ESTIMATION OF VEHICLE SIGNALS FOR AUTONOMOUS DRIVING APPLICATIONS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HALİL TEMURTAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2022

Approval of the thesis:

ESTIMATION OF VEHICLE SIGNALS FOR AUTONOMOUS DRIVING APPLICATIONS

submitted by **HALİL TEMURTAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. İlkay Ulusoy
Head of Department, **Electrical and Electronics Engineering** ________________

Prof. Dr. Klaus Werner Schmidt
Supervisor, **Electical and Electronics Engineering, METU** ________________

Examining Committee Members:

Prof. Dr. Afşar Saranlı
Electrical and Electronics Engineering, METU ________________

Prof. Dr. Klaus Werner Schmidt
Electrical and Electronics Engineering, METU ________________

Assoc. Prof. Dr. Mustafa Mert Ankaralı
Electrical and Electronics Engineering, METU ________________

Assist. Prof. Dr. Ulaş Beldek
Mechatronics Engineering, Çankaya University ________________

Assist. Prof. Dr. Halit Ergezer
Mechatronics Engineering, Çankaya University ________________

Date: 14.09.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Halil Temurtaş

Signature :

ABSTRACT

ESTIMATION OF VEHICLE SIGNALS FOR AUTONOMOUS DRIVING APPLICATIONS

Temurtaş, Halil
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Klaus Werner Schmidt

September 2022, 115 pages

The aim of this study is to develop a state estimator for autonomous driving applications. The increasing demand and usage of self-driving vehicle technologies and Advanced Driver Assistance Systems (ADAS) play a vital role in vehicle safety. In order to ensure their operations, these autonomous driving applications require the observation of various vehicle signals. However, some of the necessary signals might not be directly observed by the sensors or the sensors to observe the necessary signal might be too expensive or big for mass production. Therefore, estimating necessary vehicle signals from inexpensive sensors for autonomous driving applications is a very demanding topic in the industry. This thesis aims to observe vehicle navigation solutions, yaw rate, tire slips, and tire forces using the IMU, GPS receiver, wheel encoders, and steering angle sensor found in many modern vehicles. In order to accomplish this, an Extended Kalman Filter (EKF) architecture is proposed to estimate the vehicle position, velocity, and attitude. Then a method to observe the necessary tire states using the vehicle navigation solution is proposed. A simulation environment using Gazebo Classics Simulator and ROS-2 middle-ware is built for the study to simulate vehicle dynamics and sensor data. Also, the proposed algorithms are implemented as ROS-2 nodes for real-time experiments. The experiments have been performed using a vehicle model in this environment. The results showed that the proposed algorithm could produce position estimates within a 5-meter error, velocity estimates within the 0.5 cm/s error, and attitude estimates within the 0.5-degree error. Moreover, the proposed algorithm can observe lateral and longitudinal tire force with up to 200-N error if there is sufficient change in heading to improve the quality of attitude estimates. Results showed that the proposed architecture could estimate sufficient vehicle navigation solutions and tire states.

Keywords: autonomous vehicle, autonomous driving, estimation, sensor fusion, navigation, kalman filter, slip, contact force

ÖZ

OTONOM SÜRÜŞ UYGULAMALARI İÇİN ARAÇ SİNYALLERİNİN TAHMİNİ

Temurtaş, Halil
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Klaus Werner Schmidt

Eylül 2022 , 115 sayfa

Bu çalışmanın amacı otonom sürüş uygulamaları için durum kestirici geliştirmektir. Otonom sürüş teknolojileri ve Yardımcı Sürüş Destek Sistemleri (YSDS) için artan talep ve kullanımdaki artış, araç güvenliği üzerinde önemli bir yer oynamaktadır. Bu otonom sürüş uygulamaları, işleyişleri sağlamak için çeşitli araç sinyallerinin gözlemlenmesine ihtiyaç duymaktadır. Ancak gerekli sinyallerin bir kısmı sensörler tarafından doğrudan gözlemlenebilir olmayabilir veya gerekli sinyali gözlemlemek için kullanılan sensörler seri üretim için çok pahalı veya büyük olabilir. Bu nedenle, otonom sürüş uygulamaları için pahalı olmayan sensörlerden gerekli araç sinyallerinin tahmini endüstride çok talep edilen bir konudur. Bu tez, bir çok modern araçta bulunabilen IMU, GPS alıcısı, tekerlek kodlayıcıları ve direksiyon açısı sensörlerini kullanarak araç navigasyon çözümünü, savrulma oranını, lastik kaymalarını ve lastik kuvvetlerini gözlemlemeyi amaçlamaktadır. Bunu başarmak için; araç konum, hız ve yönelimlerini kestiren bir Genişletilmiş Kalman Filtresi (GKF) mimarisi önerilmiştir. Daha sonra, gerekli teker durumlarının araç seyrüsefer çözümü kullanılarak kestirilmesi için bir method önerilmiştir. Bu çalışma kapsamında, sensör verileri ve araç dinamiğinin simüle edilmesi amacıyla Gazebo Classics Simulator ve ROS-2 aracı yazılımı kullanılan bir simulasyon ortamı kurulmuştur. Ek olarak, önerilen algoritmalar ROS-2 düğümleri olarak gerçekleştirilmiştir. Deneyler, tasarlanan simülasyon ortamında bir araç modeli kullanılarak gerçekleştirilmiştir. Sonuçlar, önerilen algoritmanın 5 metrelik bir hata içinde konum kestirimleri, 0,5 cm/sn hata içinde hız kestirimleri ve 0,5 derecelik hata içinde yönelim kestitimleri üretebildiğini göstermiştir. Ayrıca, önerilen algoritma, yönelim tahminlerinin kalitesini iyileştirmek için baş açısında yeterli değişiklik göstermesi koşulu ile, 200-N'ye kadar hata ile yanal ve boyuna lastik kuvvetini gözlemleyebilir. Sonuçlar, tavsiye edilen algoritmaların yeterli araç seyrüsefer çözümü ve teker durumları kestirebildiğini göstermiştir.

Anahtar Kelimeler: otonom araç, otonom sürüş, kestirim, sensör füzyonu, seyrüsefer, kalman filtre, kayma, temas kuvveti

To my family

ACKNOWLEDGMENTS

Firstly, I would like to express my deepest gratitude to my supervisor Prof. Dr. Klaus Werner Schmidt, for his patience, guidance, kind support, and valuable comments. Throughout the thesis, his encouragement and positive attitude helped me continue my studies.

I would like to thank TÜBİTAK-SAGE and all my friends and colleagues. Especially to Kerem Eyice, Görkem Kandemir, Fatih Karagöz, Muzaffer Temelli, Murat Namıduru, Doğukan Kök, Barış Özdemir and Onur Gülsem who all contributed this study with their endless encouragement.

I would also like to thank to Erdem Tuna, Sarper Sertel, Mustafa Yenievli, Akif Kurt, Bedirhan Tavşan, Haluk Levent Çiçek, Ramazan Dürek and Hakan Özen for their intimate frendship and the times we spent together.

I would like to thank my mother, my father, and my sister, Tuğba, for their unconditional love and support throughout my entire life. This study would not be possible without their patience and faith in me.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ADAS	Advanced Driver Assistance Systems
ABS	Anti-lock Braking System
ESP	Electronic Stability Program
ACC	Adaptive Cruise Control
LDW	Lane Departure Warning
LKA	Lane Keeping Assistance
LCA	Lane Changing Assistance
INS	Inertial Navigation Solution
IMU	Inertial Measurement Unit
GPS	Global Positioning System
GNSS	Global Navigation Satellite Systems
NED	North East Down
WCWF	wheel-centered wheel-fixed
DCM	Direction cosine matrix
LQE	Linear Quadratic Estimator
LKF	Linear Kalman Filter
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
ROS	Robot Operating System
DDS	Data Distribution Service
QoS	Quality of Settings
URDF	Unified Robot Description Format
SDF	Simulation Description Format
FL	Front Left

FR	Front Right
RL	Rear Left
RR	Front Right
SCN	Scenario

LIST OF SYMBOLS

$r^{\gamma}_{\alpha\beta}$	Position vector of the origin of Frame α with respect to Frame β resolved at Frame γ
$v^{\gamma}_{\alpha\beta}$	Velocity vector of the Frame α with respect to Frame β resolved at Frame γ
$a^{\gamma}_{\alpha\beta}$	Acceleration vector of the Frame α with respect to Frame β resolved at Frame γ
$w^{\gamma}_{\alpha\beta}$	Angular velocity vector of Frame α with respect to Frame β resolved at Frame γ
C^{γ}_{α}	Direction Cosine Matrix (DCM) from Frame α to Frame β
$\dot{C}^{\gamma}_{\alpha}$	Time derivative of Direction Cosine Matrix C^{γ}_{α}
$v^{\gamma}_{\alpha\beta,x}$	X axis component of the velocity vector $v^{\gamma}_{\alpha\beta}$
$C_{\sigma_{t_k}}$	The longitudinal tire stiffness coefficient for k^{th} tire
$C_{\alpha_{t_k}}$	The lateral tire stiffness coefficient for k^{th} tire
$F^{t_k}_{z}$	The vertical force on the k^{th} tire
μ	The tire-road friction coefficient
R_{eff}	Effective tire radius
$v^{t_k}_{x}$	Longitudinal velocity of the k^{th} tire
w^{t_k}	Rotational Velocity of the k^{th} tire
$\Omega^{\gamma}_{\alpha\beta}$	The skew symmetric form of vector $w^{\gamma}_{\alpha\beta}$

CHAPTER 1

INTRODUCTION

1.1 Motivation

Even though the safety of vehicles in 2022 is far more advanced than the ones a decade ago, thousands of people still lose their lives in traffic accidents [6]. Moreover, according to research published by the US National Highway Safety Administration, the majority of road accidents are caused by human errors [7]. To increase road safety, researchers around the globe continue their work on driving assistance systems. The article [8] concluded that widely deployed Advanced Driver Assistance Systems (ADAS) such as ABS, ACC, and LDW have the potential to reduce crash rate by 47 percent.

These driving assistance systems require some vehicle states to be observed. For instance, the controller for the Anti-lock Braking System (ABS) provides a controlled torque to maintain the optimum value of the wheel longitudinal slip ratio [9]. By doing so, ABS allows the driver to stop the vehicle at shorter distances with controllable steering. Another important ADAS, Electronic Stability Program (ESP), can reduce the vehicle speed and lateral acceleration to prevent a rollover [10]. To achieve this, ESP controls the vehicle's yaw moment via a braking system to ensure bounded yaw rate and lateral side-slip angle. Other critical driving assistance systems such as Adaptive Cruise Control (ACC), Lane Keeping Assistance (LKA), and Lane Changing Assistance (LCA) all utilize several vehicle states, such as tire contact forces.

