psi}_1 l_{\mathrm{fa}} + (\beta_1 - \delta_w) v_{\mathrm{xx}}) \\
& + \mathrm{C}_{\mathrm{ra}} Im(-3\dot{\psi}_1 v_{\mathrm{xx}} l_{\mathrm{ra}} + 3\beta_1 \dot{v_{\mathrm{xx}}} v_{\mathrm{xx}} + \dot{\psi}_1 v_{\mathrm{xx}}^2) + I \dot{v_{\mathrm{xx}}} v_{\mathrm{xx}} m^2 (2\beta_1 \dot{v_{\mathrm{xx}}} + v_{\mathrm{xx}}(\dot{\psi}_1 - c v_{\mathrm{xx}})) \\
& + \mathrm{C}_{\mathrm{fa}} (C_{\mathrm{ra}} (I + l_{\mathrm{ra}}(-l_{\mathrm{fa}}) m)(\dot{\psi}_1 l_d - 2\dot{\psi}_1 l_{\mathrm{ra}} + 2\beta_1 v_{\mathrm{xx}} - \delta_w v_{\mathrm{xx}}) \\
& + (3\,\mathrm{l}_{\mathrm{fa}} \dot{v_{\mathrm{xx}}} \dot{\psi}_1 + (3\beta_1 - 2\delta_w) \dot{v_{\mathrm{xx}}} v_{\mathrm{xx}} + (\dot{\delta_w} + \dot{\psi}_1) v_{\mathrm{xx}}^2) mI))
\end{aligned} \tag{2.12}$$

The width of the road does not change as frequently over time as compared to the other variables of the model due to which we yield

$$\dot{w_2} = 0 \tag{2.13}$$

The rate of change of distance between the left lane and ego-vehicle changes is described by again considering the infinite part $dr$ on the road defining the driven curvature of the ego-vehicle as

$$\dot{l_1} = v_{\mathrm{xx}} \sin(\delta + \beta_1) \tag{2.14}$$

The evolution of angle between lane and vehicle velocity vector over time is described according to

$$\dot{\delta} = c v_{\mathrm{xx}} + \beta_1 \frac{C_{\mathrm{fa}} \cos\delta_w + C_{\mathrm{ra}} + \dot{v_{\mathrm{xx}}} m}{m v_{\mathrm{xx}}} + \dot{\psi}_1 \frac{C_{\mathrm{fa}} l_{\mathrm{fa}} \cos\delta_w - C_{\mathrm{ra}} l_{\mathrm{ra}}}{m v_{\mathrm{xx}}^2} - \frac{C_{\mathrm{fa}} \sin\delta_w}{m v_{\mathrm{xx}}} \tag{2.15}$$

This concludes the derivation for the dynamic model of ego-vehicle and road geometry.

### Non-linear state-space Model

The non-linear state-space model comprising of the evolution of states $\psi_1$, $\beta_1$, and $l_1$ over time for ego-vehicle is provided by

$$g_1 = \begin{pmatrix} \beta_1 \frac{-C_{\mathrm{fa}} l_{\mathrm{fa}} \cos\delta_w + C_{\mathrm{ra}} l_{\mathrm{ra}}}{I} - \dot{\psi}_1 \frac{C_{\mathrm{fa}} l_{\mathrm{fa}}^2 \cos\delta_w + C_{\mathrm{ra}} l_{\mathrm{ra}}^2}{I v_{\mathrm{xx}}} + \frac{C_{\mathrm{fa}} l_{\mathrm{fa}} \tan\delta_w}{I} \\ -\beta_1 \frac{C_{\mathrm{fa}} \cos\delta_w + C_{\mathrm{ra}} + \dot{v_{\mathrm{xx}}} m}{m v_{\mathrm{xx}}} - \dot{\psi}_1 \left(1 + \frac{C_{\mathrm{fa}} l_{\mathrm{fa}} \cos\delta_w - C_{\mathrm{ra}} l_{\mathrm{ra}}}{m v_{\mathrm{xx}}^2}\right) + \frac{C_{\mathrm{fa}} \sin\delta_w}{m v_{\mathrm{xx}}} \\ v_{\mathrm{xx}} \sin(\delta + \beta_1) \end{pmatrix} \tag{2.16}$$

The non-linear state space model consisting of the $c$, $\delta$, and $w_2$ states change over time for the road geometry is described as

$$g_2 = \begin{pmatrix} \dot{c} \\ c v_{\text{xx}} + \beta \frac{C_{\text{fa}} \cos \delta_w + C_{\text{ra}} + \dot{v_{\text{xx}}} m}{m v_{\text{xx}}} + \dot{\psi}_1 \frac{C_{\text{fa}} l_{\text{fa}} \cos \delta_w - C_{\text{ra}} l_{\text{ra}}}{m v_{\text{xx}}^2} - \frac{C_{\text{fa}} \sin \delta_w}{m v_{\text{xx}}} \\ 0 \end{pmatrix} \tag{2.17}$$

The overall non-linear state-space model for ego-motion of vehicle and road geometry can be compacted as:

$$g = [g_1 \quad g_2]^T \qquad\qquad g : \mathcal{R}^{N_{\text{x}}} \times \mathcal{R}^{N_{\text{u}}} \to \mathcal{R}^{N_{\text{x}}} \tag{2.18}$$

## 2.5 Measurement Model

The measurement model describes the relation of measurements to the states. The measurements from inertial sensors, steering wheel angle sensor, wheel speed sensors, and camera are used for computing good estimates of road geometry and the ego vehicle motion on the road.
The measurements related to the ego-vehicle are described as

$$y_1 = [\dot{\psi}_1^m \quad a_{\text{yy}}^m]^T \tag{2.19}$$

The measurements linked to the road geometry are defined as

$$y_2 = [c^m \quad \delta_2^m \quad w_2^m \quad l_1^m]^T \tag{2.20}$$

The overall measurement vector for ego-motion of vehicle and road geometry can be represented as

$$y = [\dot{\psi}_1^m \quad a_{\text{yy}}^m \quad c^m \quad \delta_2^m \quad w_2^m \quad l_1^m]^T \tag{2.21}$$

The measurement equation relating to the measurements for ego-vehicle is given by

$$h_1 = \begin{pmatrix} \dot{\psi}_1 \\ -\beta_1 \frac{C_{\text{fa}} \cos \delta_w + C_{\text{ra}} + \dot{v_{\text{xx}}} m}{m} + \dot{\psi}_1 \frac{-C_{\text{fa}} l_{\text{fa}} \cos \delta_w + C_{\text{ra}} l_{\text{ra}}}{m v_{\text{xx}}} + \frac{C_{\text{fa}}}{m} \sin \delta_w \end{pmatrix} \tag{2.22}$$

As we can observe from $h_1$ that the measurement for $a_{\text{yy}}$ has the knowledge about the states of the ego-vehicle. The measurement equation associated to the measurements for the road geometry is represented by

$$h_2 = [c \quad (\delta + \beta_1) \quad w_2 \quad l_1]^T \tag{2.23}$$

It can be seen from the equation of $h_2$ that the measurement $\delta_2$ has been split into two terms which are $\delta$ and $\beta_1$ because our concern is to estimate both of them. The overall measurement equation for ego-vehicle and road geometry is described as

$$h = [h_1 \quad h_2]^T \qquad\qquad h : \mathcal{R}^{N_{\text{x}}} \times \mathcal{R}^{N_{\text{u}}} \to \mathcal{R}^{N_{\text{y}}} \tag{2.24}$$

This concludes the background and model description of the approach.

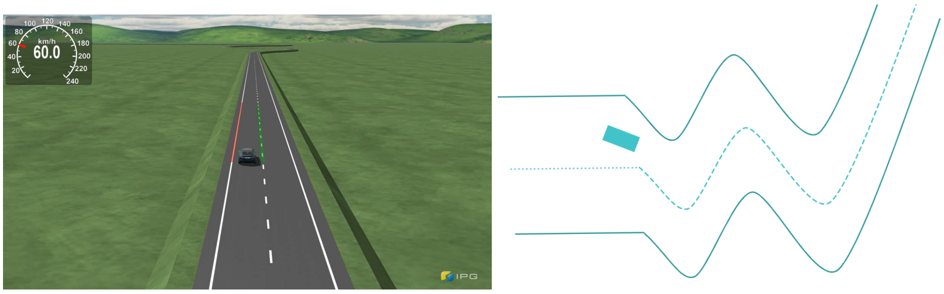
# 3 Data Exploration and Experiments

All the experiments and simulations are discussed in this chapter. The algorithm is performed in Python, and the design of scenario simulation is performed in CarMaker 9.0.

To predict the comfort and safety of a vehicle, the study of its movement in interaction with the environment is required. The vehicle is also subjected to some external disturbances like the irregularity of the route, i.e., "curvature", and requires inputs from front wheel angle, acceleration, and longitudinal velocity [26]. The simulation software, CarMaker 9.0, used in the thesis for the analysis of data, is a set of simulation tools [25] for virtual driving tests of a road vehicle.

## 3.1 Experiment

The experiments were carried out in the CarMaker simulator. Two scenarios were generated for the experiments. In the first scenario, the ego-vehicle completes a 130 ***sec*** route on a winding road with an initial speed of 90 ***km/h*** to a constant speed of 60 ***km/h*** after completing a route distance of 10 ***sec***. The depiction of the scenarios can be observed in Figure 3.1.



(a)

(b)

Figure 3.1: (a) CarMaker Simulation Tool (b) Scenario illustration

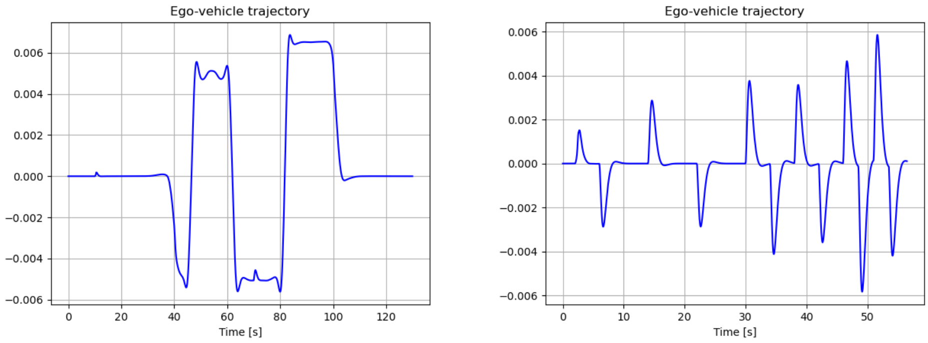
In the second scenario, the car completes a route of around 56 ***sec*** on a straight road with an initial speed of 90 ***km/h*** to a constant speed of 60 ***km/h*** from 7 ***sec*** on-wards. The illustration of the scenario is represented in Figure 3.2.

(a)

(b)

Figure 3.2: (a) CarMaker Simulation Tool (b) Scenario illustration

The trajectory of ego-vehicle for first and second scenario can be seen in Figure 3.3. In Figure 3.3 (a), it can be observed that the ego-vehicle nearly follows the same direction as the road path. From Figure 3.3 (b), it can be noted that ego-vehicle travels with some frequent lateral movements.



(a)

(b)

Figure 3.3: (a) $1^{st}$ Scenario (b) $2^{nd}$ Scenario

The electric vehicle used for the simulation of the scenario was the Tesla Model S. The values of vehicle parameters used in this thesis are described in Table 3.1.

Table 3.1: Parameter Values

| **Parameters** | **Value** |
|---|---|
| $m$ | 1800 kg |
| $l_{\mathrm{fa}}, l_{\mathrm{ra}}$ | 2.6 m, 0.578 m |
| $l_d$ | 3.178 m |
| $I$ | 1700 kg$m^2$ |

## 3.2 Sensor Data

A sensor is a device that measures a physical quantity and changes it to a digital signal. The data from the required sensors for estimating the road geometry and ego-vehicle motion were selected with the help of the CarMaker reference manual. The measurement data and input data are obtained from the inertial sensors, steering wheel angle sensors, wheel speed

Table 3.2: Sensors with description.

| Sensor | Description | Units |
|---|---|---|
| Vhcl.YawRate | Vehicle yaw velocity | $rads^{-1}$ |
| Vhcl.Steer.Ang | Steering angle at steering wheel | $rad$ |
| Car.ay | Translational acceleration of vehicle connected body without considering gravity in vehicle frame | $ms^{-2}$ |
| Car.ax | Translational acceleration of vehicle connected body without considering gravity in vehicle frame | $ms^{-2}$ |
| Car.vy | Translational velocity of vehicle connected body | $ms^{-1}$ |
| Car.vx | Translational velocity of vehicle connected body | $ms^{-1}$ |
| Car.Road.Lane.Act.Width | Width of the actual roadway lane | $m$ |
| Car.Road.Lane.OnRight.Width | Width of the lane on right side | $m$ |
| Car.Road.Lane.OnLeft.Width | Width of the lane on left side | $m$ |

sensors, and camera sensors. The sensor measurements are based on synchronized data and are sampled at every 0.04 $sec$. The details of the sensors used as the measurements and inputs along with their description can be found in Table 3.2.

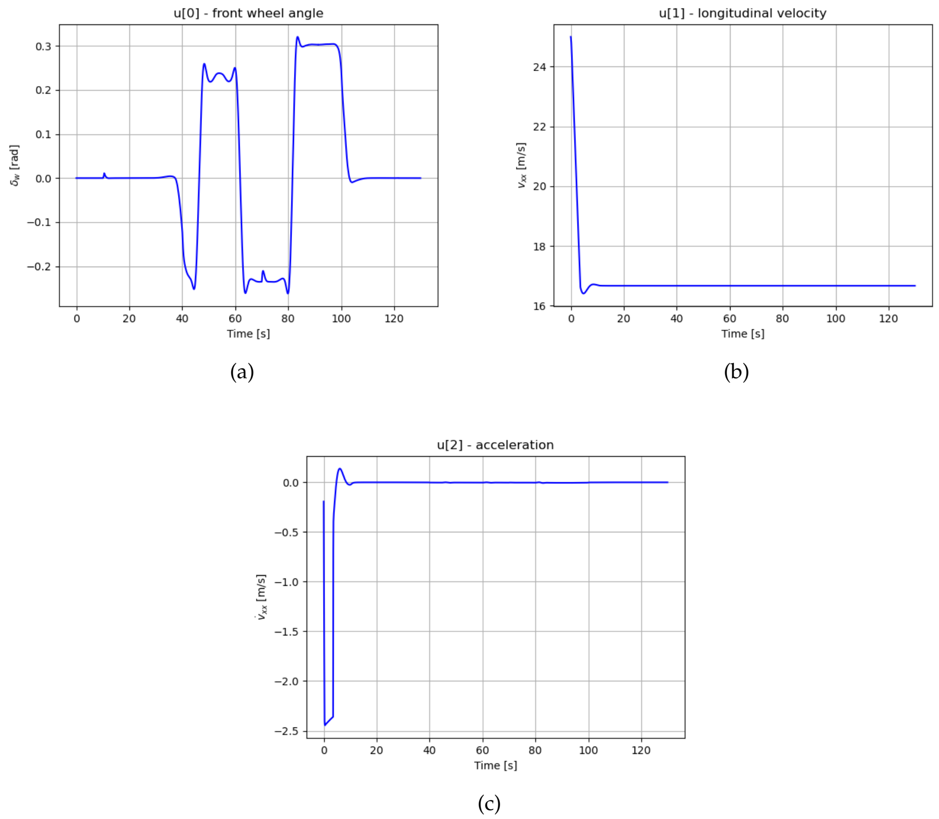
Some of the measurements are measured directly from the sensor. However, few are calculated from the knowledge of other available sensors. The road curvature is calculated from the information of the coordinates of the road segments for both of the scenarios. The distance between the ego vehicle and left lane as well as the angle between the ego vehicle longitudinal axis and lane is calculated from the information of road sensor available in CarMaker manual.

## 3.3 Sensor Data Analysis

The sensor data obtained from different measurements and inputs are analyzed in this section. The input parameters $'u'$ are front-wheel angle ($\delta_{\mathrm{w}}$), longitudinal velocity ($v_{\mathrm{xx}}$), and the acceleration ($\dot{v_{\mathrm{xx}}}$). $\delta_{\mathrm{w}}$ is measured with the help of steering wheel angle sensor and $v_{\mathrm{xx}}$ is measured at the wheels by the wheel speed sensors. $\dot{v_{\mathrm{xx}}}$ is measured by the inertial sensor, or Inertial Measurement Unit (IMU) and assumes completely flat road to avoid gravity leakage. In Figure 3.4, the measurement for front-wheel angle, longitudinal velocity, and acceleration with respect to the $1^{st}$ scenario is presented.

The sensor measurement in Figure 3.4(a) demonstrates that the front wheel angle is constant for the first 40 $sec$ then as the car turns right, the angle is negative and it is positive as the car turns left. It remains constant from 100 $sec$ till the end of the road i.e., 130 $sec$. Figure 3.4(b) shows the measurement for longitudinal velocity and we can observe that initially, the car has the longitudinal velocity of 24.5 $ms^{-1}$ which drops down to a constant velocity of 16.5 $ms^{-1}$ at around 3 $sec$ till the end. The measurement shown in Figure 3.4(c) represents the longitudinal acceleration and it can be noted that between 0 to 3 $sec$, the acceleration is negative because as the speed of the car decreases, we have the negative acceleration and positive as the speed increases which can be seen in the graph from 4 to 5 $sec$. The acceleration is constant from 6 $sec$ till the end because the car moves at a constant speed.

Figure 3.5(a) shows the measurement for the front wheel angle of the car which is changing continuously due to the small perturbations on the drive. We can also observe that as the car turns right, the yaw rate is negative and it is positive as the car turns left. The measurement in Figure 3.5(b) demonstrates that initially, the car has the longitudinal velocity of 24.5 $ms^{-1}$ which decreases to a constant velocity of 16.5 $ms^{-1}$ at around 5 $sec$ till the end. From the measurement in Figure 3.5(c) of longitudinal acceleration, it can be noticed that between 0 to 5 $sec$, the acceleration is negative as the speed of the car decreases, and positive as the



(a)

(b)

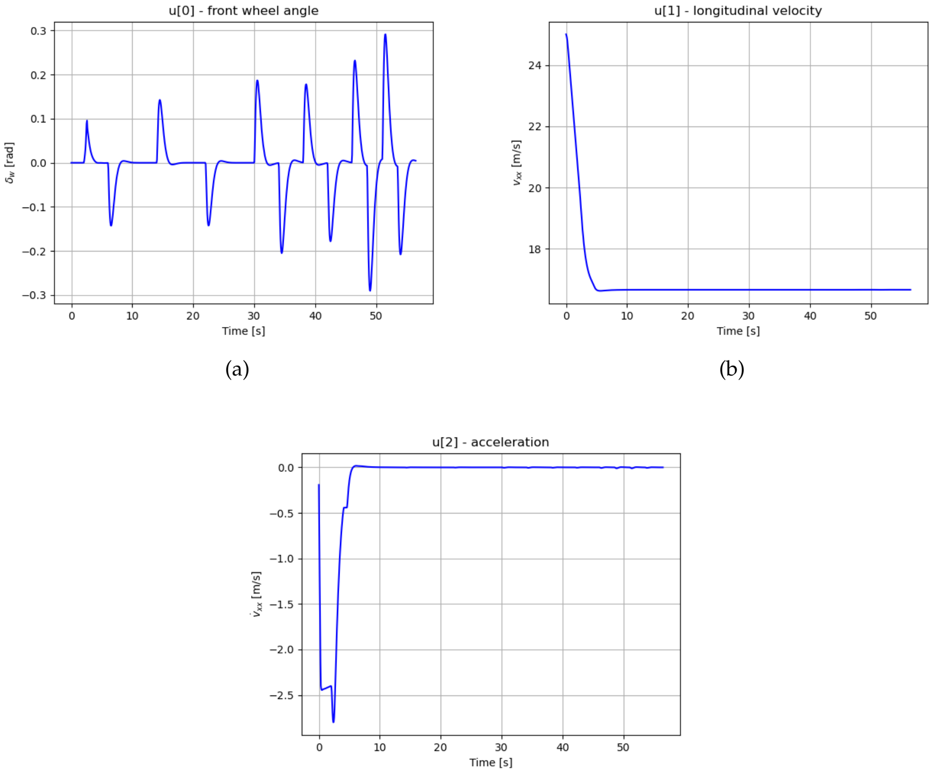
(c)

Figure 3.4: (a) Steering wheel angle (b) Longitudinal velocity (c) Acceleration

speed increases which can be seen in the graph from 4 to 5 $\boldsymbol{sec}$. The acceleration is constant from 6 $\boldsymbol{sec}$ till the end because the car moves at a constant speed.

The measurements $\boldsymbol{'y'}$ obtained from the sensors are the yaw rate $\dot{\boldsymbol{\psi}}_1^{\boldsymbol{m}}$, lateral acceleration $\boldsymbol{a}_{\text{yy}}^{\boldsymbol{m}}$, road curvature $\boldsymbol{c}^{\boldsymbol{m}}$, angle between the longitudinal axis of ego-vehicle and tangent of the road curvature $\boldsymbol{\delta}_2^{\boldsymbol{m}}$, road width $\boldsymbol{w}_2^{\boldsymbol{m}}$, and distance between the left lane and ego-vehicle $\boldsymbol{l}_1^{\boldsymbol{m}}$. In Figure 3.6, the measurements from the sensors for the $\dot{\boldsymbol{\psi}}_1^{\boldsymbol{m}}$ in $\boldsymbol{rad}$, $\boldsymbol{a}_{\text{yy}}^{\boldsymbol{m}}$ in $\boldsymbol{ms}^{-2}$, $\boldsymbol{c}^{\boldsymbol{m}}$ in $\boldsymbol{m}^{-1}$, $\boldsymbol{\delta}_2^{\boldsymbol{m}}$ in $\boldsymbol{rad}$, $\boldsymbol{w}_2^{\boldsymbol{m}}$ in $\boldsymbol{m}$, and $\boldsymbol{l}_1^{\boldsymbol{m}}$ in $\boldsymbol{m}$ for the $1^{st}$ scenario is presented.

Figure 3.6(a) shows the measurement of yaw rate and we can observe from the maneuver of ego-vehicle that for the first 40 $\boldsymbol{sec}$, the yaw rate is constant and as the car move to the right, it is negative and it is positive when the car moves to the left. It is again constant from 101 $\boldsymbol{sec}$ till the end. The lateral acceleration of a car is associated with the lateral consistency of a vehicle. The measurement for lateral acceleration can be observed in Figure 3.6(b) which shows that for the first 40 $\boldsymbol{sec}$, the lateral acceleration is constant and as the car moves to the right, it is negative and it is positive when the car moves to the left and is again constant from 101 $\boldsymbol{sec}$ till the end. In Figure 3.6(c) we can observe the measurement for the road curvature of a winding road. The road curvature is the main component in road geometry. It can be noticed that the road curvature is positive for the left side of the road and negative for the right side of the road. It is zero for the straight road. Figure 3.6(d) demonstrates the measurement for the angle between the ego vehicle longitudinal axis and lane. We can see that as the distance between ego-vehicle and left lane decreases, the angle also decreases, and as the distance increases, the angle also increases. The road width measurement can be seen in Figure 3.6(e) with a constant width of 3.5 $\boldsymbol{m}$ for the complete scenario. In Figure 3.6(f), the measurement for the distance between the ego-vehicle and left lane can be noticed. As soon



(a)

(b)

(c)

Figure 3.5: (a) Steering wheel angle. (b) Longitudinal velocity. (c) Acceleration.

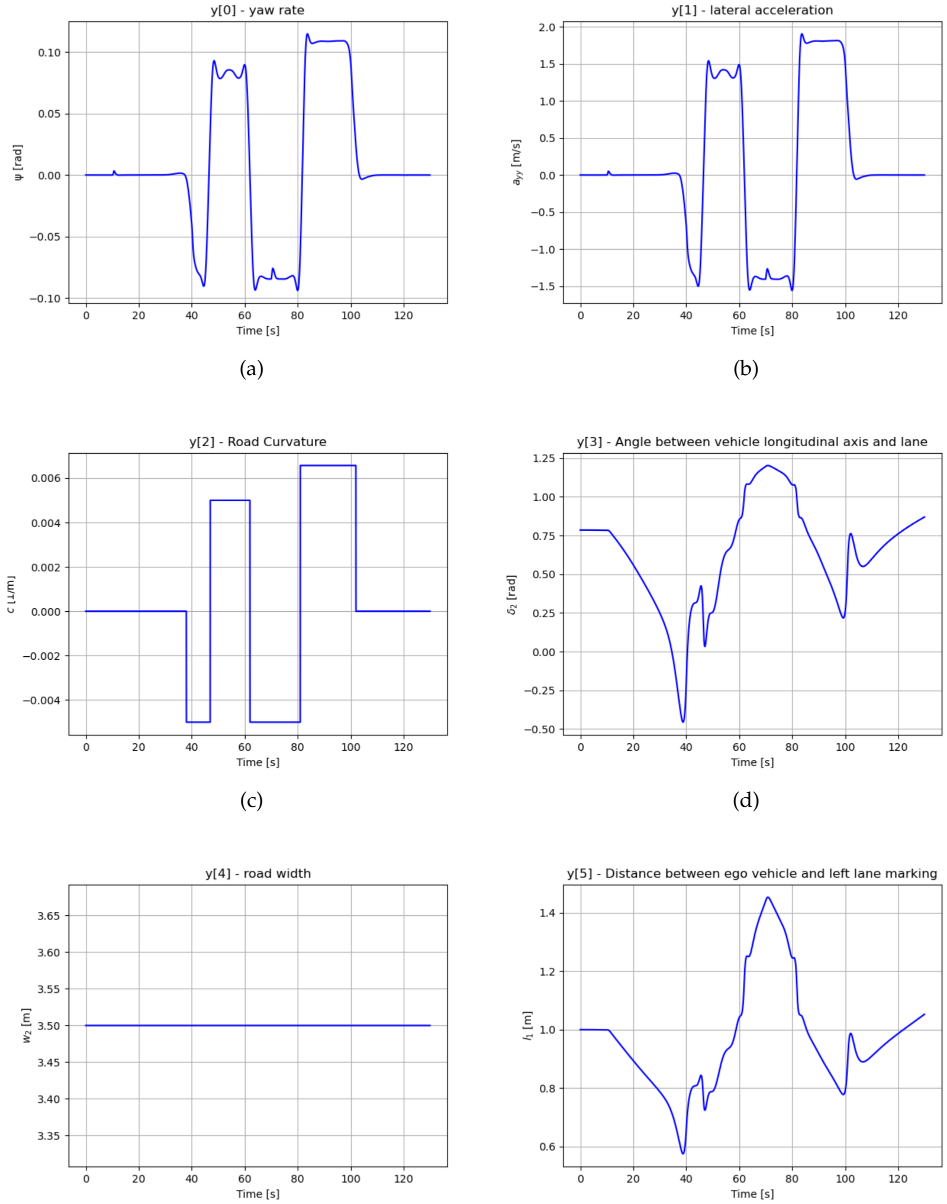
as the car moves to the left, the distance decreases, and as it moves to the right, the distance increases. In figure 3.7, these measurements are shown under the second scenario.

Figure 3.7(a) shows the measurement for yaw rate and we can observe that the yaw rate changes constantly from positive to negative as the car shifts move from left to right side. It can be noticed from the measurement of lateral acceleration in Figure 3.7(b) that as the car moves to the left, the lateral acceleration is positive and it is negative when the car moves to the right and it continuously changes between positive and negative due to the frequent lateral movement of the car. In Figure 3.7(c) the measurement for road curvature of a straight road can be seen. The road curvature is zero as the road is straight without any curve. In Figure 3.7(d), it can be noticed from the measurement of the angle between the ego vehicle longitudinal axis and lane that it changes frequently. When the distance between the ego-vehicle and left lane decreases, the angle also decreases, and as the distance increases, the angle also increases. From Figure 3.7(e), the measurement of the road width can be noticed that the road has a constant width of 3.5 $\boldsymbol{m}$ for the entire scenario. In Figure 3.7(f), the measurement for the distance between the ego vehicle and left lane can be noticed and we can see that it changes constantly. As the car moves to the left, the distance decreases, and as the car moves to the right, the distance increases.