Moreover, with the advances of onboard computer processing capabilities and advances in the sensors, autonomous driving capabilities of the vehicles increases rapidly. Vehicles with self-driving capabilities can drive to destination without any human interaction. However, to achieve this, self-driving vehicle not only able to recognize their surroundings, but also their exact location in Earth. To create more robust navigation solution, autonomous vehicles these days equipped with more sensors such as GPS receiver and LIDAR. However, not every signal is easily observable with a sensor. Or necessary sensors to observe some vehicle states might be too expensive of big to use in the vehicles. In order to overcome these problems, researchers in the automotive industry work on estimation algorithms to observe necessary vehicle signals from inexpensive cost sensors.

In this thesis, a method to observe following vehicle signals from onboard sensors will be proposed:

- Vehicle navigation solution (position, velocity, attitude)
- Vehicle yaw rate
- Tire longitudinal slip ratio
- Tire lateral side-slip angle
- Tire longitudinal contact force
- Tire lateral contact force

1.2 Literature Survey

The research in vehicle state estimation can be divided into four parts: estimation of vehicle navigation solution, estimation of vehicle longitudinal dynamics, estimation of lateral dynamics, and estimation of friction models parameters.

[11] proposes a method to estimate the longitudinal velocity using accelerometers and encoders. However, the algorithm relies on determining the wheel slip to avoid using encoder readings. Similarly, [12] proposes a more robust solution to estimate only longitudinal linear velocity from IMU and encoder readings to estimate longitudinal tire dynamics. [13] uses only GPS velocity solution to estimate longitudinal slip ratio. Furthermore, [14] combines INS and GPS solutions to estimate a more accurate longitudinal slip ratio.

[15] uses an EKF based on a 5-DOF nonlinear vehicle dynamics model to estimate the side-slip angle. However, the results of this study align with the nonlinear model used to construct the filter. The results diverge from the ideal 10-DOF vehicle model in higher slip conditions such as heavy breaking. [16] proposes a similar state estimator based on a 5-DOF nonlinear model; however, the algorithms of this study were only fed with a high-quality IMU. The author suggests future research with low-cost IMU. In [17], [18], [19] conducts experiments with real vehicle to evaluate the proposed work. The proposed methods in [17], [18], [19] suggest the usage of EKF to observe not only lateral forces but also normal forces to increase the quality of lateral force estimation. The proposed filter includes heading, lateral and longitudinal velocity, and lateral tire forces in the state vector [20] proposes an estimator based on the Interacting Multiple Model algorithm to observe side-slip angle in various conditions. [21] developed a method for estimating side-slip angle using inexpensive sensors by using a kinematics observer. The proposed algorithm in [21] can also be used to observe road bank angles and tire cornering stiffness. [22] estimates the side-slip angle as a weighted mean of two different estimation algorithms. The first proposed method uses a kinematic formulation, and the second one uses a state observer. [23] proposes some identification methods using sensor measurements such as steering angle, lateral acceleration, and the yaw rate. Although the proposed methodology of [23] does not include heavy filtering and mostly relies on simple algebraic relationships, the results present promising results.

[24] proposes a vehicle model to estimate the lateral and vertical tire forces using a Kalman filter. The model in [24] utilizes the roll, pitch dynamics of the vehicle, and gravity components of the accelerometer measurement. However, the force estimates are only valid for the cases where the vehicle parameters are constant due to assumptions. [25], [26] also propose an EKF structure to observe critical vehicle states such as side-slip angle and lateral and longitudinal tire forces. [27] utilizes an UKF-based estimation for estimation tire longitudinal slip, friction coefficient. [27], as a result, states that potential usage of roll, pitch rate, and GPS velocity could increase the estimation quality. Similarly, [28] proposes a UKF-based estimator to realize the vehicle states. [29] combines INS solution with GPS solution and wheel encoder measurements to create a more accurate vehicle navigation solution.

[30] tries to estimate the road friction parameters from the IMU measurements. [31] discussed different methods to observe tire cornering stiffness from vehicle lateral dynamic equations. Furthermore, [32] proposes a method to estimate tire cornering and normal forces from different vehicle measurements such as acceleration, pitch, yaw rate, and suspension displacements. [33], [34], and [35] propose different methods to estimate the road slope using an accelerometer, steering angle sensor, wheel encoder, and yaw rate measurements.

1.3 Scope of the Thesis

The thesis proposes a filter architecture to observe vehicle navigation solution. Furthermore, it proposes an architecture to observe necessary tire states from the estimated navigation solution.

In order to estimate the position, velocity, and attitude of the vehicle, an Error-state Extended Kalman Filter-based filter is proposed. The proposed filter architecture combines the inertial navigation solution (INS) produced from the inertial measurement unit (IMU) measurements and GPS receiver measurements.

Moreover, a method is proposed to observe longitudinal slip ratio and side-slip angle from the knowledge of vehicle navigation solution, steering angle sensor measurement, and wheel encoder measurements. Then, a method is proposed to observe longitudinal and lateral tire contact forces. The contribution of the study is to propose an algorithm that can produce a high-quality navigation solution and observe both lateral and longitudinal dynamics of the tire, unlike most of the work discussed in *Section 1.2*. Moreover, the majority of the studies in the literature concentrate on the lateral movement of the vehicle; therefore, any irregularity of the road, such as bank angle, and road slope, compromises the estimation quality. Thanks to the 6-DOF navigation solution estimation, any road irregularities can be compensated through coordinate transformations.

Lastly, a simulation environment that governs the dynamics of the vehicle and sensors is utilized to validate the proposed algorithms.

1.4 Outline of the Thesis

This thesis consists of five chapters. It begins with an introductory chapter that explains the motivation for the study and literature survey on the topic. The rest of the thesis is organized as follows:

Chapter 2 presents necessary background knowledge for constructing a loosely coupled INS/GPS filter and tire contact force calculation. This chapter firstly focuses on the kinematic identities of a vehicle. Then, a detailed introduction to tire dynamics is presented. Finally, the underlying mechanisms of Kalman filtering are discussed in **Chapter 2**.

A methodology for the proposed INS/GPS filter architecture is provided in the first part of **Chapter 3**. Then, this chapter continues with the methods to estimate the tire slip variables and tire contact forces from the knowledge of navigation solutions.

In **Chapter 4**, performance of the proposed algorithms are investigated. This chapter begins with an explanation of the simulation environment built for the thesis. Then, the results of the Monte Carlo analysis in the simulation environment are presented and discussed.

Finally, in **Chapter 5**, concluding remarks of the thesis are presented.

CHAPTER 2

THEORETICAL BACKGROUND

2.1 Introduction

In this study, an Extended Kalman Filter is used to estimate some of the vehicle states necessary for the autonomous driving applications. This chapter provides the theoretical background necessary to follow the problem of sensor fusion for autonomous vehicle applications. The chapter begins by explaining the basics of kinematics, such as coordinate frames, attitude representation, and corresponding frame transformation methods. Then, the necessary background for the tire dynamics are introduced. Finally, an introduction to Kalman filtering is presented.

2.2 Kinematics

Kinematics is a subfield of classical physics developed to describe the motion of a robotic mechanism or body without considering the forces and torques that cause the movement. The problem of effectively representing the position, orientation, and their derivatives with respect to time has been the focus of the robotics community for many years [36]. In this section, necessary background information about kinematics is given.

2.2.1 Coordinate Frames

Coordinates frames are essential parts of any problem that requires the kinematic analysis of the movement, such as the modeling, simulation, and navigation. In this section, the coordinate frames used in this thesis to analyze the motion of an autonomous vehicle are defined.

- **Inertial Cartesian Frame (c-frame)**

  The inertial cartesian frame is the reference frame used in this study which has a fixed origin with respect to the simulated world and not rotating relative to the world. The x, y, and z axes are aligned with the topographic directions, and point to the north, east, and down. The study aims to estimate the navigation solution of the vehicle and tires with respect to the inertial cartesian frame.

- **Local Navigation Frame (NED, n-frame)**

  The Local navigation frame is a frame used in this study that is rigidly attached to the simulated vehicle. The local navigation frame axes are parallel to the inertial cartesian frame axes and aligned with the topographic directions. The x, y, and z axes point to the north, east, and down directions. The aim is to represent the estimated navigation solution of the vehicle and tires in the local navigation frame. Due to its axes alignment, this kind of frames can sometimes be referred to as the NED frame. [37]

- **Body Frame (b-frame)**

  The body frame is a frame used in this study that is rigidly attached to the simulated vehicle. And its origin coincides with the center of the IMU. The body frame axes are aligned with the vehicle, and the x, y, and z axes point to the nose, right, and bottom of the simulated vehicle. The body frame, local navigation frame, and inertial reference frame with respect to each other are illustrated in *Figure 2.1*.

- **Wheel Centered Wheel Fixed Frame (w-frame)**

  The wheel-centered wheel-fixed (WCWF) frame is a frame used in this study that is rigidly attached to the wheel, and its origin coincides with the center of the wheel. WCWF axes are aligned with the wheel, and x, y, and z axes point to the nose, right, and bottom of the simulated wheel. Since there are four wheels attached to the vehicle, four different WCWF frames are defined in the scope of this study.

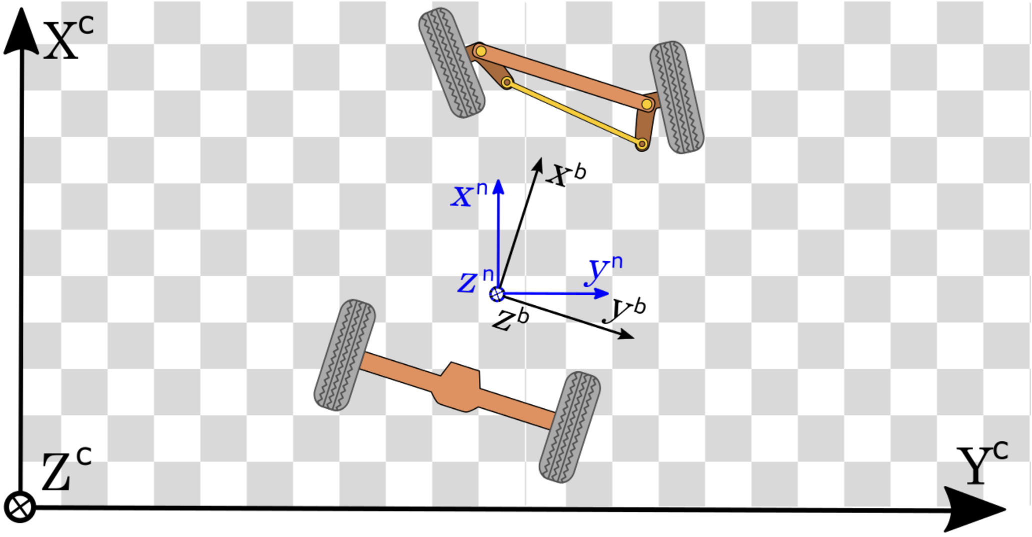

Figure 2.1: Vehicle Body Frame

- **Tire Frame (*t*-frame)**

  The tire frame is a special kind of wander-azimuth frame whose origin coincides with the center of the wheel. This frame can be constructed by rotating the body frame around its z axis with an azimuth angle and transporting it to the wheel center. In this study, the azimuth angle at each simulation step is the steering angle of the wheel. Since there are four wheels attached to the vehicle, four different tire frames are defined in the scope of this study. The WCWF frame and tire frame with respect to each other are illustrated in *Figure 2.2*.