As there is no reference signal for the float angle, $\boldsymbol{\beta_1}$ and angle between the lane and velocity vector of the vehicle, $\delta$ so they are calculated by using the formula [18]

$$\boldsymbol{\beta_1} = \boldsymbol{tan^1}(\frac{v_{\mathrm{yy}}}{v_{\mathrm{xx}}}) \tag{3.1}$$

$$\delta \overset{\Delta}{=} \delta_2 - \boldsymbol{\beta_1} \tag{3.2}$$

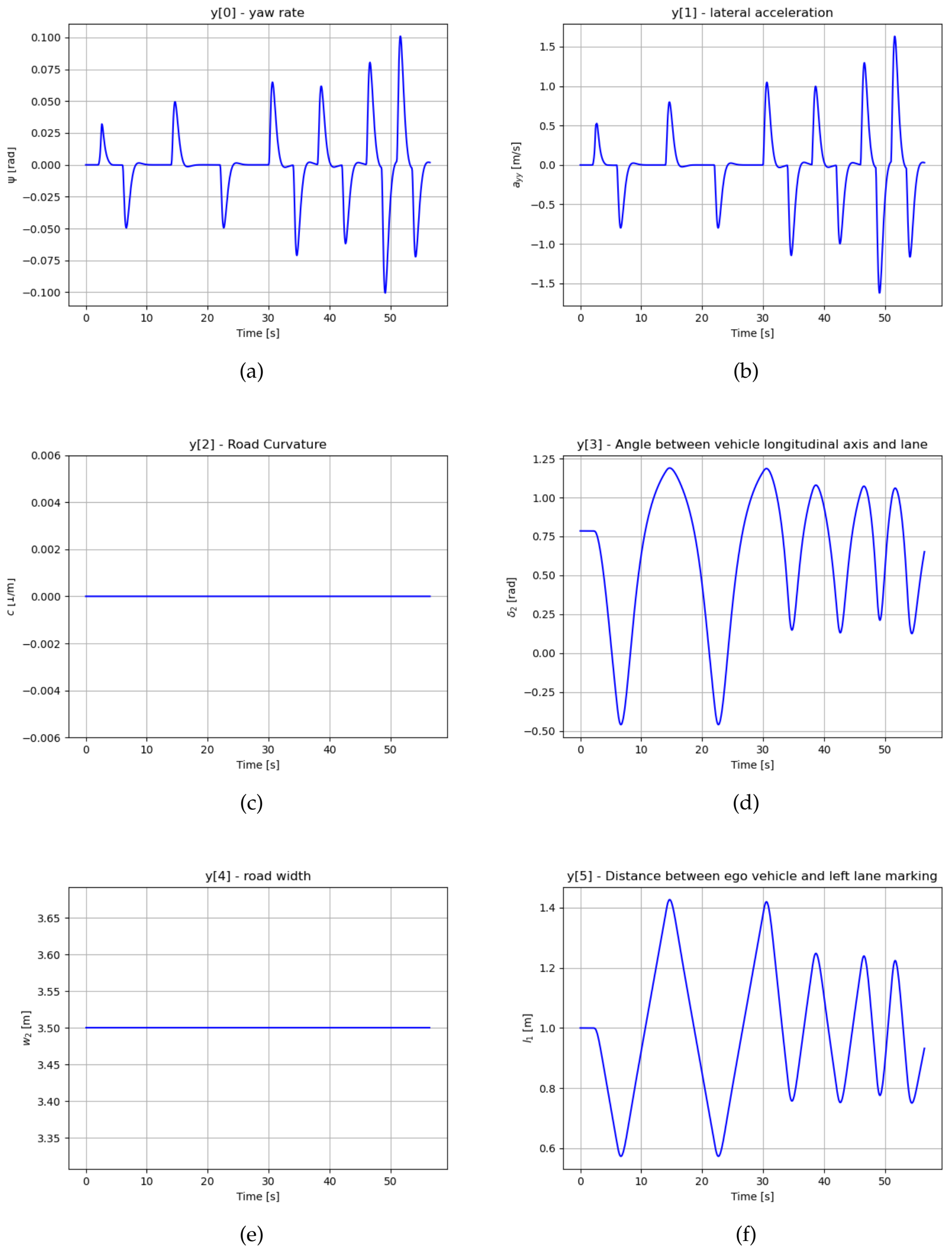


(a) (b)

(c) (d)

(e) (f)

Figure 3.6: (a) Yaw Rate of the Vehicle (b) Lateral Acceleration (c) Road Curvature (d) Angle between longitudinal axis of ego-vehicle and tangent of the road curvature (e) Road width (f) Distance between the left lane and ego-vehicle



(a)

(b)

(c)

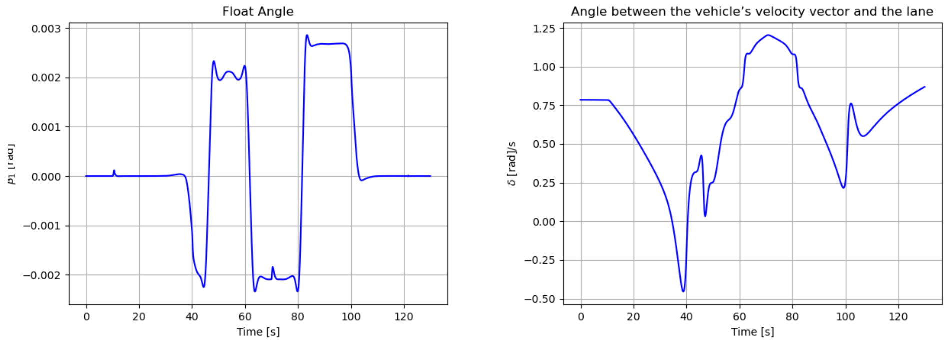
(d)

(e)

(f)

Figure 3.7: (a) Yaw Rate of the Vehicle (b) Lateral Acceleration (c) Road Curvature (d) Angle between longitudinal axis of ego-vehicle and tangent of the road curvature (e) Road width (f) Distance between the left lane and ego-vehicle

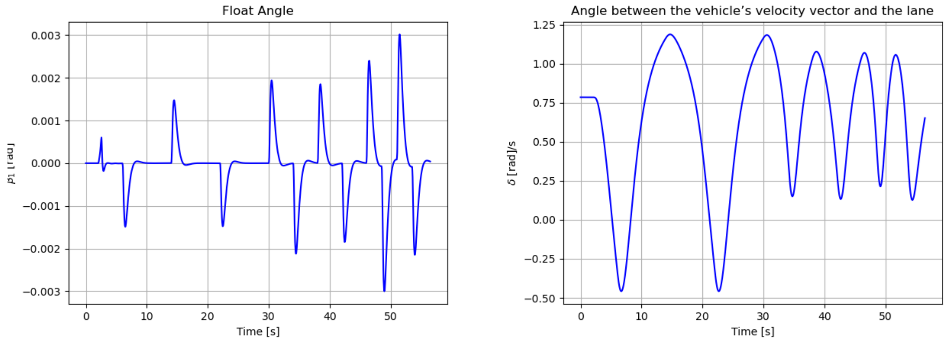
The reference signal for $\boldsymbol{\beta_1}$ and $\delta$ for the first scenario is presented in Figure 3.8.



(a) (b)

Figure 3.8: (a) Float Angle (b) Angle between the lane and velocity vector of the vehicle

The reference signal for $\boldsymbol{\beta_1}$ and $\delta$ for the second scenario is presented in Figure 3.9.



(a) (b)

Figure 3.9: (a) Float Angle (b) Angle between the lane and velocity vector of the vehicle

The measurement data obtained is then fed into the tracker for evaluating the performance and functioning of the algorithm.

# 4 Method

This chapter provides a method used in achieving the state-of-the-art algorithm. The UKF in this chapter implements the dynamic model equations described in chapter 2.

## 4.1 Unscented Kalman Filter (UKF)

As we discussed earlier that for a nonlinear, and non-Gaussian case, we can use either EKF or UKF to solve the problem of non-linearity. In this thesis, we have utilized a UKF method. The UKF is a recursive way of estimating the states and uses the "Unscented Transformation (UT)" technique which requires choosing a set of sampling points called sigma-points. The idea of the UKF consists in choosing the sigma points to calculate the transformation of these points for the state estimation without using the Jacobian calculation. Thus, instead of approximating a nonlinear function $'g'$ by a Taylor series as in EKF, the UKF extracts in a deterministic way the sigma points and passes them through a non-linear function [23].

## 4.2 Unscented Transformation (UT)

UT allows propagating the point sets known as the sigma points through the non-linear state transition and measurement functions. The new estimate, as well as the error covariance, are then calculated by averaging over the sigma-point [4]. The algorithm in this approach makes use of these sigma points as an input to the state transition function and measurement function to obtain a new state of the transformed points. The sigma points spread can be determined through the dimension of the state space vector [28]. Figure 4.1 shows the UT in UKF.

## 4.3 UKF Algorithm

The flow chart of the algorithm is presented in Figure 4.2.

First the initialization of filter is performed with $\boldsymbol{x_0}$ as the initial state vector, and $\boldsymbol{P_0}$ as the initial state covariance. The parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\kappa}$ determines the spread of the sigma points. Then the sigma points are generated which are propagated through non-linear state transition function to obtain the predicted state and covariance, and measurement function to obtain

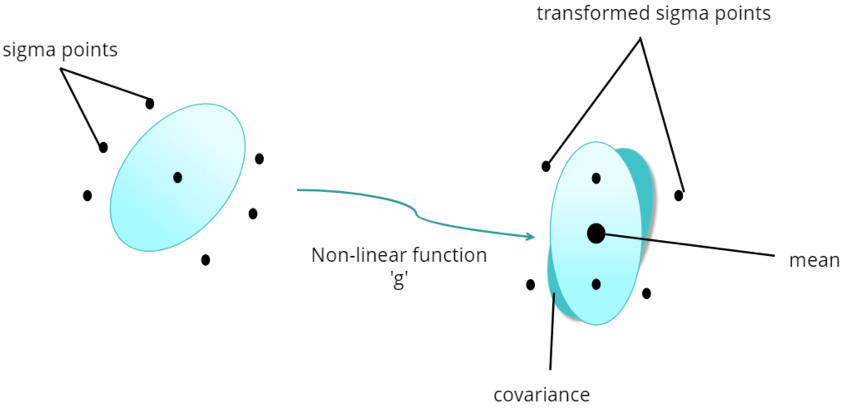


Figure 4.1: Unscented Transformation in UKF

the predicted measurement and covariance. From the predicted state and measurement, the estimate of cross-covariance is obtained to calculated the Kalman gain which is then used to estimate the state and covariance at time step $k$ and the process is repeated.

For the initialization of filter, we first need to define a state-space vector $\hat{x}$ and the equations [27] used for the UKF are provided here

$$\hat{x}_k = x_0 \tag{4.1}$$

A new matrix $\chi$ composed of $(2N + 1)$ sigma points is formed. The amount of sigma points depends upon the state space vector dimensionality $N$ [13]. The first sigma point [28] is given by

$$\chi_k = \hat{x}_k \tag{4.2}$$

which is simply the state space vector in our case. Each column in $\chi$ contains $(2N + 1)$ sigma points for one dimension. The $(2N + 1)$ weights which can be at the positive or negative side of the sigma points [33] are calculated as

$$\Delta x^{\mathrm{i}} = (\sqrt{qP_k})_i \qquad for \qquad i = 1, 2, \ldots, N$$

$$\Delta x^{\mathrm{N+i}} = -(\sqrt{qP_k})_i \qquad for \qquad i = 1, 2, \ldots, N$$

For computing the delta, Cholesky matrix square root is used

$$\chi^i_{\mathrm{k\text{-}1}} = \chi_{\mathrm{k}} + \Delta x^{\mathrm{i}} \qquad for \qquad i = 1, 2, \ldots, 2N$$

where $P_k$ is the state covariance matrix and $q = \alpha^2(N + \kappa)$. $\alpha$ and $\kappa$ are the scaling parameters in UKF. These scaling parameters defines how far sigma points are from the mean. A default choice of $\kappa$ is $\kappa = 0$ and $\alpha$ should be chosen between 0 and 1 as it determines the spread of the sigma points [32]. The parameter $\beta$ can be used to represent additional higher orders on the distribution of the representation of the Gaussian. $\beta$ is usually chosen to be equal to 2. Each sigma point has two weights associated with it [29]: $W_m$ which is used for the calculation of the mean and $W_c$ used for the calculation of covariance. The weighted mean $W_m$ is equal to:

$$W^0_m = 1 - \frac{N}{q} \tag{4.3}$$

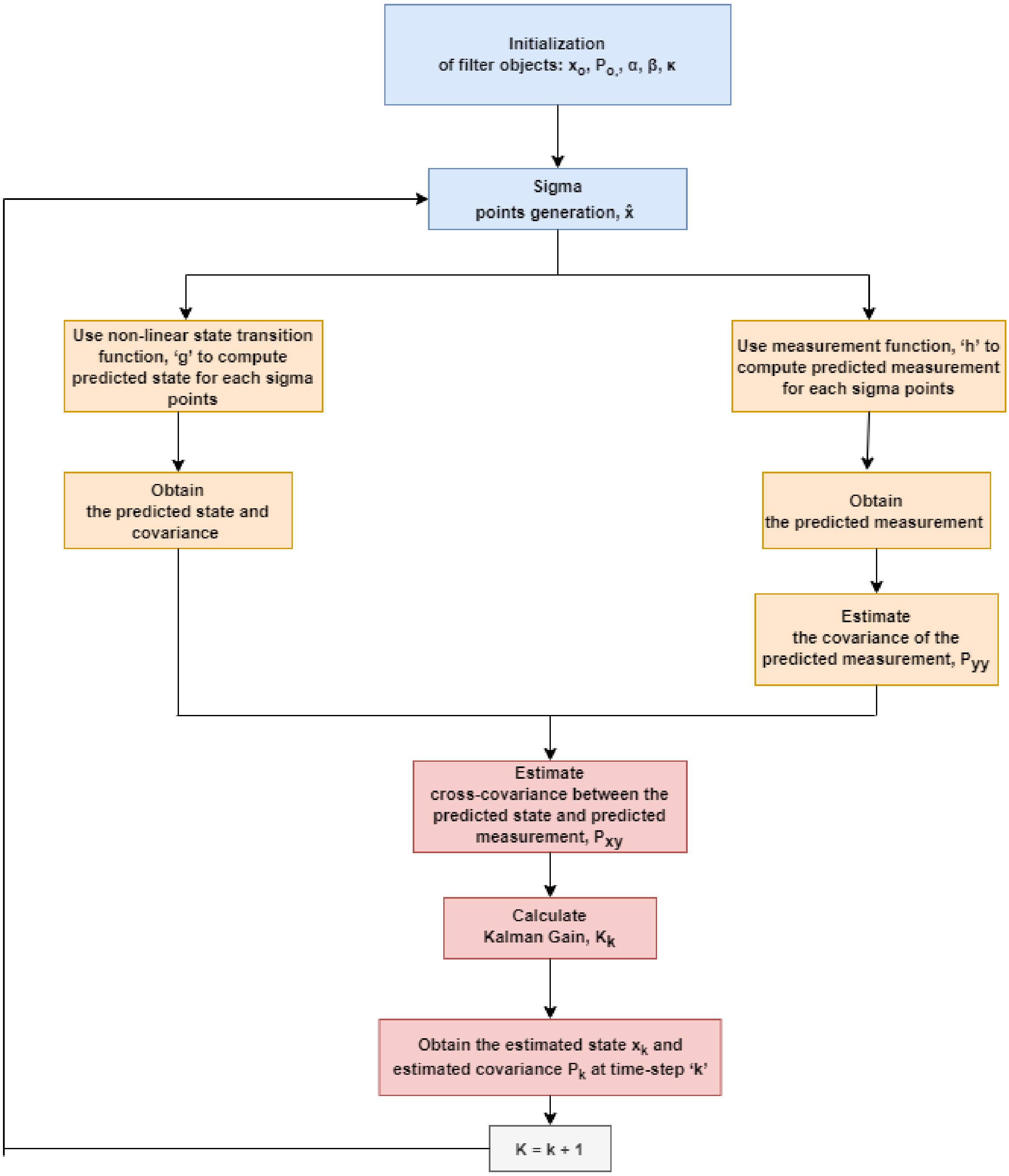

Initialization of filter objects: x$_o$, P$_o$,, α, β, κ
Sigma points generation, x̂
Use non-linear state transition function, 'g' to compute predicted state for each sigma points
Use measurement function, 'h' to compute predicted measurement for each sigma points
Obtain the predicted state and covariance
Obtain the predicted measurement
Estimate the covariance of the predicted measurement, P$_{yy}$
Estimate cross-covariance between the predicted state and predicted measurement, P$_{xy}$
Calculate Kalman Gain, K$_k$
Obtain the estimated state x$_k$ and estimated covariance P$_k$ at time-step 'k'
K = k + 1

Figure 4.2: Flow Chart of the Algorithm

$$W_m^i = \frac{1}{2q} \qquad for \qquad i = 1, 2, \ldots, 2N$$

The weighted covariance $W_c$ is equal to:

$$W_c^0 = (2 - \alpha^2 + \beta) - \frac{N}{q} \tag{4.4}$$

$$W_c^i = \frac{1}{2q} \qquad for \qquad i = 1, 2, \ldots, 2N$$

## 4.4 Predict Step

In the prediction step, we take the generated sigma points and propagate them through a nonlinear function $g$ and compute the predicted state for the sigma points [6].

$$\chi_{\text{k-1}}^{'i} = g(\chi_{\text{k-1}}^{i}, u_{\text{k}}) \qquad for \qquad i = 1, 2, \ldots, 2N$$

The variables of vehicle dynamics, state-space model, and inputs are the parameters used in the non-linear function $g$. These parameters are described briefly in Table 2.1.
The predicted state at next time step is provided by

$$\chi_{\text{k-1}}^{'} = \sum_{i=0}^{2M} W_m^i \chi_{\text{k-1}}^{'i} \tag{4.5}$$

and the covariance of the predicted state is given by

$$P_{\text{x}} = \sum_{i=0}^{2M} W_c^i (\chi_{\text{k-1}}^{'i} - \chi_{\text{k-1}}^{'})(\chi_{\text{k-1}}^{'i} - \chi_{\text{k-1}}^{'})^T + Q_{\text{k}} \tag{4.6}$$

where Q is the process noise covariance matrix.

## 4.5 Update Step

In the update step, we take the measurement function $h$ and apply it again to the sigma points to convert the sigma points from predicting step into measurement space. The measurements predicted for every sigma point [29] is given by:

$$Y_{\text{k-1}}^{'i} = h(\chi_{\text{k-1}}^{'i}, u_{\text{k}}) \qquad for \qquad i = 1, 2, \ldots, 2N$$

Predicted measurements at next time step is given by:

$$Y_{\text{k}}^{'} = \sum_{i=0}^{2M} W_m^i Y_{\text{k-1}}^{'i} \tag{4.7}$$

The estimated covariance of the predicted measurement is given by

$$P_{\text{y}} = \sum_{i=0}^{2M} W_c^i (Y_{\text{k-1}}^{'i} - Y_{\text{k}}^{'})(Y_{\text{k-1}}^{'i} - Y_{\text{k}}^{'})^T + R_{\text{k}} \tag{4.8}$$

where R is the measurement noise covariance matrix. The estimated cross-covariance between the predicted measurement and state is given by

$$P_{\text{xy}} = \frac{1}{2q} \sum_{i=0}^{2M} (\chi_{\text{k-1}}^{'i} - \chi_{\text{k-1}}^{'})(Y_{\text{k-1}}^{'i} - Y_{\text{k}}^{'})^T \tag{4.9}$$

The Kalman gain is the combination of covariance error of estimate and covariance error of measurement and is computed by

$$K = P_{\text{xy}} P_{\text{y}}^{-1} \tag{4.10}$$

Finally, the estimated state and the estimated error covariance at $\boldsymbol{k}$ time step is computed by

$$\hat{x}_{\mathrm{k}} = \chi^{'}_{\mathrm{k\text{-}1}} + \boldsymbol{K}(\boldsymbol{Y}_{\mathrm{k}} - \boldsymbol{Y}^{'}_{\mathrm{k}}) \tag{4.11}$$

$$\boldsymbol{P}_{\mathrm{k}} = \boldsymbol{P}_{x} - \boldsymbol{K}\boldsymbol{P}_{Y}\boldsymbol{K}^{T} \tag{4.12}$$

At each time-step, the filter re-estimates the state as the measurements are continuously received by the filter and the state is changed because of state-space model. By accumulating all the measurements, the better estimate of a state is obtained. This concludes the details and description of the UKF used in the thesis.

# 5 Results and Discussion

This chapter discusses the results achieved from our approach. The data used for evaluation of the our algorithm is described in chapter 3.

## 5.1 Initialization of the filter parameters

To initialize the filter, the parameters of UKF (initial state $\boldsymbol{x_0}$, initial state covariance $\boldsymbol{P_0}$, measurement noise covariance matrix $\boldsymbol{R}$, and process noise covariance matrix $\boldsymbol{Q}$) are set to some initial values. The initial value for $\boldsymbol{x_0}$, and $\boldsymbol{P_0}$ are set to a small value. The measurement noise covariance matrix $\boldsymbol{R}$ and process noise covariance matrix $\boldsymbol{Q}$ are the tuning parameters in this thesis. The covariance matrices are the uncertainties that we expect in our system. The variances for all parameters were investigated because each parameter would have a completely different scale of variance e.g. if we have 1 variance for road width which is very small but for yaw rate variance of 1 would be a huge uncertainty and if the variance is set to infinity then it means that we are very uncertain about the initial conditions of our model. One value may be small for some parameters and high for other parameters. So, the initial conditions are fixed carefully to get a good output of the filter.

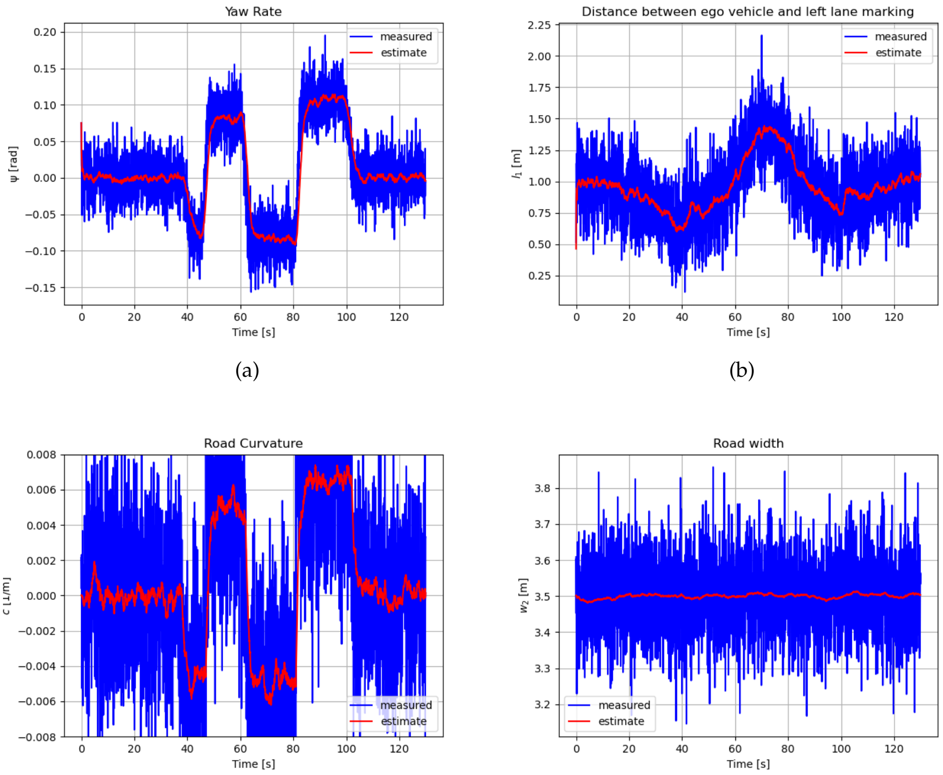
## 5.2 Testing of the filter

When the filter is implemented, the next step is to investigate the functioning of the filter by conducting experiments. Some experiments which are performed to test the filter are: adding noise to the measurements, reducing the Q matrix, and increasing the R matrix. The details of the tests and their results are discussed in the following sub-sections.

### Adding noise to the measurements

We test the performance of our filter by adding some realistic amount of noise to all the measurements fed to the tracker. In this experiment, Gaussian noise is added with zero mean and different parameters have different noise variances. Noise is added to investigate the sensitivity of the filter because when the filter receives the noisy input then it will produce noisy output but in reduced form. If there is no noise in the measurements then the data would be so clean and the need to estimate the parameters jointly would not be required.

The joint effect can be only be observed if the data has a more realistic amount of noise and then we can observe the real benefits of the filter e.g. when a vehicle is turning then its heading would help in estimating the road curvature and the curvature in return will help in the heading of the vehicle. The results are shown in Figures 5.1 and 5.2 for scenario 1 scenario 2 respectively. We have compared the measured and estimated output for yaw rate $\dot{\psi}_1^m$, road curvature $c^m$, distance/offset between the ego-vehicle and left lane $l_1^m$, road width $w_2^m$ as the system contains both the estimates and measurements for these parameters, and by observing the results we can see that even with the addition of noise, we can still achieve better tracking performance as expected.



(a)

(b)

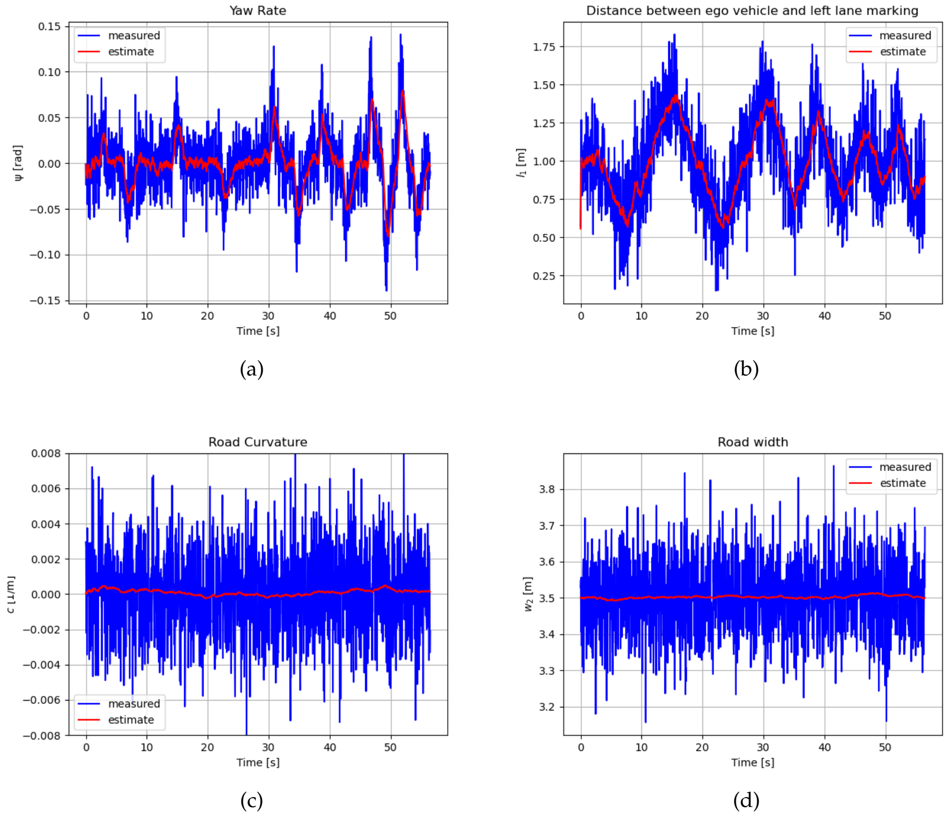
(c)

(d)

Figure 5.1: (a) Yaw Rate (b) Distance between the left lane and ego-vehicle (c) Road Curvature (d) Road width

### Reducing Q matrix

We have also tested the filter performance by reducing the process noise covariance matrix $Q$ in the prediction step. The results are shown in Figures 5.3 and 5.4 for scenario 1 and scenario 2. As we reduce the $Q$ by dividing it by some large number, then our measurement model assumes that quite less noise is received by the filter. As the prediction is the most important step in the implementation, so our filter will follow the prediction and will ignore the measurements. This can be observed from the results shown in Figures 5.3 and 5.4 that by reducing $Q$, more filtering is introduced and the filter follows the prediction model with less confidence in measurements, and the expected results are obtained. So, we can say that the higher the value of $Q$, the better will be the tracking of the data and noise; and with lower $Q$, the better will be the rejection of the noise and ignoring the measurement data, hence smoother estimates would be achieved.