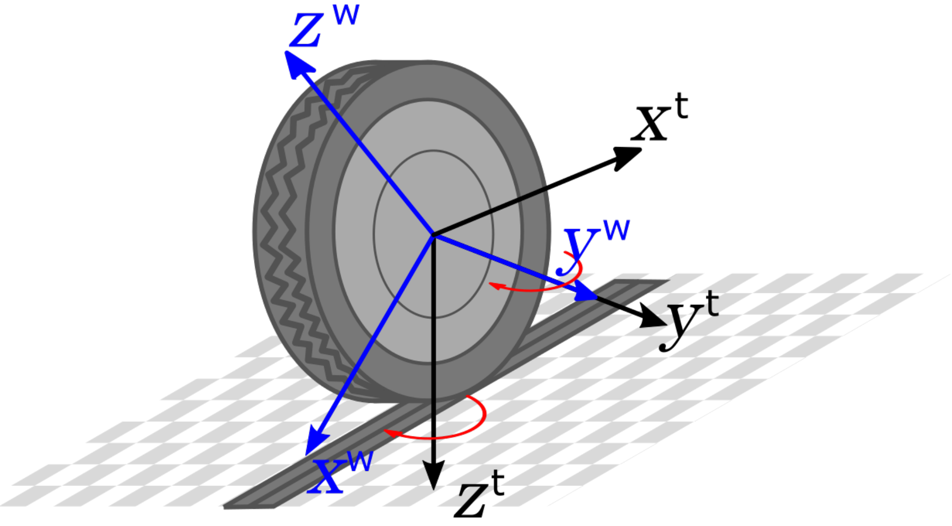

Figure 2.2: WCWF and Tire Frames

2.2.2 Attitude and Coordinate Transformations

The orientation of a coordinate frame's axes with respect to another coordinate frame's axes can be described by three coordinate transformation methods. A vector identity resolved in one coordinate frame can be expressed in another frame using these transformations. In this section, three methods used for coordinate transformations are presented.

- **Euler Angles:**

  The Euler angles are used to describe the orientation between frame α and frame β by three discrete rotations. The first rotation of $\psi_{\beta\alpha}$ degree around the z- axis of frame α is called yaw rotation, and the second rotation of $\theta_{\beta\alpha}$ degree around the y-axis of the rotated frame is called pitch rotation and the final rotation of $\phi_{\beta\alpha}$ degree around x-axis of the rotated frame is called roll rotation. The roll ($\phi_{\beta\alpha}$), pitch ($\theta_{\beta\alpha}$), and yaw ($\psi_{\beta\alpha}$) angles are used in the successive rotations are called Euler Angles that describe the orientation of the frame-α with respect to frame-β. The Euler rotation from frame-β to frame-α may be denoted by a vector:

  $$\boldsymbol{\psi}_{\beta\alpha} = \begin{bmatrix} \phi_{\beta\alpha} & \theta_{\beta\alpha} & \psi_{\beta\alpha} \end{bmatrix}^T$$

- **Direction Cosine Matrices:**

  Direction cosine matrix (DCM) or coordinate transformation matrix is a 3 × 3 matrix used to describe the orientation between two coordinate frames. A vector (v^α) that is resolved in frame-α may be resolved in frame-β using the coordinate transformation matrix (C_α^β) as in *Equation 2.1* where the subscript of the C_α^β represents the "from" coordinate frame and superscript of the matrix represents "to" coordinate frame of the transformation. Relation between Euler angles and DCM is given in *Equations 2.2 & 2.3* .

  $$v^\beta = C_\alpha^\beta v^\alpha \tag{2.1}$$

$$C_{\alpha}^{\beta} = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix} \tag{2.2}$$

$$\begin{pmatrix} \phi_{\beta\alpha} \\ \theta_{\beta\alpha} \\ \psi_{\beta\alpha} \end{pmatrix} = \begin{pmatrix} \operatorname{arctan2}\left(C^{\beta}_{\alpha3,2}, C^{\beta}_{\alpha3,3}\right) \\ -\arcsin\left(C^{\beta}_{\alpha1,3}\right) \\ \operatorname{arctan2}\left(C^{\beta}_{\alpha2,1}, C^{\beta}_{\alpha1,1}\right) \end{pmatrix} \tag{2.3}$$

- **Quaternions:**

A coordinate transformation can also be represented using a quaternion, a hyper-complex number with four components:

$$\mathbf{q} = [q_0 \;\; q_1 \;\; q_2 \;\; q_3]^T$$

The first element q_0 is the scalar part of a quaternion and the function of the magnitude of the rotation. In contrast, other elements are the vector part of the quaternion and the function of both magnitude and axis of the rotation. As with other transformation methods, only three quaternion components are independent [37].

$$\mathbf{q}_{\beta}^{\alpha} = \begin{pmatrix} \cos(\mu_{\beta\alpha}/2) \\ e_{\beta\alpha,1}\sin(\mu_{\beta\alpha}/2) \\ e_{\beta\alpha,2}\sin(\mu_{\beta\alpha}/2) \\ e_{\beta\alpha,3}\sin(\mu_{\beta\alpha}/2) \end{pmatrix} \tag{2.4}$$

A quaternion can be defined as in *Equation 2.4* where $\mu_{\beta\alpha}$ is the magnitude of the rotation, and $\mathbf{e}_{\beta\alpha}$ is the axis of rotation and is a unit vector. The relation between Euler angles and quaternions is given in *Equations 2.5 & 2.6* while the relation between transformation matrices and quaternions is given in *Equations 2.7 & 2.8* .

$$\boldsymbol{\psi}_{\beta\alpha} = \begin{pmatrix} \operatorname{arctan2}\left[2(q^{\alpha}_{\beta0}q^{\alpha}_{\beta1} + q^{\alpha}_{\beta2}q^{\alpha}_{\beta3}), 1 - 2{q^{\alpha}_{\beta1}}^2 - 2{q^{\alpha}_{\beta2}}^2\right] \\ \arcsin\left[2(q^{\alpha}_{\beta0}q^{\alpha}_{\beta2} - q^{\alpha}_{\beta1}q^{\alpha}_{\beta3})\right] \\ \operatorname{arctan2}\left[2(q^{\alpha}_{\beta0}q^{\alpha}_{\beta3} + q^{\alpha}_{\beta1}q^{\alpha}_{\beta2}), 1 - 2{q^{\alpha}_{\beta2}}^2 - 2{q^{\alpha}_{\beta3}}^2\right] \end{pmatrix} \tag{2.5}$$

$$\mathbf{q}_{\beta}^{\alpha} = \begin{pmatrix} \cos\left(\frac{\phi_{\beta\alpha}}{2}\right)\cos\left(\frac{\theta_{\beta\alpha}}{2}\right)\cos\left(\frac{\psi_{\beta\alpha}}{2}\right) + \sin\left(\frac{\phi_{\beta\alpha}}{2}\right)\sin\left(\frac{\theta_{\beta\alpha}}{2}\right)\sin\left(\frac{\psi_{\beta\alpha}}{2}\right) \\ \sin\left(\frac{\phi_{\beta\alpha}}{2}\right)\cos\left(\frac{\theta_{\beta\alpha}}{2}\right)\cos\left(\frac{\psi_{\beta\alpha}}{2}\right) - \cos\left(\frac{\phi_{\beta\alpha}}{2}\right)\sin\left(\frac{\theta_{\beta\alpha}}{2}\right)\sin\left(\frac{\psi_{\beta\alpha}}{2}\right) \\ \cos\left(\frac{\phi_{\beta\alpha}}{2}\right)\sin\left(\frac{\theta_{\beta\alpha}}{2}\right)\cos\left(\frac{\psi_{\beta\alpha}}{2}\right) + \sin\left(\frac{\phi_{\beta\alpha}}{2}\right)\cos\left(\frac{\theta_{\beta\alpha}}{2}\right)\sin\left(\frac{\psi_{\beta\alpha}}{2}\right) \\ \cos\left(\frac{\phi_{\beta\alpha}}{2}\right)\cos\left(\frac{\theta_{\beta\alpha}}{2}\right)\sin\left(\frac{\psi_{\beta\alpha}}{2}\right) - \sin\left(\frac{\phi_{\beta\alpha}}{2}\right)\sin\left(\frac{\theta_{\beta\alpha}}{2}\right)\cos\left(\frac{\psi_{\beta\alpha}}{2}\right) \end{pmatrix} \tag{2.6}$$

$$C_{\alpha}^{\beta} = \begin{pmatrix} \sum_{i=0}^{3} {q_{\beta i}^{\alpha}}^2 - 2\sum_{i=2}^{3} {q_{\beta i}^{\alpha}}^2 & 2(q_{\beta 1}^{\alpha} q_{\beta 2}^{\alpha} - q_{\beta 3}^{\alpha} q_{\beta 0}^{\alpha}) & 2(q_{\beta 1}^{\alpha} q_{\beta 3}^{\alpha} + q_{\beta 2}^{\alpha} q_{\beta 0}^{\alpha}) \\ 2(q_{\beta 1}^{\alpha} q_{\beta 2}^{\alpha} + q_{\beta 3}^{\alpha} q_{\beta 0}^{\alpha}) & \sum_{i=0}^{3} {q_{\beta i}^{\alpha}}^2 - 2{q_{\beta 1}^{\alpha}}^2 - 2{q_{\beta 3}^{\alpha}}^2 & 2(q_{\beta 2}^{\alpha} q_{\beta 3}^{\alpha} - q_{\beta 1}^{\alpha} q_{\beta 0}^{\alpha}) \\ 2(q_{\beta 1}^{\alpha} q_{\beta 3}^{\alpha} - q_{\beta 2}^{\alpha} q_{\beta 0}^{\alpha}) & 2(q_{\beta 2}^{\alpha} q_{\beta 3}^{\alpha} + q_{\beta 1}^{\alpha} q_{\beta 0}^{\alpha}) & \sum_{i=0}^{3} {q_{\beta i}^{\alpha}}^2 - 2\sum_{i=1}^{2} {q_{\beta i}^{\alpha}}^2 \end{pmatrix} \tag{2.7}$$

$$\mathbf{q}_{\beta}^{\alpha} = \begin{pmatrix} \frac{1}{2}\sqrt{1 + C_{\alpha 1,1}^{\beta} + C_{\alpha 2,2}^{\beta} + C_{\alpha 3,3}^{\beta}} \\ \dfrac{C_{\alpha 3,2}^{\beta} - C_{\alpha 2,3}^{\beta}}{4\ q_{\beta 0}^{\alpha}} \\ \dfrac{C_{\alpha 1,3}^{\beta} - C_{\alpha 3,1}^{\beta}}{4\ q_{\beta 0}^{\alpha}} \\ \dfrac{C_{\alpha 2,1}^{\beta} - C_{\alpha 1,2}^{\beta}}{4\ q_{\beta 0}^{\alpha}} \end{pmatrix} \tag{2.8}$$

2.2.3 Derivative with respect to Rotating Frame

Similar to attitude representation, a vector quantity such as position and velocity of a frame-β can be defined with respect to another frame-α as $v_{\alpha\beta}$ where the first letter in the subscript represents the "*reference*" frame and the second letter represents the "*origin*" frame of the vector quantity. However, a vector quantity can be described in infinitely many ways using different vector basis. To prevent confusion, a third coordinate "*resolving*" coordinate frame can be used to specify the vector basis in which the quantity is expressed. In this study, a resolving frame of a vector quantity is denoted in the superscript as:

$$r^{\gamma}_{\alpha\beta}$$

The resolving frame can be the same frame as the reference frame to simplify the computations or can rotate with respect to the reference frame. Moreover, a vector quantity resolved in a frame-α can be resolved in another frame-γ using *Equation 2.1*.

The velocity of a vehicle is equal to the derivative of its position, and its acceleration is equal to the derivative of its velocity if and only if the position, velocity, and acceleration are expressed in the same reference coordinate frame. However, for some applications like navigation, the resolving frame might rotate relative to the reference frame; thus, derivative operation with respect to the rotating frame should be defined.

$$\begin{aligned} \frac{d}{dt} r^{\gamma}_{\alpha\beta} &\neq v^{\gamma}_{\alpha\beta} \\ \frac{d}{dt} v^{\gamma}_{\alpha\beta} &\neq a^{\gamma}_{\alpha\beta} \end{aligned} \tag{2.9}$$

$$\begin{aligned} \frac{d}{dt} r^{\alpha}_{\alpha\beta} &= \dot{r}^{\alpha}_{\alpha\beta} = v^{\alpha}_{\alpha\beta} \\ \frac{d}{dt} v^{\alpha}_{\alpha\beta} &= \dot{v}^{\alpha}_{\alpha\beta} = a^{\alpha}_{\alpha\beta} \end{aligned} \tag{2.10}$$

In order to calculate the derivative of a variable whose reference and resolving frames rotate relative to each other, the derivative of a transformation matrix from frame-α to frame-γ can be defined as in *Equation 2.11*. Where the $w^{\gamma}_{\gamma\alpha}$ represents the rotation rate between frame frame-α and frame-γ.

$$\begin{aligned} \dot{C}^{\gamma}_{\alpha} &= w^{\gamma}_{\gamma\alpha} \times C^{\gamma}_{\alpha} \\ &= C^{\gamma}_{\alpha} w^{\gamma}_{\gamma\alpha} \times \end{aligned} \tag{2.11}$$

And to simplify the operation, the skew symmetric matrix version ($\Omega^{\gamma}_{\gamma\alpha}$) of a vector ($w^{\gamma}_{\gamma\alpha}$) can be used instead of the cross product (*Equation 2.12*). Moreover, the skew symmetric matrices can be transported to another coordinate frame using DCM as in *Equation 2.13* [37].

$$\Omega_{\gamma\alpha}^{\gamma} = w_{\gamma\alpha}^{\gamma}\times \tag{2.12}$$

$$\Omega_{\beta\alpha}^{\sigma} = C_{\gamma}^{\sigma}\Omega_{\beta\alpha}^{\gamma}C_{\sigma}^{\gamma} \tag{2.13}$$

The position of the frame-β can be resolved at its reference frame using *Equation 2.14*. Furthermore, the relation between the derivative of the position and velocity of a frame-α whose reference and resolving frames rotate relative to each other can be examined in *Equation 2.15*. Similarly, the relation of the derivative of the velocity and acceleration of the same frame is given in *Equation 2.16*.

$$r_{\alpha\beta}^{\gamma} = C_{\alpha}^{\gamma}r_{\alpha\beta}^{\alpha} \tag{2.14}$$

$$\begin{aligned}\frac{d}{dt}r_{\alpha\beta}^{\gamma} &= \frac{d}{dt}(C_{\alpha}^{\gamma}r_{\alpha\beta}^{\alpha})\\ &= C_{\alpha}^{\gamma}r_{\alpha\beta}^{\dot{\alpha}} + \dot{C}_{\alpha}^{\gamma}r_{\alpha\beta}^{\alpha}\\ &= C_{\alpha}^{\gamma}v_{\alpha\beta}^{\alpha} + w_{\gamma\alpha}^{\gamma}\times r_{\alpha\beta}^{\gamma}\\ &= v_{\alpha\beta}^{\gamma} + w_{\gamma\alpha}^{\gamma}\times r_{\alpha\beta}^{\gamma}\end{aligned} \tag{2.15}$$

$$\frac{d}{dt}v_{\alpha\beta}^{\gamma} = a_{\alpha\beta}^{\gamma} + w_{\gamma\alpha}^{\gamma}\times v_{\alpha\beta}^{\gamma} \tag{2.16}$$

Additionally derivative of the Euler angles can be expressed in terms of rotation rate and Euler angles as in *Equation 2.17* [38].

$$\begin{pmatrix}\dot{\phi}_{\beta\alpha}\\ \dot{\theta}_{\beta\alpha}\\ \dot{\psi}_{\beta\alpha}\end{pmatrix} = \begin{pmatrix}1 & \sin\phi_{\beta\alpha}\tan\theta_{\beta\alpha} & \cos\phi_{\beta\alpha}\tan\theta_{\beta\alpha}\\ 0 & \cos\phi_{\beta\alpha} & -\sin\phi_{\beta\alpha}\\ 0 & \sin\phi_{\beta\alpha}/\cos\theta_{\beta\alpha} & \cos\phi_{\beta\alpha}/\cos\phi_{\beta\alpha}\end{pmatrix} w_{\beta\alpha}^{\alpha} \tag{2.17}$$

2.2.4 Strapdown Inertial Navigation

In the scope of this study, one of the goals is to estimate the navigation solution of a vehicle in the simulation environment. The navigation solution consists of the position, velocity, and attitude vectors that can describe the location of any kinematic object in 3-D space. The aim in this study is to estimate the position (r^n_{cb}) and velocity (v^n_{cb}) vectors of a body frame (b) with respect to inertial cartesian frame (c) represented in local navigation frame (n) and the attitude (ψ^n_{nb}) of the body frame with respect to local navigation frame. In this section, update of a navigation solution using the IMU measurements (w^b_{cb}, f^b_{cb}) and previous navigation solution is introduced.

- **Attitude Update with DCM:**

  The first method to update the Euler angles over a finite interval is to use transformation matrices. The DCM can be updated using the measurements from the IMU as in *Equation 2.18* [37]. The corresponding Euler angles can be calculated using *Equation 2.3*.

$$\begin{aligned} C^n_b(+) &= C^n_b(-)(I + \Omega^b_{cb}\tau_i) - \cancelto{0}{\Omega^n_{cn}} C^n_b(-)\tau_i \\ &= C^n_b(-)(I + \Omega^b_{cb}\tau_i) \end{aligned} \tag{2.18}$$

- **Attitude Update Quaternions:**

  Quaternions can also be used to update the Euler angles over a finite interval. The DCM can be updated using the measurements from the IMU as in *Equation 2.19* where the operation "$\circ$" is called the quaternion product [37]. The corresponding Euler angles can be calculated using *Equation 2.5*.

$$\begin{aligned} q^b_n(+) &= q^b_n(-) \circ \begin{pmatrix} 1 \\ \frac{1}{2} w^b_{cb}\tau_i \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{1}{2} \cancelto{0}{w^b_{cn}} \tau_i \end{pmatrix} \circ q^b_n(-) \\ &= q^b_n(-) \circ \begin{pmatrix} 1 \\ \frac{1}{2} w^b_{cb}\tau_i \end{pmatrix} \end{aligned} \tag{2.19}$$

- **Velocity Update:**

  In order to update the velocity solution over a finite interval, the derivative of the vehicle velocity can be calculated using *Equation 2.10* as:

$$\frac{d}{dt}v_{cb}^{n} = a_{cb}^{n} + \cancelto{0}{w_{nc}^{c}} \times v_{cb}^{n} = a_{cb}^{n} \tag{2.20}$$

The equation can be discretized over a small time interval (τ_i) as in *Equation 2.21*. By rearranging the terms, the equation becomes to *Equation 2.22* where the terms corresponds to total acceleration are the specific force measurement (f_{cb}^{b}) and gravitational acceleration (g^n). Specific force is the non-gravitational force per unit mass on a body sensed with respect to inertial frame [37]; thus, the gravitational acceleration should be compensated using the gravitational models.

$$\frac{v_{cb}^{n}(t_k) - v_{cb}^{n}(t_k+\tau_i)}{\tau_i} = \frac{a_{cb}^{n}(t_k) + a_{cb}^{n}(t_k+\tau_i)}{2} \approx a_{cb}^{n}(t_k+\tau_i) \tag{2.21}$$

$$\begin{aligned} v_{cb}^{n}(t_k+\tau_i) &= v_{cb}^{n}(t_k) + a_{cb}^{n}(t_k+\tau_i)\tau_i \\ &= v_{cb}^{n}(t_k) + (f_{cb}^{n}(t_k+\tau_i) + g^{n}(t_k+\tau_i))\tau_i \\ &= v_{cb}^{n}(t_k) + (C_b^n(t_k+\tau_i)f_{cb}^{b}(t_k+\tau_i) + g_n(t_k+\tau_i))\tau_i \end{aligned} \tag{2.22}$$

- **Position Update:**

  In order to update the velocity solution over a finite interval, the derivative of the vehicle velocity can be calculated using *Equation 2.9* as:

$$\frac{d}{dt}r_{cb}^{n} = v_{cb}^{n} + \cancelto{0}{w_{nc}^{c}} \times r_{cb}^{n} = v_{cb}^{n} \tag{2.23}$$