(a)

(b)

(c)

(d)

Figure 5.2: (a) Yaw Rate (b) Distance between the left lane and ego-vehicle (c) Road Curvature (d) Road width

### Increasing R matrix

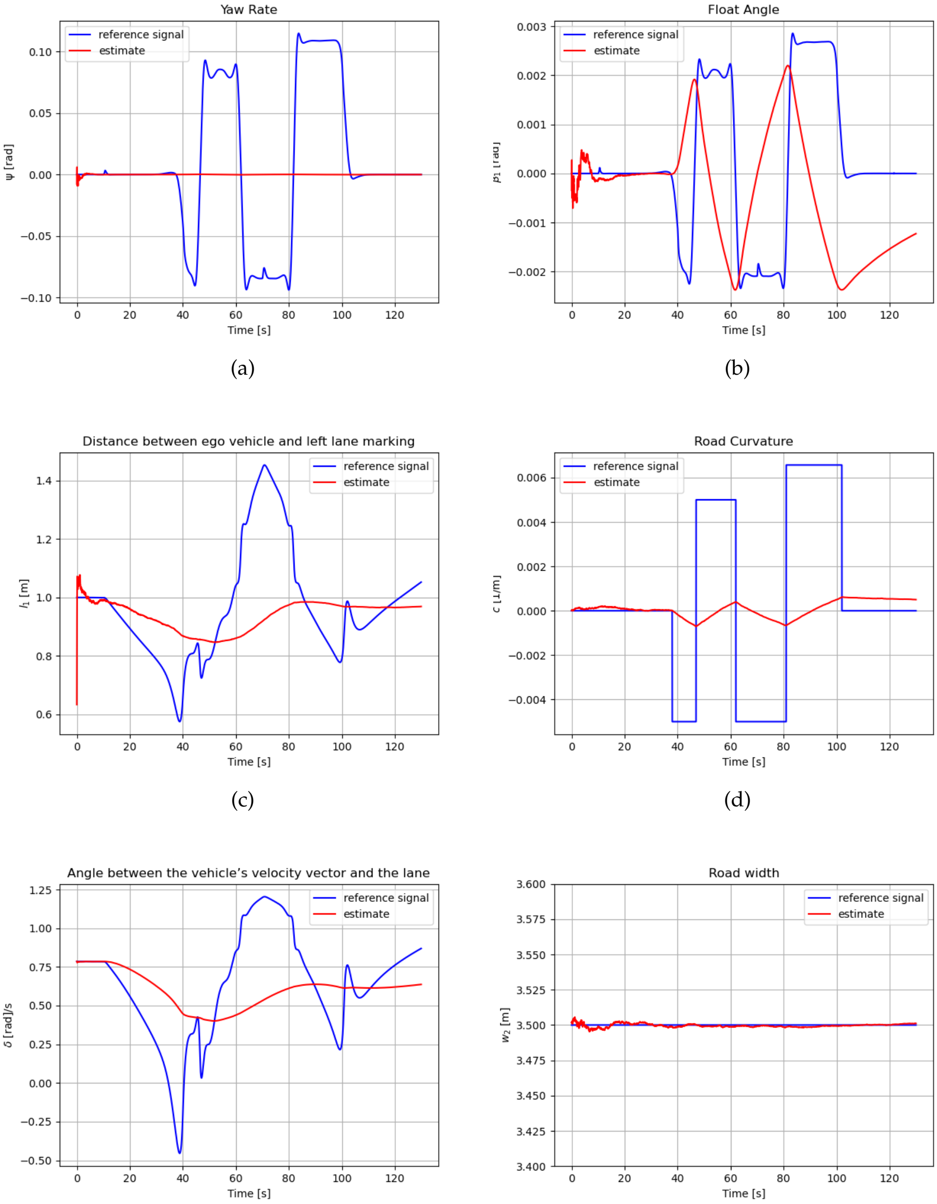
We can test the filter performance by increasing the measurement noise covariance matrix $\boldsymbol{R}$ by multiplying the matrix with some large numbers in the update step. When we increase $\boldsymbol{R}$, it means that this measurement will not be used in the update and the filter only trusts the predicted model and in this case, the measurements are ignored. The results are shown in figure 5.5 and 5.6 for scenario 1 scenario 2, respectively.

More noisy observation means less accurate estimates, and rarely trusted by the filter, therefore it trusts more in the prediction of the current state based on the previous state and current dynamics. We can also observe in 5.5 and 5.6 that as $\boldsymbol{R}$ increases, the measurements received by the filter matter less, and only the prediction is received as expected.

## 5.3 Comparison between Estimated and Reference Signals

A comparison has been performed between the estimated and the reference signals to examine how good the estimates of our states are for both of the scenarios. Figure 5.7 shows the comparison results from the 1$^{st}$ scenario.

As we can observe in Figure 5.7(a) that the estimate of $\dot{\boldsymbol{\psi}}_1$ matches the reference signal very well. In Figure 5.7(b), we can see the estimate for $\boldsymbol{\beta_1}$ and a small amount of variation can be observed between the estimated and reference signal. Figures 5.7(c) and (d) represents the estimate for $\boldsymbol{l_1}$ and $\boldsymbol{c}$ and we can see that the estimated signal has a very less amount of noise and matches the reference signal well. Figures 5.7(e) and (f) represents $\delta$, and $\boldsymbol{w_2}$ and by



(a)

(b)

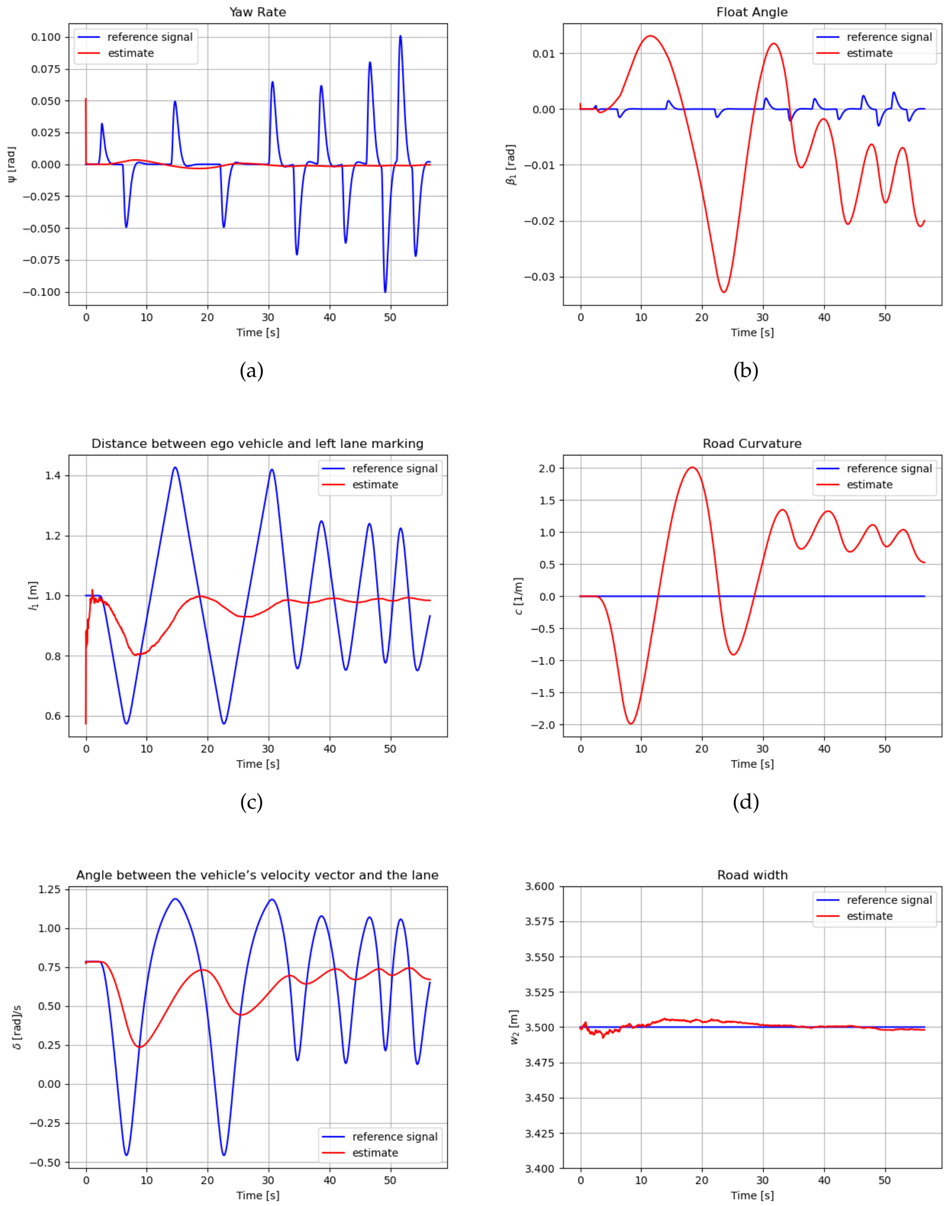
(c)

(d)

(e)

(f)

Figure 5.3: (a) Yaw Rate (b)Float angle (c) Distance between the left lane and ego-vehicle (d) Road Curvature (d) Angle between the lane and velocity vector of the vehicle (e) Road width

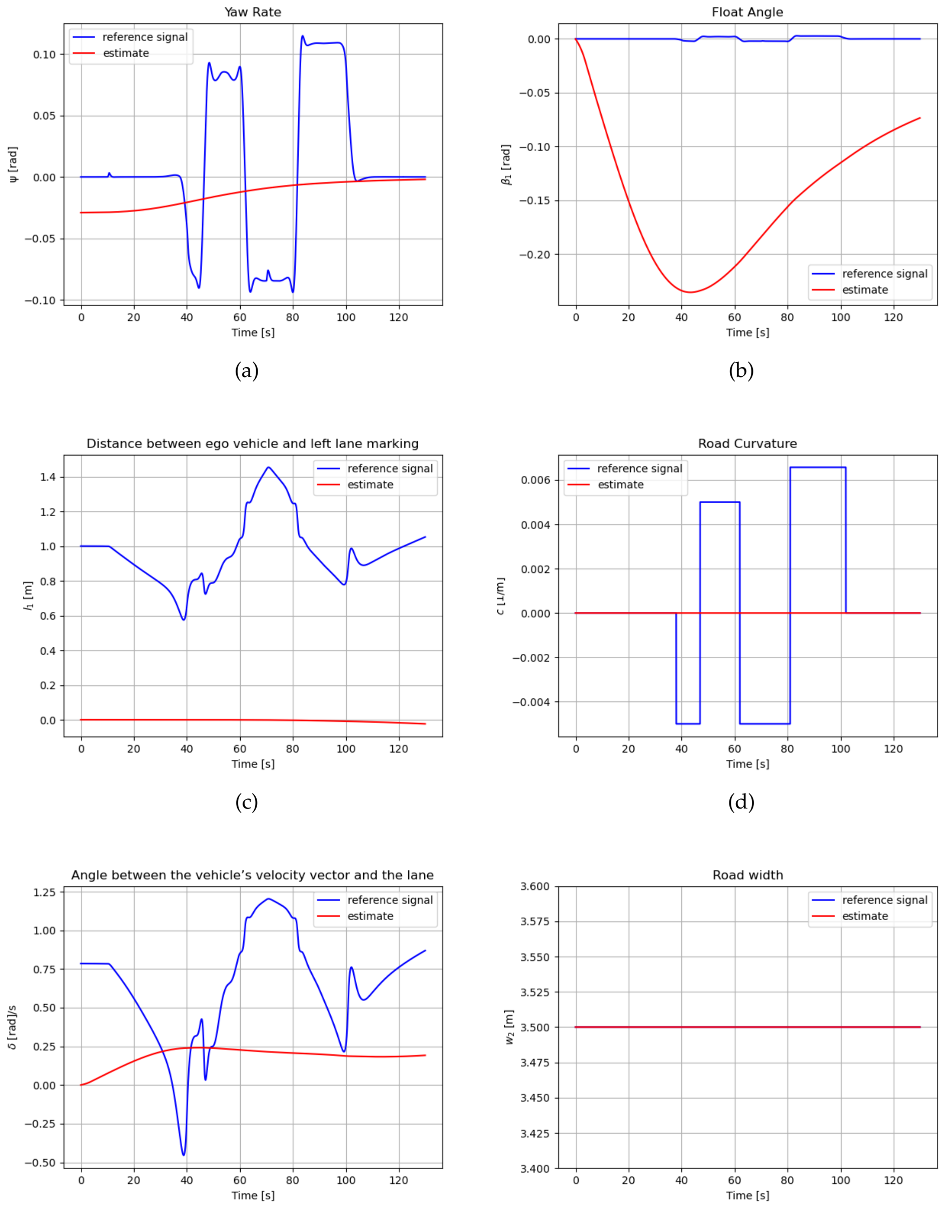


(a) (b)

(c) (d)

(e) (f)

Figure 5.4: (a) Yaw Rate (b)Float angle (c) Distance between the left lane and ego-vehicle (d) Road Curvature (d) Angle between the lane and velocity vector of the vehicle (e) Road width



(a)

(b)

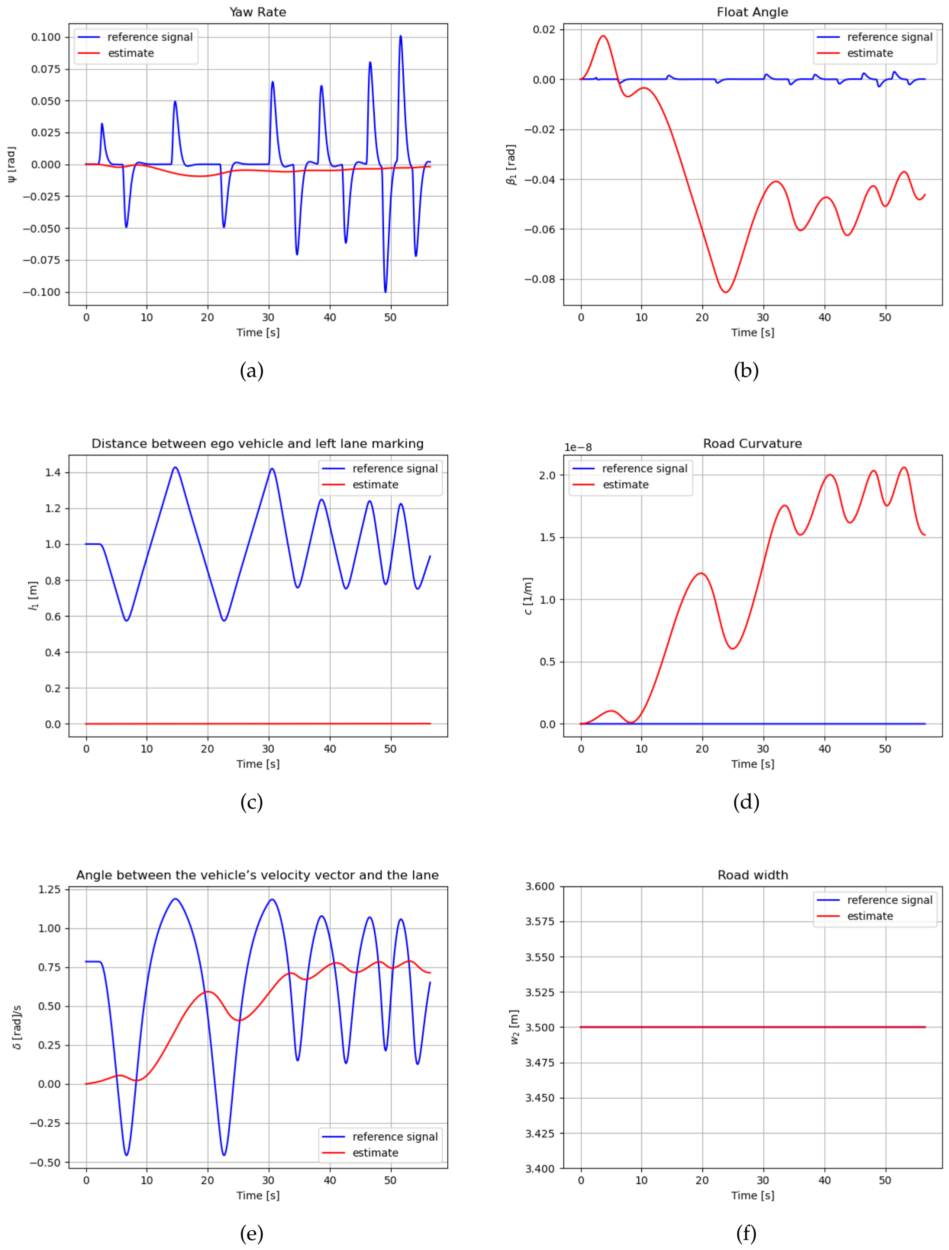
(c)

(d)

(e)

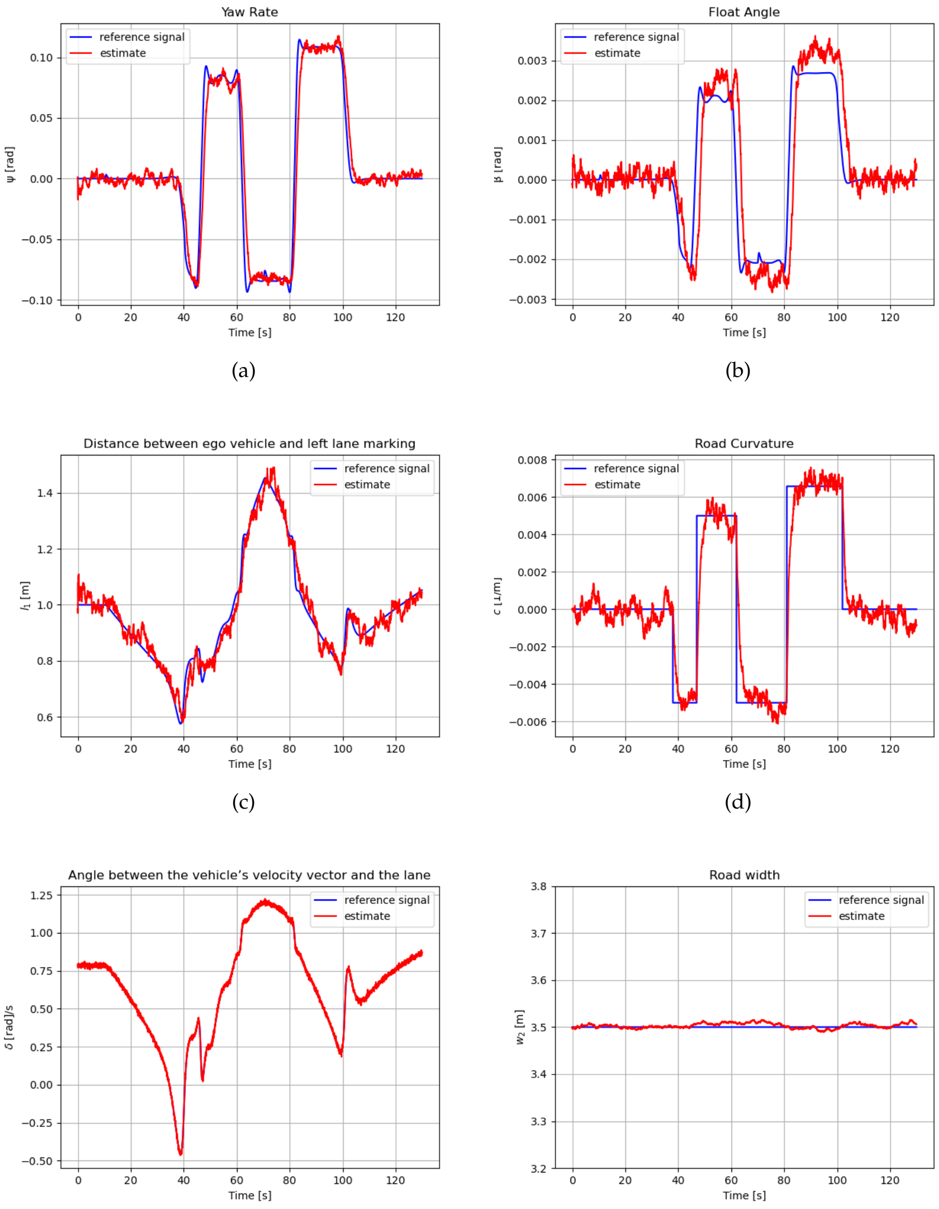
(f)

Figure 5.5: (a) Yaw Rate (b)Float angle (c) Distance between the left lane and ego-vehicle (d) Road Curvature (d) Angle between the lane and velocity vector of the vehicle (e) Road width



(a) (b) (c) (d) (e) (f)

Figure 5.6: (a) Yaw Rate (b)Float angle (c) Distance between the left lane and ego-vehicle (d) Road Curvature (d) Angle between the lane and velocity vector of the vehicle (e) Road width



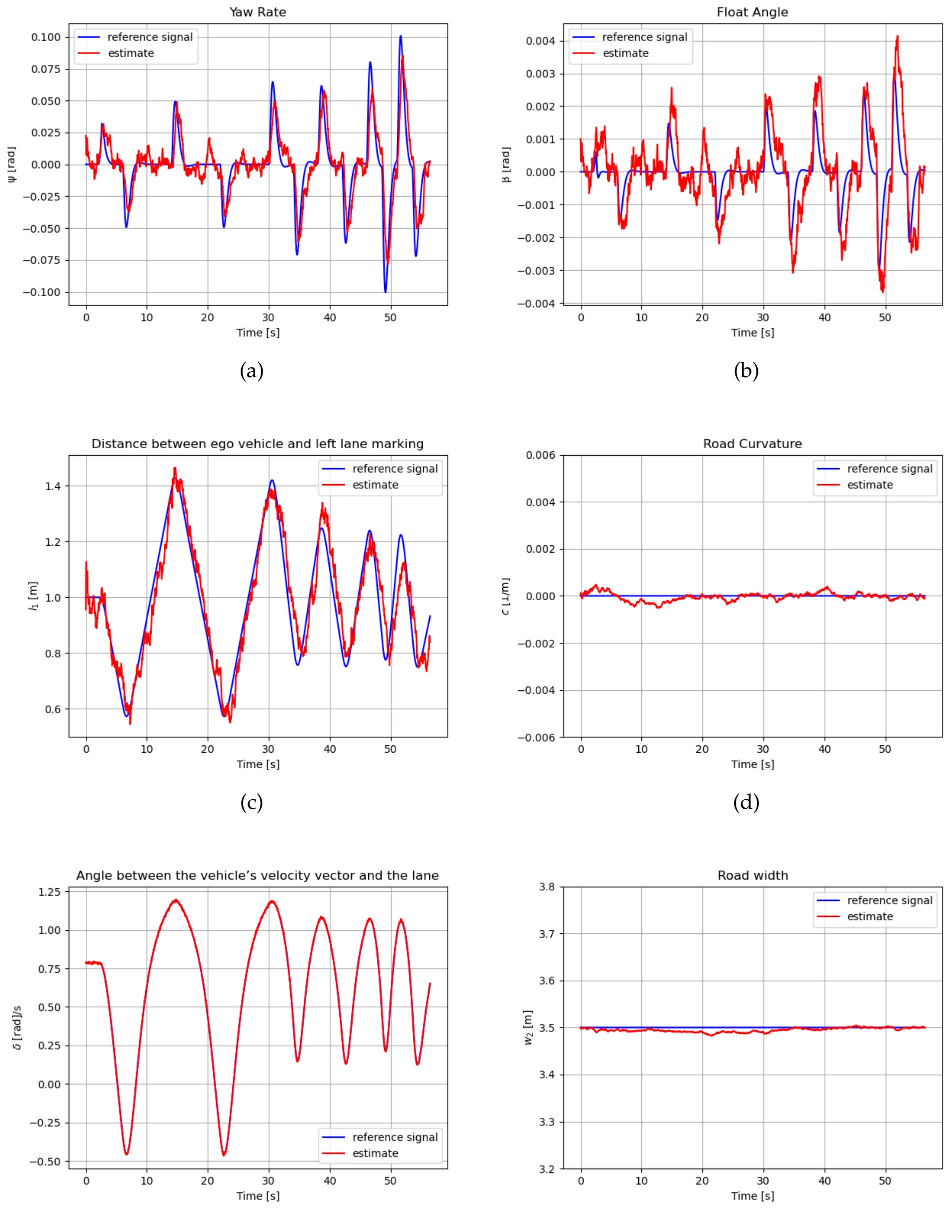
(a) (b)

(c) (d)

(e) (f)

Figure 5.7: (a) Yaw Rate (b) Float angle (c) Distance between the left lane and ego-vehicle (d) Road Curvature (d) Angle between the lane and velocity vector of the vehicle (e) Road width

observing the results we can see that the estimated signal matches the reference signal very well for both of the estimates.
Figure 5.8 shows the comparison results from $2^{nd}$ scenario.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.8: (a) Yaw Rate. (b)Float angle. (c) Distance between the left lane and ego-vehicle (d) Road Curvature. (d) Angle between the lane and velocity vector of the vehicle. (e) Road width.

As we can see in Figure 5.8(a) that the estimate of $\dot{\psi}_1$ matches the reference signal well with a small amount of difference. In Figure 5.8(b), we can observe the estimate for $\beta_1$ with a very less variation between the estimated and reference signal. Figures 5.8(c) and (d) represent the estimate for $l_1$ and $c$ and we can see that the estimated signal has a very less amount of noise and matches the reference signal well. Figures 5.8(e) and (f) represents $\delta$, and $w_2$ and

Table 5.1: RMSE for $1^{st}$ and $2^{nd}$ scenario.

| Parameters | RMSE for $1^{st}$ scenario | RMSE for $2^{nd}$ scenario |
|---|---|---|
| $\dot{\psi}_1$ | 0.01064 | 0.01435 |
| $\beta_1$ | 0.001104 | 0.000411 |
| $l_1$ | 0.04126 | 0.06569 |
| $c$ | 0.0010937 | 0.0001597 |
| $\delta$ | 0.008836 | 0.008781 |
| $r_2$ | 0.004761 | 0.005184 |

by observing the results we can see that the estimated signal matches the reference signal very well for both of the estimates.

From these results, it can be interpreted that the simulated estimates matches the reference signals better in all of the estimates. We can also notice that the behavior of the algorithm is almost similar in both of the scenarios. It can also be noted that the process of tuning the filter parameters plays an important role in estimating the parameters and finding the desired performance of the filter. These results also indicate that the UKF can estimate the noisy measurements.

## 5.4 Performance Evaluation

The performance of the joint algorithm is evaluated by calculating the RMSE between the measured and estimated values. The RMSE is used to discover any errors in the filter. The lower the value of RMSE, the better will be the performance of the filter. The formula used for calculating RMSE is:

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{N}(Y_i - \hat{x}_i)^2} \tag{5.1}$$

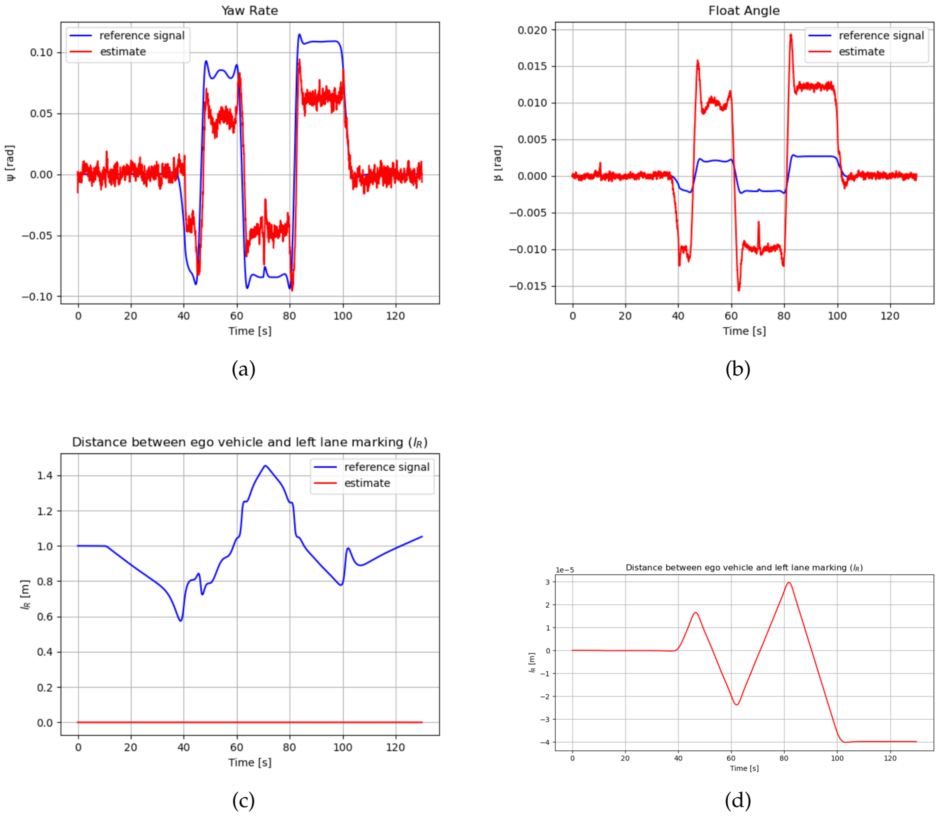
where $M$ is the number of measurement observations, $Y$ is the reference measurements, and $\hat{x}$ is the estimates.