The equation can be discretized over a small time interval (τ_i) as in *Equation 2.24*. By rearranging the terms, the equation becomes to *Equation 2.25*.

$$\frac{r_{cb}^{n}(t_k) - r_{cb}^{n}(t_k+\tau_i)}{\tau_i} = \frac{v_{cb}^{n}(t_k) + v_{cb}^{n}(t_k+\tau_i)}{2} \approx v_{cb}^{n}(t_k+\tau_i) \tag{2.24}$$

$$r_{cb}^{n}(t_k+\tau_i) = r_{cb}^{n}(t_k) + v_{cb}^{n}(t_k+\tau_i)\tau_i \tag{2.25}$$

2.2.5 Inertial Navigation Errors

The errors in the navigation solution originate from three main sources: IMU measurement, initialization, and processing errors. The navigation error for a position, velocity, and acceleration solution can be defined as the difference between the actual value and the INS solution as in *Equation 2.26* Where the $\tilde{x}$ represents the erroneous solution for the quantity, and the x represents the true value of that quantity. In order to represent the attitude errors, the DCM error can be represented as in *Equation 2.27*. For small angle errors, the Euler angle errors and the DCM error can be related by *Equation 2.28* [37]. The erroneous transformation matrix can be represented in terms of Euler angle errors as in *Equation 2.29* using *Equations 2.27 & 2.28* .

$$\delta x = \widetilde{x} - x \tag{2.26}$$

$$\delta C_{\alpha}^{\gamma} = \widetilde{C_{\alpha}^{\gamma}} C_{\gamma}^{\alpha} \tag{2.27}$$

$$[\delta\psi_{\gamma\alpha}\times] = \delta\Psi_{\gamma\alpha}^{\gamma} \approx \delta C_{\alpha}^{\gamma} - I_3 \tag{2.28}$$

$$\widetilde{C_{\alpha}^{\gamma}} \approx [I_3 + \delta\Psi_{\gamma\alpha}^{\gamma}] C_{\alpha}^{\gamma} \tag{2.29}$$

The IMU measurement errors can be separated into constant bias errors, wide band sensor noise errors, scale factor errors, misalignment errors and bias instability errors [39].

- **Accelerometer Measurement Errors:**

  The accelerometer inside the IMU measures the specific force with respect to the inertial frame. The specific force measurement error is defined similarly to acceleration and velocity errors (*Equation 2.30*). The measurement output can be modeled using different error sources as in *Equation 2.31* where b_a represents the effect of constant (δb_{as}) and in-run bias errors (δb_{ad}), M_a represents the effects of scale factor and misalignment errors, and the w_a represents the random walk error. The velocity random walk error can be modeled as white noise.

$$\delta f_{cb}^{b} = \widetilde{f_{cb}^{b}} - f_{cb}^{b} \tag{2.30}$$

$$\widetilde{f^b_{cb}} = f^b_{cb} + b_a + M_a f^b_{cb} + w_a \tag{2.31}$$

$$M_a = \begin{bmatrix} s_{a,x} & m_{a,xy} & m_{a,xz} \\ m_{a,yx} & s_{a,y} & m_{a,yz} \\ m_{a,zx} & m_{a,zy} & s_{a,z} \end{bmatrix} \tag{2.32}$$

- **Gyroscope Measurement Errors:**

  The gyroscope inside the IMU measures the rotation rate with respect to the inertial frame. The measurement error is defined similarly to acceleration and velocity errors (*Equation 2.33*). The measurement output can be modeled using different error sources as in *Equation 2.34* where b_g represents the effect of constant (δb_{gs}) and in-run bias errors (δb_{gd}), M_g represents the effects of scale factor and misalignment errors, and the w_g represents the random walk error. The angular random walk error can be modeled as white noise.

$$\delta w^b_{cb} = \widetilde{w^b_{cb}} - w^b_{cb} \tag{2.33}$$

$$\widetilde{w^b_{cb}} = w^b_{cb} + b_g + M_g w^b_{cb} + w_g \tag{2.34}$$

$$M_g = \begin{bmatrix} s_{g,x} & m_{g,xy} & m_{g,xz} \\ m_{g,yx} & s_{g,y} & m_{g,yz} \\ m_{g,zx} & m_{g,zy} & s_{g,z} \end{bmatrix} \tag{2.35}$$

2.2.6 Transposition of Navigation Solution

For most applications, estimation of the navigation of one point in a system is necessary for reasons such as cost-effectiveness. The calculation of Euler angles between frame-α and frame-γ from the attitude between frame-α, and frame-β is straightforward if the transformation between frame-β and frame γ is known:

$$C^\gamma_\alpha = C^\gamma_\beta C^\beta_\alpha \tag{2.36}$$

$$r^\gamma_{\alpha\beta} = r^\gamma_{\alpha\eta} + r^\gamma_{\eta\beta} \tag{2.37}$$

Similarly, the position vector between frame-α and frame-γ can easily be calculated from the position vectors between frames-$\alpha\&\beta$ and frames-$\beta\&\gamma$ if the vectors are both resolved in the same coordinate frame (*Equation 2.37*). Furthermore, if they are not resolved in the same frame, they can easily be transported to the desired coordinate frame using *Equation 2.1.* However, the calculation of velocity with respect to another reference frame is not as easy as finding a position. Using *Equations 2.9 & 2.37* , the derivative of the position can be calculated with two different method as in *Equation 2.38.*

$$\begin{aligned} v^{\gamma}_{\alpha\beta} + w^{\gamma}_{\gamma\alpha} \times r^{\gamma}_{\alpha\beta} &= \frac{d}{dt} r^{\gamma}_{\alpha\beta} \\ &= \frac{d}{dt}(r^{\gamma}_{\alpha\eta} + r^{\gamma}_{\eta\beta}) \\ &= \frac{d}{dt}(C^{\gamma}_{\alpha} r^{\alpha}_{\alpha\eta} + C^{\gamma}_{\eta} r^{\eta}_{\eta\beta}) \\ &= C^{\gamma}_{\alpha} v^{\alpha}_{\alpha\eta} + \dot{C}^{\gamma}_{\alpha} r^{\alpha}_{\alpha\eta} + C^{\gamma}_{\eta} v^{\gamma}_{\eta\beta} + \dot{C}^{\gamma}_{\eta} r^{\eta}_{\eta\beta} \\ &= v^{\gamma}_{\alpha\eta} + v^{\gamma}_{\eta\beta} + w^{\gamma}_{\gamma\alpha} \times r^{\gamma}_{\alpha\eta} + w^{\gamma}_{\gamma\eta} \times r^{\gamma}_{\eta\beta} \end{aligned} \tag{2.38}$$

By rearranging *Equation 2.38*, the formula for velocity transportation for a general case where the reference frame and resolving frame are not identical and rotate with respect to another can be obtained:

$$v^{\gamma}_{\alpha\beta} = v^{\gamma}_{\alpha\eta} + v^{\gamma}_{\eta\beta} + w^{\gamma}_{\gamma\alpha} \times r^{\gamma}_{\beta\eta} + w^{\gamma}_{\gamma\eta} \times r^{\gamma}_{\eta\beta} \tag{2.39}$$

For a specific case where the reference frame and resolving frame are identical, the identity becomes:

$$v^{\alpha}_{\alpha\beta} = v^{\alpha}_{\alpha\eta} + v^{\alpha}_{\eta\beta} + w^{\alpha}_{\alpha\eta} \times r^{\alpha}_{\eta\beta} \tag{2.40}$$

Similarly, by taking the derivative of *Equation 2.40*, the acceleration transportation for the specific case where the reference frame and resolving frame are identical can be obtained. Each term in the acceleration transport identity in the *Equation 2.41* have special names in the literature [40], the names of corresponding terms can be examined in *Table 2.1*.

Table 2.1: The Components of the Acceleration Transfer Identity

Acceleration of Frame η	$a^{\alpha}_{\alpha\eta}$
Local Acceleration of Frame β w.r.t. η	$a^{\alpha}_{\eta\beta}$
Coriolis Acceleration	$2w^{\alpha}_{\alpha\eta} \times v^{\alpha}_{\eta\beta}v$
Euler Acceleration	$\dot{w}^{\alpha}_{\alpha\eta} \times r^{\alpha}_{\eta\beta}$
Centripetal Acceleration	$w^{\alpha}_{\alpha\eta} \times w^{\alpha}_{\alpha\eta} \times r^{\alpha}_{\eta\beta}$

$$a^{\alpha}_{\alpha\beta} = a^{\alpha}_{\alpha\eta} + a^{\alpha}_{\eta\beta} + 2w^{\alpha}_{\alpha\eta} \times v^{\alpha}_{\eta\beta} + \dot{w}^{\alpha}_{\alpha\eta} \times r^{\alpha}_{\eta\beta} + w^{\alpha}_{\alpha\eta} \times w^{\alpha}_{\alpha\eta} \times r^{\alpha}_{\eta\beta} \tag{2.41}$$

2.2.7 Ackermann Steering Geometry

Ackermann steering geometry was developed by Rudolph Ackermann for horse-drawn carriages in 1818 [41] to solve the problem of determining the turn angle of both inside and outside wheels in order for the vehicles to follow the circle of different radii. The physical illustration of the Ackermann geometry is given in *Figure 2.3*. For simulation and control application, instead of controlling two different wheels, an imaginary central wheel could represent the behavior of both wheels [42]. The relations between steering angles of the front wheels and the steering angle of the imaginary center wheel are given in *Equation 2.42*. Similarly, an imaginary central wheel can be controlled to control the linear velocity of the wheels. The relations between linear velocities of the rear wheels and linear velocity of the imaginary center wheel are given in *Equation 2.43*. The detailed analysis of Ackermann identities is presented in **Appendix A**.

$$\begin{aligned}\tan\delta_L &= \frac{L\tan\delta_C}{L + W/2\tan\delta_C}\\ \tan\delta_R &= \frac{L\tan\delta_C}{L - W/2\tan\delta_C}\end{aligned} \tag{2.42}$$

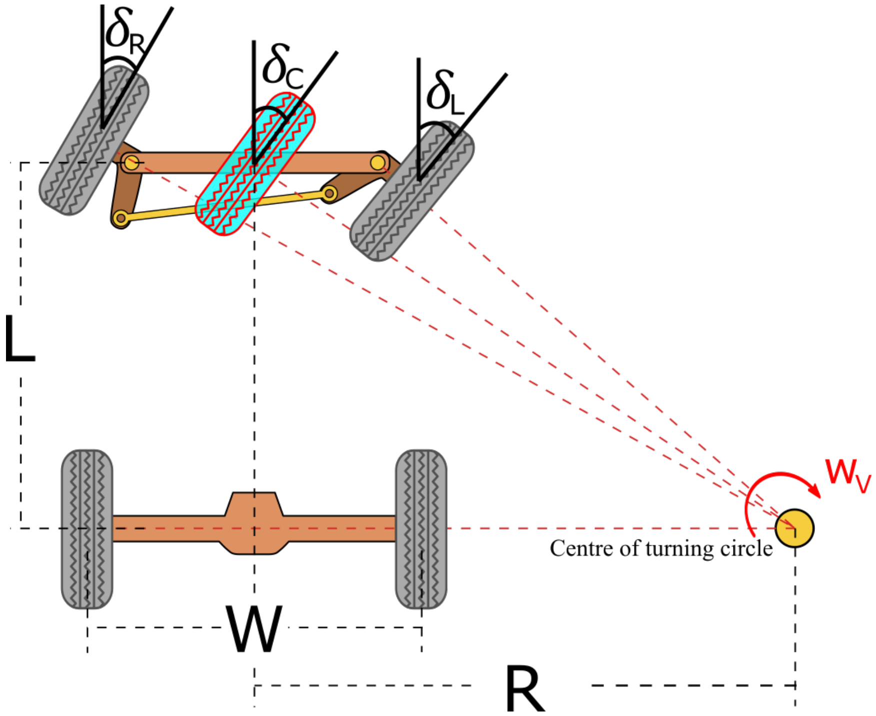

Figure 2.3: Ackermann Steering Geometry [1]

$$
\begin{aligned}
v_L &= v_C \;\frac{L + W/2\tan\delta_C}{L} \\
v_R &= v_C \;\frac{L - W/2\tan\delta_C}{L}
\end{aligned}
\tag{2.43}
$$

2.3 Tire Dynamics

The dynamics can be defined as the study of forces and their effects on motion. The dynamic equations of motion describe the relationship between the contact forces and the actuation acting on the body and the motion resulting from them [36]. In this section, general concepts for longitudinal and lateral tire dynamics are introduced. Then, the tire models used to estimate the tire contact forces are discussed.

Tires are the only contact points between the roads and the vehicle. Therefore, the performance of a vehicle is directly influenced by the characteristics of the tire [2]. Furthermore, the tire can be considered an essential part of a vehicle for simulation, control, and performance evaluation applications since the quality of the tire model directly affects the performance and accuracy of vehicle simulations.

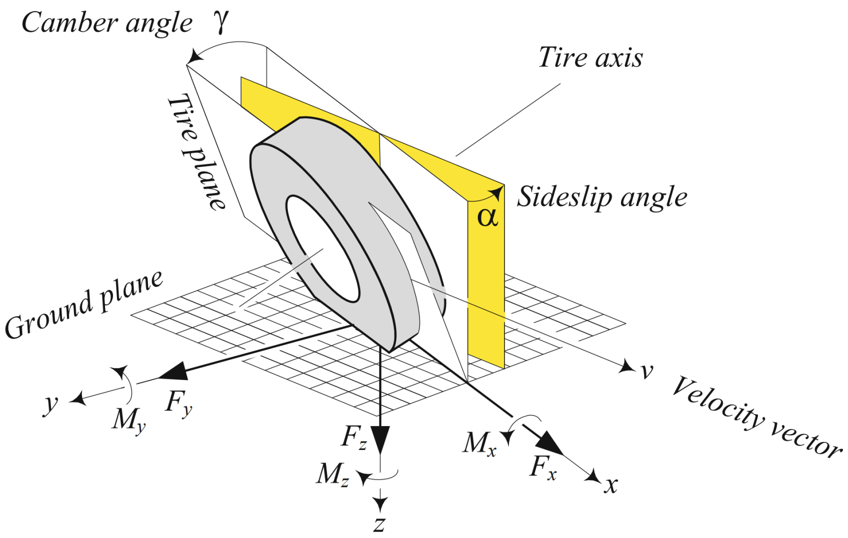

Figure 2.4: Tire Dynamics [2]

The 6-DOF dynamic model of the tire can be described by three contact forces and three moments. The forces and torques that define the tire dynamics are summarized in *Figure 2.4*. These forces and torques can be defined on the tire frame defined in *Section 2.2.1*.

The longitudinal contact force (F_x) is a force acting on the x-axis of the tire with respect to the inertial frame. If the longitudinal force is positive, the vehicle is accelerating. Furthermore, if the force is negative, the vehicle is breaking. The lateral contact force (F_y) is the complementary force acting on the y-axis of the tire with respect to the inertial frame. By definition, F_y is orthogonal to F_x, and the vertical force (F_z) is the normal force at the surface tire contact area. This force can also be called *"wheel load"* [2]. The lateral and longitudinal forces can be modeled as a function of vertical wheel load.

The moment about the x-axis of the tire is referred to as the roll moment (M_x). Roll moment causes the tire to turn with respect to its x-axis and creates a pitch angle for the tire orientation. The pitch angle of the tire can sometimes be referred to as the camber angle. This moment, for this study, is assumed to be zero. The pitch moment (M_y) is the moment about the y-axis of the tire. The Pitch moment causes the tire to turn with respect to its y-axis and causes the tire rotation. This moment can be used to rotate the wheels with desired rotational velocity. The z moment, or Yaw Moment

(M_z), causes the tire to turn with respect to its z-axis and causes to change of tire heading angle. Yaw moment can be used to rotate the tires to the desired steering angle.

2.3.1 Longitudinal Tire Dynamics

The longitudinal tire force F_x is a friction force from the tire surface contact. The experimental results have proved that the longitudinal tire force depends on the longitudinal slip ratio (σ_x^t), the normal load (F_z), and the friction coefficient (μ) of the tire-surface contact area [43].

The longitudinal slip of a tire can be defined as the difference between its longitudinal velocity at its axle (v_x^t) and its equivalent rotational velocity ($R_{eff}w^t$) from the rotational velocity of the wheel (w^t) where R_{eff} is defined as the effective radius of the tire. The physical representation of the longitudinal slip is given in *Figure 2.5*. Therefore, the longitudinal slip ratio (σ_x^t) for a tire can be defined to be the ratio of longitudinal slip and longitudinal velocity as in *Equation 2.44* [2].

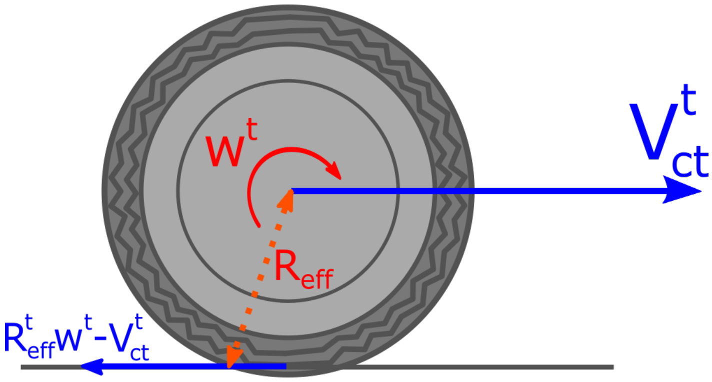

Figure 2.5: Tire Longitudinal Slip

$$\sigma_x^t = \frac{R_{eff}w^t - v_x^t}{v_x^t} \tag{2.44}$$

The dependency of the longitudinal tire force to the longitudinal slip ratio for a typical case is given in the *Figure 2.6*. As can be seen from the figure, for a small slip values, the longitudinal tire force can be modelled as a linear function of the slip ratio. However, small-slip assumption is not valid for all cases. Therefore, nonlinear tire models such as the Pacejka *"Magic Formula"* [44] model and Dugoff tire model [43] are used to calculate the longitudinal tire models.

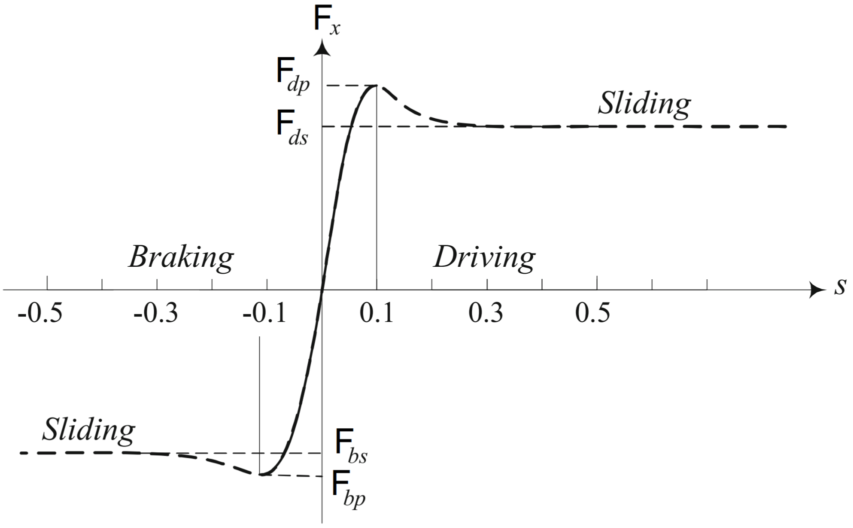

Figure 2.6: Longitudinal Slip Ratio vs Longitudinal Force [2]

Although there are many analytical models to describe the characteristics of the tire, they do not always lead to accurate results for every situation [45]. Due to their various assumptions, these models can not model the behavior of the tire under high or combined slip situations. To solve these problems, Hans Pacejka, a Professor from the Delft University of Technology, proposed an empirical tire model. Although the proposed model has no particular physical basis, it can represent various tire constructions and operating conditions; therefore, this model is famously known as the *Magic Formula*.