Table 5.1 shows the RMSE with first column representing the RMSE for the $1^{st}$ scenario and the second column representing the RMSE for $2^{nd}$ scenario for $\dot{\psi}_1$, $\beta_1$, $l_1$, $c$, $\delta$, and $w_2$. From these results, we can observe that the RMSE for $\dot{\psi}_1$ is slightly more in $2^{nd}$ scenario as compare to $1^{st}$. The RMSE for $\beta_1$ in $1^{st}$ scenario is almost three times larger as compare to the $2^{nd}$. Similarly, when comparing the RMSE for $l_1$ in both scenarios, we can see that scenario $1^{st}$ has comparatively less RMSE. For $c$, a slight difference can be observed between scenario $1^{st}$ and $2^{nd}$. An unobservant difference can be noticed for $\delta$ in both of the scenarios and for $w_2$ a slightly larger RMSE can be seen in the $2^{nd}$ scenario. We can infer from these results that with small RMSE, the accuracy, and performance of our filter is good.

## 5.5 Performance Comparison

Our algorithm is compared with the baseline algorithm for evaluating the performance of the approach. The baseline algorithm is designed from the current algorithm by estimating the parameters of the ego-vehicle and road model separately. We divided these algorithms into three parts and compared them. The details of the algorithm and their results are discussed below:

**Algorithm 1:** The algorithm 1 represents the jointly ego-motion of vehicle and road geometry estimation. The details of this algorithm can be found in chapter 2 and chapter 4.



(a)

(b)

(c)

(d)

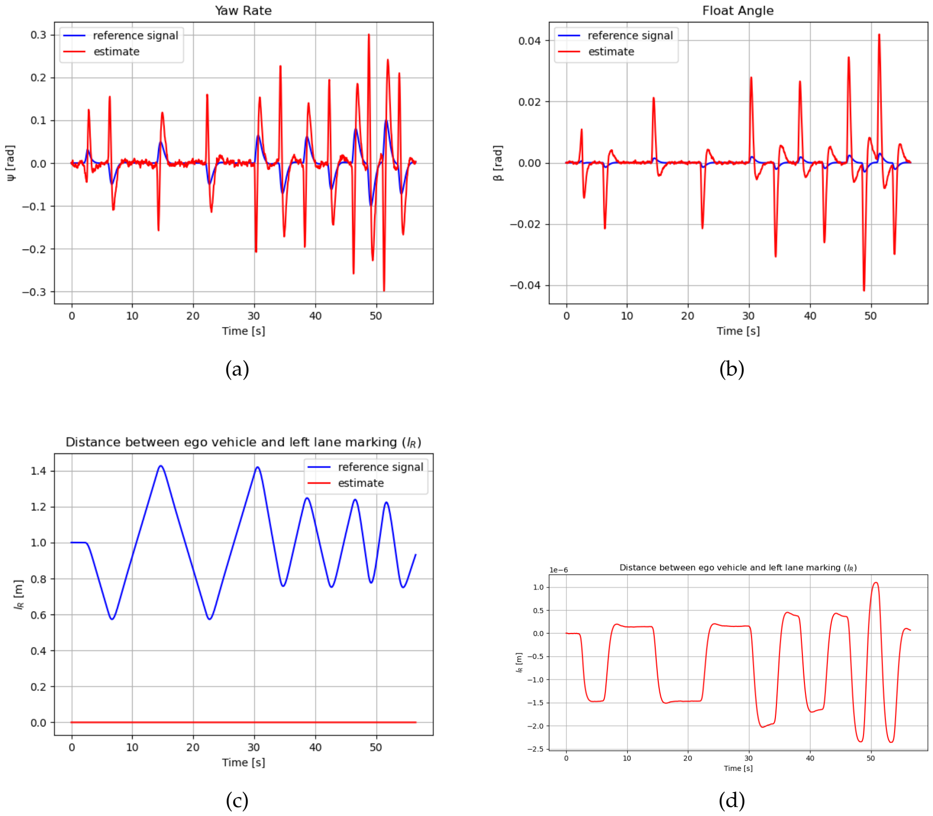
Figure 5.9: (a) Yaw Rate. (b) Float angle.(c) Distance between ego-vehicle and left lane. (d) Zoomed $l_1$

**Algorithm 2:** The algorithm 2 represents the independent parameter estimation for the ego-vehicle model. The parameters in algorithm 2 consist of yaw rate $\dot{\psi}_1$, float angle $\beta_1$ and distance between ego-vehicle and left lane $l_1$. The state vector for the ego-vehicle model is described in equation 2.5. The non-linear state-space model for only ego-vehicle can be represented as

$$g_1 = \begin{pmatrix} \beta_1 \frac{-C_{\text{fa}} l_{\text{fa}} cos\delta_w + C_{\text{ra}} l_{\text{ra}}}{I} - \dot{\psi}_1 \frac{C_{\text{fa}} l_{\text{fa}}^2 cos\delta_w + C_{\text{ra}} l_{\text{ra}}^2}{I v_{\text{xx}}} + \frac{C_{\text{fa}} l_{\text{fa}} tan\delta_w}{I} \\ -\beta_1 \frac{C_{\text{fa}} cos\delta_w + C_{\text{ra}} + v_{\dot{x}\text{x}} m}{m v_{\text{xx}}} + \dot{\psi}_1 \left(1 + \frac{C_{\text{fa}} l_{\text{fa}} cos\delta_w - C_{\text{ra}} l_{\text{ra}}}{m v_{\text{xx}}^2}\right) + \frac{C_{\text{fa}} sin\delta_w}{m v_{\text{xx}}} \\ v_{\text{xx}} sin(\beta_1) \end{pmatrix} \tag{5.2}$$

As this algorithm contains the parameters for the ego-vehicle model only, so the road geometry parameters are disabled and assigned zero value. The measurements and measurement equations for ego-vehicle are provided in equations 2.19 and 2.22, respectively.

The results of algorithm 2 are compared with the reference signal and the results are shown in Figure 5.9 for the first scenario. From Figure 5.9(a) we can see that the simulated signal for $\dot{\psi}_1$ has not been estimated completely as compared to the reference signal. It can also observed from Figure 5.9(b) that estimate of $\beta_1$ does not fully follow the reference signal. Figures 5.9(c) and (d) represent the $l_1$ and zoomed $l_1$, respectively. It can be observed that $l_1$ is estimated too small it is nearly equal to zero due to the reason that the parameter estimation in algorithm 2 depends only on the information of ego-vehicle motion without any information of road geometry. From these results, we can see that the simulated signal does not follow the reference signal and not estimating the true parameters for $\dot{\psi}_1$, $\beta_1$, and $l_1$ as



(a)

(b)

(c)

(d)

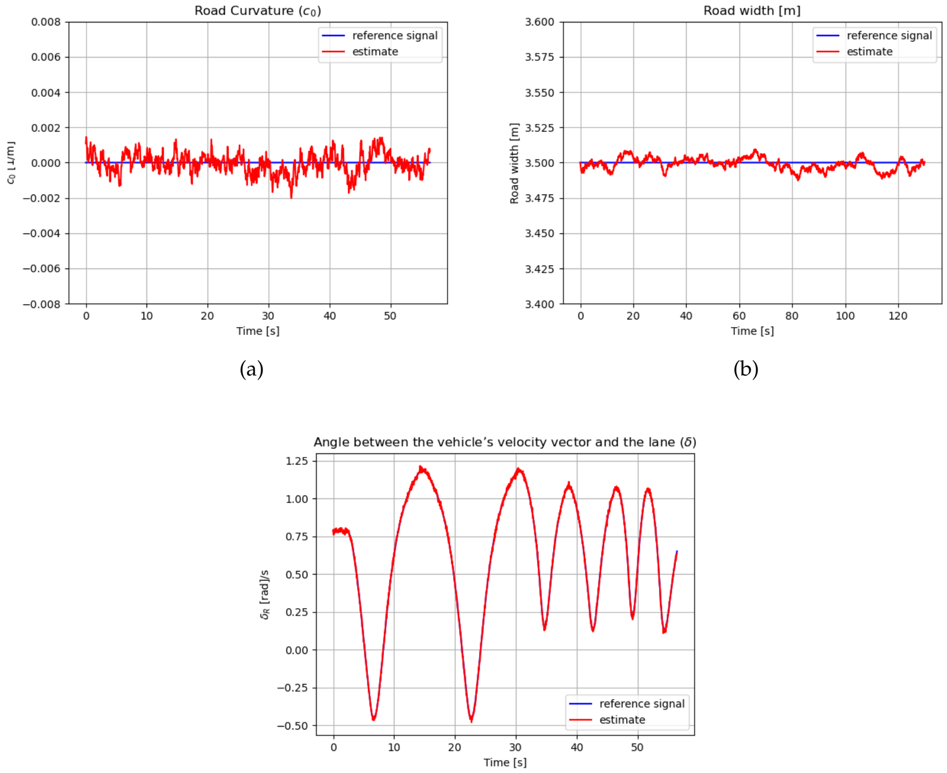
Figure 5.10: (a) Yaw Rate. (b) Float angle.(c) Distance between ego-vehicle and left lane. (d) Zoomed $\boldsymbol{l_1}$

compare to algorithm 1 because in algorithm 1, we have an additional source of information of road parameters which strengthens the estimation of the states.

The results of algorithm 2 are also compared with the reference signal for the second scenario and the results are shown in Figure 5.10. From Figure 5.10(a), we can see that the simulated signal for $\dot{\psi}_1$ does not perfectly follow the reference signal. It can also be noticed from Figure 5.10(b) that the estimate of $\beta_1$ does not perfectly follow the reference signal. Figures 5.10(c) and (d) represent the $l_1$ and focused $l_1$ signal, respectively. From the output of $l_1$, we can observe that there is a huge difference between the estimated and reference signal. From these results, it can be noticed that $\dot{\psi}_1$, $\beta_1$, and $l_1$ are not estimated properly as compare to algorithm 1 which also contains the information of the road geometry.

**Algorithm 3:** The algorithm 3 represents the independent parameter estimation for the road model. It contains road curvature $c$, angle between the lane and velocity vector of the vehicle $\delta$ and road width $w_2$ parameters. The state vector for the road model is described in equation 2.6. The change of road curvature to time and the non-linear state-space model for the road geometry can be represented respectively as:

$$\begin{aligned}\dot{c} = \frac{1}{(Im^2 v_{\mathrm{xx}})^4}(C_{\mathrm{fa}}^2(I + l_{\mathrm{fa}}^2 m)(\dot{\psi}_1 l_{\mathrm{fa}} + (-\delta_w)v_{\mathrm{xx}}) + \dot{v_{\mathrm{xx}}} Im^2(v_{\mathrm{xx}}(-cv_{\mathrm{xx}}))\\ +C_{\mathrm{fa}}(C_{\mathrm{ra}}(I + l_{\mathrm{ra}}(-l_{\mathrm{fa}})m)(-\delta_w v_{\mathrm{xx}})\\ +I((-2\delta_w)m\dot{v_{\mathrm{xx}}} v_{\mathrm{xx}} + \delta_w v_{\mathrm{xx}}^2))\end{aligned} \tag{5.3}$$



(a)

(b)

(c)

Figure 5.11: (a) Road curvature. (b) Road width. (c) Angle between the lane and velocity vector of the vehicle.

$$g_2 = \begin{pmatrix} \dot{c} \\ cv_{\mathrm{xx}} + \frac{C_{\mathrm{fa}} \sin \delta_w}{m v_{\mathrm{xx}}} \\ \mathbf{0} \end{pmatrix} \tag{5.4}$$

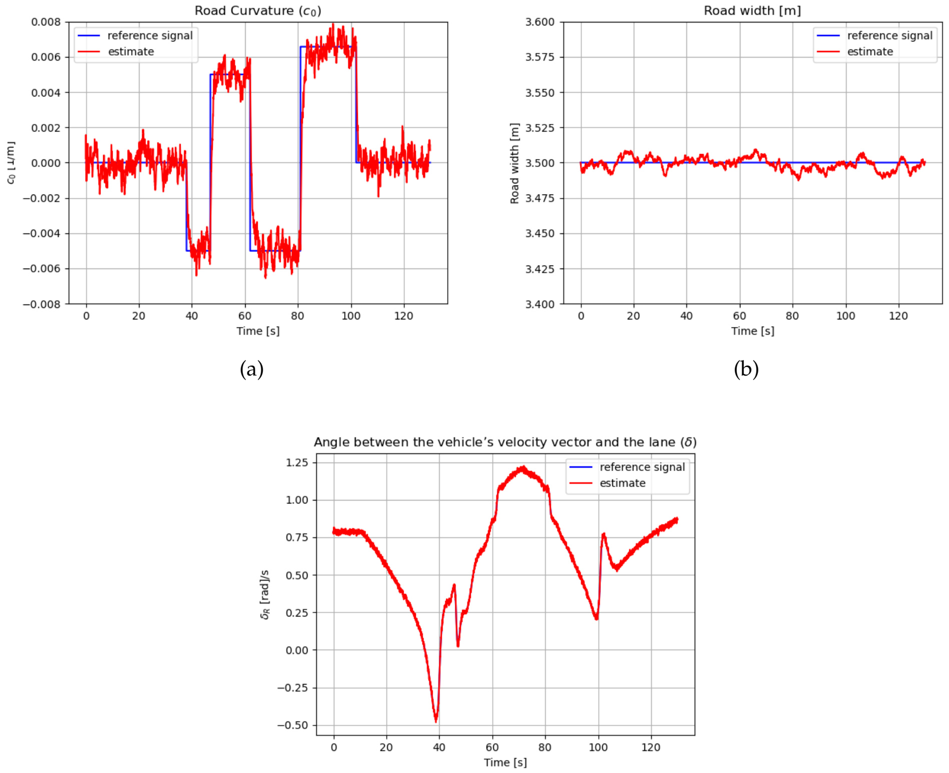
As this algorithm contains the parameters for road geometry only, so the ego-vehicle parameters are disabled and assigned zero value. The measurements for road geometry are given by equation 2.20. The measurement equation for road geometry is given by:

$$h_2 = [c \;\; (\delta) \;\; r_2 \;\; l_1]^T \tag{5.5}$$

The results of algorithm 3 are compared with the reference signal and are shown in Figures 5.11 and 5.12 for scenario $1^{st}$ and $2^{nd}$. By observing the results for both of the scenarios, it can be easily noted that if we do not have the information of ego-vehicle parameter then an irregular behavior can be attained for the simulated signal of $\boldsymbol{c}$, $\boldsymbol{\delta}$, and $\boldsymbol{r_2}$ as compare to algorithm 1 which contains the joint ego-vehicle and road geometry estimation.