The Pacejka tire model to calculate the longitudinal tire force uses the longitudinal tire slip ratio as an input and is given in *Equation 2.45*. The model parameters of the tire models are called the stiffness factor (B_σ^t), shape factor (C_σ^t), peak value (D_σ^t), and curvature factor (E_σ^t). The relation between these parameters and the normal load (F_z) is given in *Equation 2.46* [44]. Finally, the dependency of the longitudinal tire force on the model parameters C_σ & E_σ is given in *Figures 2.7 & 2.8.*

$$F_x^t = D_\sigma^t \sin(C_\sigma^t \tan^{-1}(B_\sigma^t \sigma^t - E_\sigma^t(B_\sigma^t \alpha_\sigma^t - \arctan(B_\sigma^t \sigma^t)))) \tag{2.45}$$

$$
\begin{aligned}
B_\sigma &= \frac{C_{F\sigma}}{C_\sigma \; D_\sigma} \\
C_{F\sigma} &= c_8 \; F_z \\
D_\sigma &= \mu \; F_z
\end{aligned}
\tag{2.46}
$$

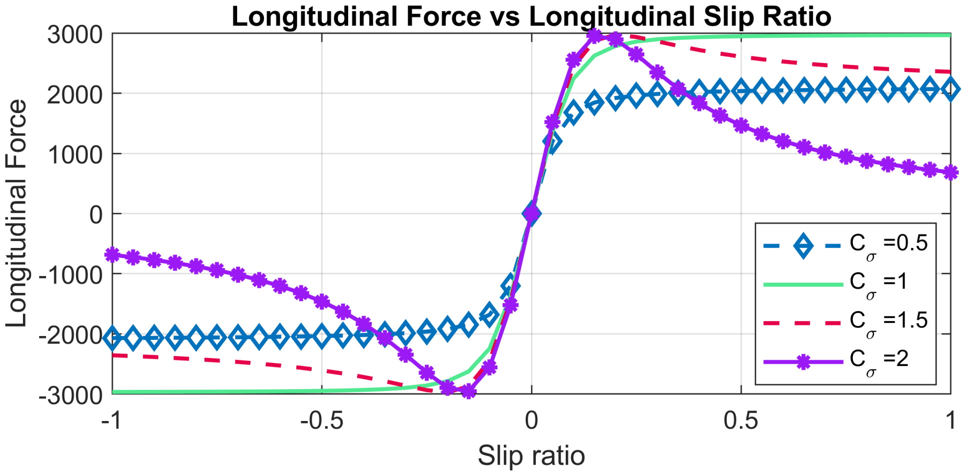

Figure 2.7: Pacejka Longitudinal Force Model Dependency on C_σ

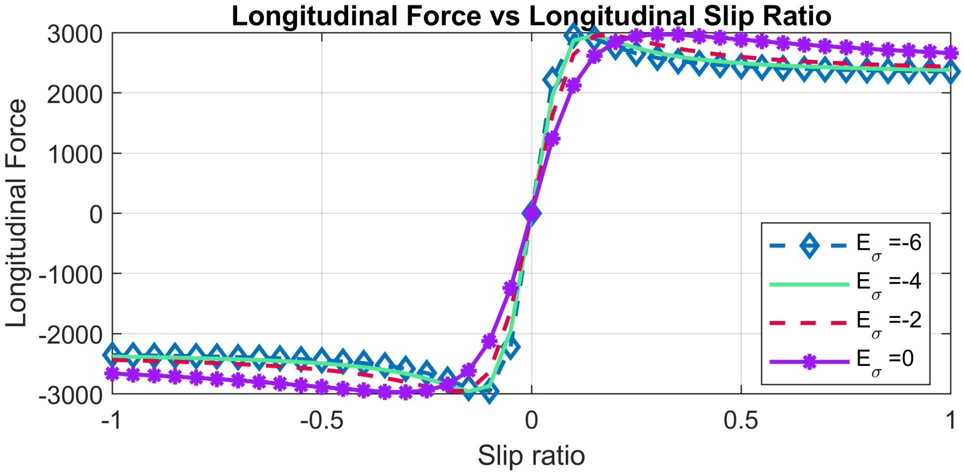

Figure 2.8: Pacejka Longitudinal Force Model Dependency on E_σ

2.3.2 Lateral Tire Dynamics

The lateral tire force F_y is a reaction force at the surface contact to side-slip angle. The side-slip angle (α^{t_k}) of a tire can be defined as the difference between its orientation

($\psi^n_{nt_k}$) and the orientation of its velocity vector ($v^n_{et_k}$). The physical representation of the side-slip angle is given in *Figure 2.10*. Therefore, the side slip angle can be calculated using *Equation 2.47* [2]. The physical realization of the stress caused by the side-slip angle and resultant lateral force is given in *Figure 2.9*. Similarly, the experimental results have showed that the lateral tire force depends on the side-slip angle (α^t), the normal load (F_z), and the friction coefficient (μ) of the tire-surface contact area [43].

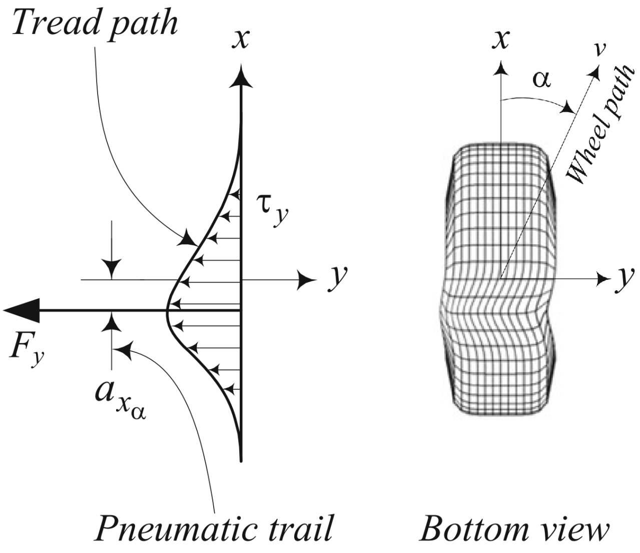

Figure 2.9: The Stress Distribution and the Resultant Lateral Force [2]

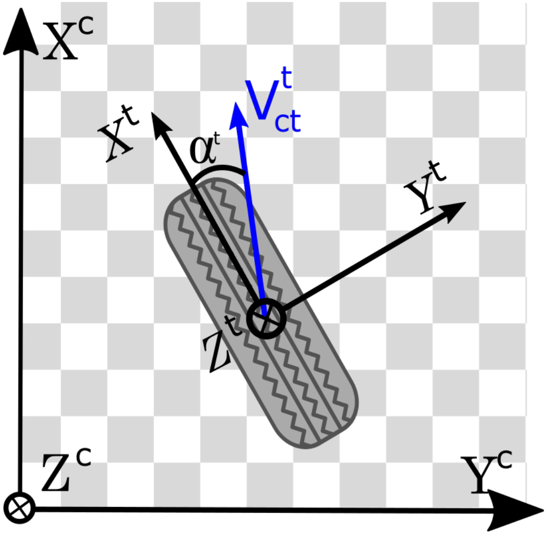

Figure 2.10: Tire Side-Slip Angle

$$\alpha^t = \psi_{nt}^n - \text{arctan2}\left(v_{et,y}^n, v_{et,x}^n\right) \tag{2.47}$$

Similar to longitudinal force calculation, the Pacejka tire model can be used to calculate the lateral tire force. Although the formula is in the same format, the *Magic Formula* to calculate lateral tire forces uses the side-slip angle as an input. The tire model for lateral force calculation is given in *Equation 2.48*. The relation between model parameters and the normal load is similar to the longitudinal case and is given in *Equation 2.49* [44]. The dependency of the lateral tire force computed by the Pacejka tire model on the parameters C_α & E_α is given in *Figures 2.11 & 2.12*.

$$F_y^t = D_\alpha^t \sin(C_\alpha^t \tan^{-1}(B_\alpha^t \alpha^t - E_\alpha^t(B_\alpha^t \alpha^t - \arctan(B_\alpha^t \alpha^t)))) \tag{2.48}$$

$$\begin{aligned} B_\alpha &= C_{F\alpha}/C_\alpha/D_\alpha \\ C_{F\alpha} &\approx c_1\ c_2\ \sin(2\ \arctan(1/c_2))\ F_z \\ D_\alpha &= \mu\ F_z \end{aligned} \tag{2.49}$$

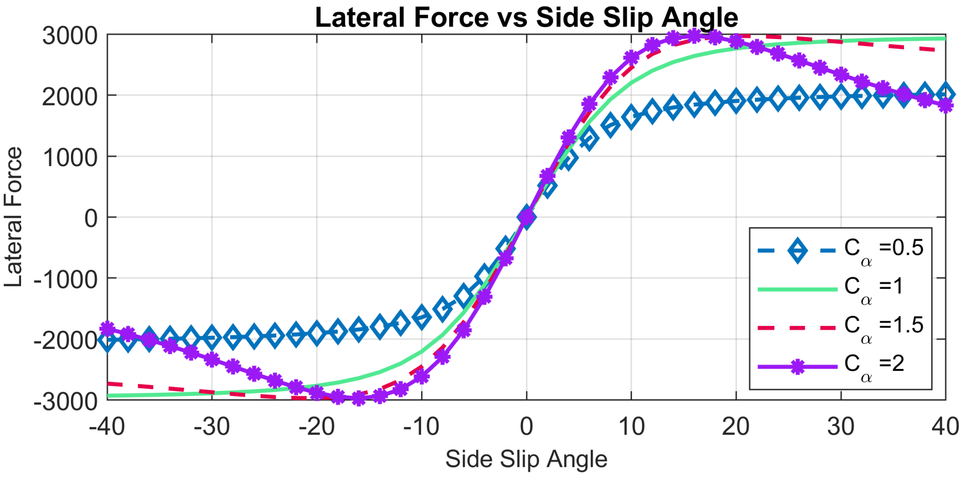

Figure 2.11: Pacejka Lateral Force Model Dependency on C_α

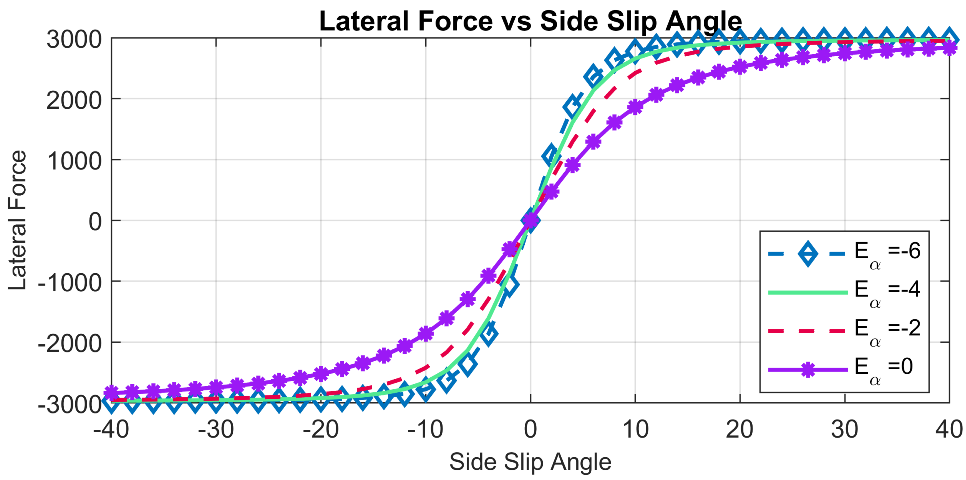

Figure 2.12: Pacejka Lateral Force Model Dependency on E_α

2.4 Kalman Filtering

Kalman filter or Linear Quadratic Estimator (LQE) is a recursive Bayesian estimation method [46], that was developed by R.Kalman [47] in 1960 to solve the discrete data linear filtering problem. At the beginning of the section, the linear Kalman filter (LKF) equations are provided. Then, the extended Kalman filter (EKF), an extension of LKF for the nonlinear problems, is provided.