### RMSE Comparison for Algorithm 1,2, and 3

The RMSE of algorithms 1, 2, and 3 are compared and the results are shown in Tables 5.2 and 5.3 for $1^{st}$ and $2^{nd}$ scenarios, respectively. In these tables, the columns represent the parameters ($\dot{\boldsymbol{\psi}}_1$, $\boldsymbol{\beta_1}$, $\boldsymbol{l_1}$, $\boldsymbol{c}$, $\boldsymbol{\delta}$, and $\boldsymbol{w_2}$) and rows represent algorithms 1,2, and 3. From the comparison between algorithms 1 and 2 in Table 5.2, we can see that the RMSE for $\dot{\boldsymbol{\psi}}_1$ and $\boldsymbol{\beta_1}$ in algorithm 2 are slightly larger as compared to algorithm 1 and the RMSE for $\boldsymbol{l_1}$ in algorithm



(a)

(b)

(c)

Figure 5.12: (a) Road curvature. (b) Road width. (c) Angle between the lane and velocity vector of the vehicle.

2 is too large as compared to our algorithm. Similarly, it can be observed from the comparison between algorithm 1 and algorithm 3 that the RMSE for $c$, and $r_2$ are comparatively larger from algorithm 1. It can also noticed that $\delta$ is twice as large in algorithm 3 as compare to algorithm 1. From these results, it is clear that our algorithm is considerably improved. In the case of scenario $2^{nd}$, by comparison between algorithms 1 and 2 as shown in Table 5.2, it can be interpreted from the output that the RMSE for $\dot{\psi}_1$ in algorithm 2 is slightly larger as compared to algorithm 1. A large amount of increase in the RMSE for $\beta_1$ and $l_1$ can also be observed in algorithm 2 as compare to our algorithm. In same way, when comparing algorithm 1 to algorithm 3, six times increase can be noticed for $c$ in algorithm 3. The RMSE for $\delta$ also shows twice increment in algorithm 3 as compare to algorithm 1, and the RMSE for $w_2$ is slightly larger from algorithm 1. From these results, we can infer that our algorithm is very improved as compared to the other two algorithms.

Table 5.2: Comparison of RMSE for scenario $1^{st}$.

| RMSE for Scenario $1^{st}$ | | | | | | |
|---|---|---|---|---|---|---|
| | $\dot{\psi}_1$ | $\beta_1$ | $l_1$ | $c$ | $\delta$ | $r_2$ |
| Algorithm 1 | 0.01064 | 0.001104 | 0.04126 | 0.0010937 | 0.008836 | 0.004761 |
| Algorithm 2 | 0.0288 | 0.0060 | 0.986 | – | – | – |
| Algorithm 3 | – | – | – | 0.001131 | 0.0101 | 0.00595 |

The comparison for of RMSE for scenario $2^{nd}$ is presented in Table 5.3.

Table 5.3: Comparison of RMSE for scenario $2^{nd}$.

| RMSE for Scenario $2^{nd}$ | | | | | | |
|---|---|---|---|---|---|---|
| | $\dot{\psi}_1$ | $\beta_1$ | $l_1$ | $c$ | $\delta$ | $r_2$ |
| Algorithm 1 | 0.01435 | 0.000411 | 0.06569 | 0.0001597 | 0.008781 | 0.005184 |
| Algorithm 2 | 0.0670 | 0.00763 | 1.006 | – | – | – |
| Algorithm 3 | – | – | – | 0.000627 | 0.01049 | 0.00623 |

By comparing the results between algorithm 1 and algorithm 2, we can infer that the accuracy of our approach is improved for $\dot{\psi}_1$, $\beta_1$, and $l_1$ when they are estimated jointly in algorithm 1 as compare to algorithm 2 when estimated independently. Similarly, by noticing the results of algorithm 1 and algorithm 3, we can observe that the accuracy for $c$, $\delta$, and $w_2$ is more in algorithm 1 as compare to algorithm 3 when estimated independently. From these results, we can also discover the important parameters of the model. It can be noticed that by excluding the parameter of road geometry in algorithm 2, the filter performance gets worst. Similarly, in algorithm 3, when we do not include the information of ego-vehicle parameters, then the expected output from the filter is not achieved. It can also be noticed that by estimating the model parameters jointly we get better results as compared to estimating them individually. From these outputs, it can be inferred that when we have an algorithm that estimates the parameters of a ego-vehicle and road geometry alone and we have another algorithm that uses jointly information and estimates the same parameters more accurately because it uses an additional source of information which enhances the performance.

The role of the jointly algorithm is to take account of all the dependencies between different parameters e.g., if the car moves to the right then yaw rate, road curvature, distance between ego-vehicle and left lane, and angle between vehicle velocity vector and lane also changes which means that there are a bunch of parameters that are highly connected and we can take advantage of that. So, joint estimation improves the results and we should utilize it to get better estimates of the parameters.

# 6 Conclusion

This thesis aims to evaluate a sensor fusion algorithm that can be utilized for combined estimation of ego-vehicle and road geometry for ADAS. The sensor fusion algorithm combines the information from inertial sensors, wheel speed sensors, steering wheel angle sensor, and camera for estimating the road geometry and ego-vehicle motion. We have utilized UKF for the state estimation of the non-linear system because UKF estimates the state in a simplified way without using the complex computations. Two scenarios were considered for the experiments. The data for $1^{st}$ scenario contains 130 seconds information from a winding road, and $2^{nd}$ scenario involves a 56 seconds drive on a straight road.

For evaluation, each estimated parameter was compared with the reference signal and it can be interpreted from these results that the simulated estimates match the measurements from the car sensor well in these two scenarios which means that the car's behavior in the simulation matches the actual movement of the car in even the transient conditions. We also examined the performance of the tracker by calculating RMSE between the estimated and true signal and performed tests to verify the working of the tracker. By observing the results, we can say that we have more confidence in functioning of our tracker. We also compared our algorithm with the baseline algorithms which separately estimate the parameters ego-vehicle and road geometry and by examining the results we can interpret that when the parameters of ego-vehicle and road geometry are estimated jointly then better accuracy is achieved instead of estimating them alone/independently.

In a nutshell, we can infer from the results of the research thesis that the UKF is able to estimate the states of the system very well in an easier way. The parameters of ego-vehicle motion and road geometry are highly connected with each other which provides better performance and functioning of our joint algorithm.

# 7 Future Work

The proposed algorithm in this thesis provides a platform for jointly estimating the ego-vehicle motion and road geometry estimation but still, other possibilities can be added to the existing platform. Some of the options are as follows:

1. An external radar sensor measurement data could be added to the existing algorithm. The radar will provide the measurements of leading vehicles and by using the information of leading vehicles, we can get a better estimation of road geometry.

2. Through radar, it would also be possible to detect and track road barriers such as tunnels and guard rails for enhancing the performance of vehicle.

3. Ego-vehicle can detect the surrounding obstacles through camera and radar sensors to avoid collision and select the suitable route for the ego-vehicle.

4. The performance of the algorithm could be examined under various scenarios considering different weather and road conditions.

# Bibliography

[1] Nabil Benmansour, Raphaël Labayrade, Didier Aubert, and Sébastien Glaser. "Stereovision-based 3D lane detection system: a model driven approach". In: *2008 11th International IEEE Conference on Intelligent Transportation Systems*. IEEE. 2008, pp. 182–188.

[2] Nabil Benmansour, Raphaël Labayrade, Didier Aubert, Sébastien Glaser, and Dominique Gruyer. "A model driven 3D lane detection system using stereovision". In: *2008 10th International Conference on Control, Automation, Robotics and Vision*. IEEE. 2008, pp. 1277–1282.

[3] Keshav Bimbraw. "Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology". In: *2015 12th international conference on informatics in control, automation and robotics (ICINCO)*. Vol. 1. IEEE. 2015, pp. 191–198.

[4] X Cheng and Vikass Monebhurrun. "Uncertainty evaluation in numerical modeling of complex devices". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 67. 1. IOP Publishing. 2014, p. 012019.

[5] Andreas Eidehall, Jochen Pohl, and Fredrik Gustafsson. "Joint road geometry estimation and vehicle tracking". In: *Control Engineering Practice* 15.12 (2007), pp. 1484–1494.

[6] Chengling Fang, Jiang Liu, Songqing Ye, and Ju Zhang. "The Geometric Unscented Kalman Filter". In: *arXiv preprint arXiv:2009.13079* (2020).

[7] Uwe Franke and Armin Joos. "Real-time stereo vision for urban traffic scene understanding". In: *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No. 00TH8511)*. IEEE. 2000, pp. 273–278.

[8] Stefan Fuchshumer, Kurt Schlacher, and Thomas Rittenschober. "Nonlinear vehicle dynamics control-a flatness based approach". In: *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE. 2005, pp. 6492–6497.

[9] Ángel F Garcıa-Fernández, Lars Hammarstrand, Maryam Fatemi, and Lennart Svensson. "Bayesian road estimation using onboard sensors". In: *IEEE Transactions on Intelligent Transportation Systems* 15.4 (2014), pp. 1676–1689.

[10] Fredrik Gustafsson. *Statistical sensor fusion*. Studentlitteratur, 2010.

[11] Fredrik Gustafsson and Gustaf Hendeby. "Some relations between extended and unscented Kalman filters". In: *IEEE Transactions on Signal Processing* 60.2 (2011), pp. 545–555.

[12] Lars Hammarstrand, Maryam Fatemi, Ángel F Garcıa-Fernández, and Lennart Svensson. "Long-range road geometry estimation using moving vehicles and roadside observations". In: *IEEE Transactions on Intelligent Transportation Systems* 17.8 (2016), pp. 2144–2158.

[13] Rambabu Kandepu, Bjarne Foss, and Lars Imsland. "Applying the unscented Kalman filter for nonlinear state estimation". In: *Journal of process control* 18.7-8 (2008), pp. 753–768.

[14] Bernd Kitt, Andreas Geiger, and Henning Lategahn. "Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme". In: *2010 ieee intelligent vehicles symposium*. IEEE. 2010, pp. 486–492.

[15] Jelena Kocić, Nenad Jovičić, and Vujo Drndarević. "Sensors and sensor fusion in autonomous vehicles". In: *2018 26th Telecommunications Forum (TELFOR)*. IEEE. 2018, pp. 420–425.

[16] Christian Lundquist. "Sensor fusion for automotive applications". PhD thesis. Linköping University Electronic Press, 2011.

[17] Christian Lundquist and Thomas B Schon. "Road geometry estimation and vehicle tracking using a single track model". In: *2008 IEEE Intelligent Vehicles Symposium*. IEEE. 2008, pp. 144–149.

[18] Christian Lundquist and Thomas B Schön. "Joint ego-motion and road geometry estimation". In: *Information Fusion* 12.4 (2011), pp. 253–263.

[19] Christian Lundquist and Thomas B Schön. "Recursive identification of cornering stiffness parameters for an enhanced single track model". In: *IFAC Proceedings Volumes* 42.10 (2009), pp. 1726–1731.

[20] Joel C McCall and Mohan M Trivedi. "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation". In: *IEEE transactions on intelligent transportation systems* 7.1 (2006), pp. 20–37.

[21] Amr Mohamed, Jing Ren, Moustafa El-Gindy, Haoxiang Lang, and AN Ouda. "Literature survey for autonomous vehicles: sensor fusion, computer vision, system identification and fault tolerance". In: *International Journal of Automation and Control* 12.4 (2018), pp. 555–581.

[22] J Piao and Mike McDonald. "Advanced driver assistance systems from autonomous to cooperative approach". In: *Transport reviews* 28.5 (2008), pp. 659–684.

[23] Mathieu St-Pierre and Denis Gingras. "Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system". In: *IEEE Intelligent Vehicles Symposium, 2004*. IEEE. 2004, pp. 831–835.

[24] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[25] Tummala Charan Reddy, Gudivada Rupesh, and Rajani Shree. "VIRTUAL REALITY SIMULATION". In: (2019).

[26] Lukas Traub, Vadim Butakov, and Robin Simpson. "Parameter identification for a multi-body vehicle model". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 521–526.

[27] Rudolph Van Der Merwe and Eric A Wan. "The square-root unscented Kalman filter for state and parameter-estimation". In: *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*. Vol. 6. IEEE. 2001, pp. 3461–3464.

[28] EA Wan and R Van Der Merwe. "Adaptive Systems for Signal Processing, Communications, and Control Symposium". In: (2000).

[29] Mark Wielitzka, Alexander Busch, Matthias Dagen, and Tobias Ortmaier. "Unscented kalman filter for state and parameter estimation in vehicle dynamics". In: *Kalman Filters-Theory for Advanced Applications*. InTech, 2018, pp. 56–75.

[30] Koichiro Yamaguchi, Takeo Kato, and Yoshiki Ninomiya. "Vehicle ego-motion estimation and moving object detection using a monocular camera". In: *18th International Conference on Pattern Recognition (ICPR'06)*. Vol. 4. IEEE. 2006, pp. 610–613.

[31] Kim Youngjoo and Bang Hyochoong. "Introduction to Kalman filter and its applications". In: *Intechopen, London* (2018).

[32] Xudong Zhang, Dietmar Göhlich, and Chenrui Fu. "Comparative study of two dynamics-model-based estimation algorithms for distributed drive electric vehicles". In: *Applied Sciences* 7.9 (2017), p. 898.

[33] Wang Zhong and Chen Xin. "Fault detection of UAV fault based on a SFUKF". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 563. 5. IOP Publishing. 2019, p. 052099.