2.4.1 Linear Kalman Filter

Solutions of a vector differential equation in the form of *Equation 2.50* that can describe a linear dynamic system have been the focus of filter development for many decades. One of the first filter methods to solve this kind of problem was introduced by Norbert Wiener in the 1940s for stationary signals [48][49].

$$\begin{aligned} \dot{x(t)} &= F(t)x(t) + D(t)u(t) \\ y(t) &= D(t)x(t) \end{aligned} \tag{2.50}$$

The linear or discrete Kalman filter is firstly developed to solve the corresponding for linear, non-stationary signals. The non-stationary signal is a signal whose properties vary with time. A signal with sinusoidal components whose phases, frequencies, and amplitudes vary with time can be an example of non-stationary signals [50]. The non-stationary dynamic system can be modelled in discrete domain analogously as in *Equation 2.51* where $\Phi(s;t)$ represents the transition matrix of the system from time t to time s (*Equation 2.52*) and the $w(t)$ is the zero-mean, independent Gaussian stochastic process that can be described by *Equation 2.53* [47].

$$\begin{aligned} x(t+1) &= \Phi(t+1;t)x(t) + w(t) \\ y(t) &= D(t)x(t) \end{aligned} \tag{2.51}$$

$$\Phi(t+1;t) = exp(F) = \sum_{i=0}^{\infty} F^i/i! \tag{2.52}$$

$$\begin{aligned} E\{w(t)\} &= 0 \quad \forall t \\ E\{w(t)w^T(t+\tau)\} &= \begin{cases} Q(t), & \tau = 0 \\ 0, & \tau \neq 0 \end{cases} \end{aligned} \tag{2.53}$$

The problem of the discrete Kalman filter can be defined as to estimation of the process that can be modeled by the difference equation in *Equation 2.54* from the measurements (observations) of the process. The observations are assumed to be the linear functions of the states as in the vector equation in *Equation 2.55*. Where x_k represents the $n \times 1$ process state vector at time $t = t_k$, Φ_k represents the $n \times n$ state transition matrix from t_k to $t_k + 1$ ($\Phi(t_k + 1; t_k)$), H_k represents the measurement matrix ($H(t)$) at time $t = t_k$ and the w_k and v_k represents the white sequences with known covariance and zero cross-covariance with respect to each other. While w_k can be used to model the noise on the system, v_k can be used to model the noise on the measurements. The discrete-time notation used throughout the study follows the similar notation, such that a_k represents the vector $a(t)$ at time $t = t_k$ and the A_k represents the matrix $A(t)$ at time $t = t_k$.

$$x_{k+1} = \Phi_k x_k + w_k \tag{2.54}$$

$$z_k = H_k x_k + v_k \tag{2.55}$$

The covariance matrices of the vectors w_k and v_k are defined by the from *Equation 2.56* to *Equation 2.58.* The Q_k matrix is called *the system noise covariance matrix*, and the R_k matrix is called *the measurement noise covariance matrix* which are essential parts of Kalman filtering.

$$E\{w_k w_i^T\} = \begin{cases} Q_k, & i = k \\ 0, & i \neq k \end{cases} \tag{2.56}$$

$$E\{v_k v_i^T\} = \begin{cases} R_k, & i = k \\ 0, & i \neq k \end{cases} \tag{2.57}$$

$$E\{w_k v_i^T\} = 0 \quad \forall k, i \tag{2.58}$$

The LKF is a recursive filter; thus, the prior estimate about state x is assumed to be known to estimate the state's current value. The estimated state at k_{th} discrete step can be denoted by $\hat{x_k}$ while the last known estimate of the state prior to measurement can be denoted by $\hat{x_k}^-$. One of the key aspects of Kalman filtering is the knowledge of error covariance of the system. In order to calculate the error covariance matrix, the estimation error prior to measurement can be defined as in *Equation 2.59.* And, the error covariance matrix (P_k) can be defined as in *Equation 2.60* with an assumption that the estimation error has zero mean.

$$e_k^- = x_k - \hat{x_k}^- \tag{2.59}$$

$$P_k^- = E\{e_k^- {e_k^-}^T\} = E\{(x_k - \hat{x_k}^-)(x_k - \hat{x_k}^-)^T\} \tag{2.60}$$

The Kalman filter proposes the estimate of the current state from the knowledge of prior state estimate and the measurement as in *Equation 2.61* where the K_K is called the Kalman gain. The purpose of the Kalman gain is to minimize the diagonal elements of the error covariance matrix P_k such that the estimation error is minimized.

$$\hat{x_k} = \hat{x_k}^- + K_k(z_k - H_k\hat{x_k}^-) \tag{2.61}$$

The estimation error covariance to be minimized is defined in *Equation 2.62*. The estimation error term can be expanded using *Equation 2.61* to found a relationship between Kalman gain and error covariance matrix (*Equation 2.63*).

$$\begin{aligned} P_k &= E\{e_k {e_k}^T\} \\ &= E\{[x_k - \hat{x_k}][x_k - \hat{x_k}]^T\} \\ &= E\{[e_k^- - K_k(H_k e_k^- + v_k)][e_k^- - K_k(H_k e_k^- + v_k)]^T\} \end{aligned} \tag{2.62}$$

$$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k {K_k}^T \tag{2.63}$$

The optimal gain matrix K_k that minimizes the mean-square estimation error is given in *Equation 2.64* [46]. Then using the *Equations 2.63 & 2.64* error covariance matrix update can be written in terms of prior measurement error covariance and Kalman gain as in *Equation 2.65*.

$$K_k = P_k^- {H_k}^T (H_k P_k^- {H_k}^T + R_k)^{-1} \tag{2.64}$$

$$P_k = (I - K_k H_k) P_k^- \tag{2.65}$$

The state estimate can be propagated to next sampling time using the *Equation 2.66* while the error covariance matrix can be propagated using *Equation 2.67* [46]. The general flow of the Linear Kalman Filter loop is given in *Figure 2.13*.

$$\hat{x}_{k+1}^- = \Phi_k \hat{x}_k \tag{2.66}$$

$$P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k \tag{2.67}$$

2.4.2 Extended Kalman Filter

The main assumption of the discrete Kalman filter is that the dynamic system is linear and can be modeled by such equations. However, in real-life applications where Kalman filtering is needed most, the systems are generally highly nonlinear. A Kalman filter developed for nonlinear systems by linearizing the Kalman filter about

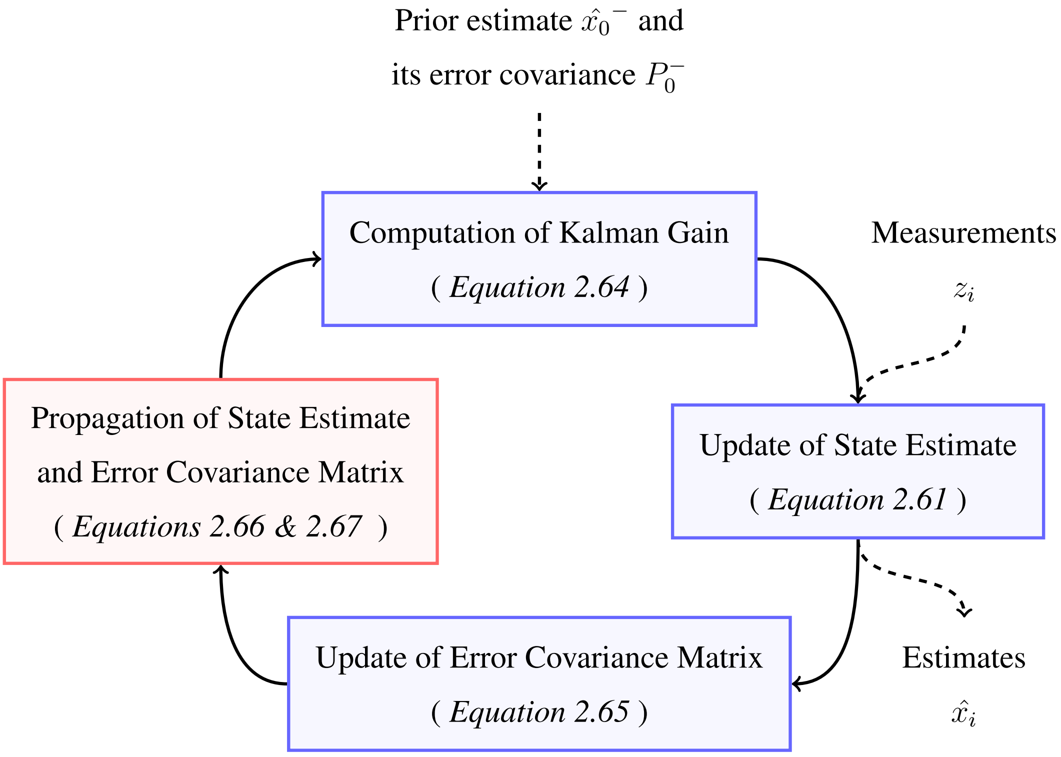
Prior estimate $\hat{x_0}^-$ and
its error covariance P_0^-
Computation of Kalman Gain
(*Equation 2.64*)
Measurements
z_i
Update of State Estimate
(*Equation 2.61*)
Propagation of State Estimate
and Error Covariance Matrix
(*Equations 2.66 & 2.67*)
Estimates
$\hat{x}_i$
Update of Error Covariance Matrix
(*Equation 2.65*)

Figure 2.13: Kalman Filter Loop

the current mean and covariance is called Extended Kalman Filter (EKF). A stochastic non-linear system can be described by *Equation 2.68*. The measurements of a nonlinear system can be modeled by a nonlinear function of the system states and noise as in *Equation 2.69*.

$$\dot{x}(t) = f(x(t), t) + g(w(t), t) \tag{2.68}$$

$$z(t) = h(x(t), t) + v(t)) \tag{2.69}$$

Like linear systems, nonlinear systems can be modeled in discrete domain analogously as in *Equation 2.70*. Then, the main objective of EKF can be summarized as observing the states of such systems with the knowledge of measurements that can be described by *Equation 2.71*.

$$x_{k+1} = f(x_k, t_k) + g(w_k, t_k) \tag{2.70}$$

$$z_k = h(x_k, t_k) + v_k \tag{2.71}$$

However, in the EKF, the linearization process can be applied to system error dynamics instead of system dynamics. The reasoning behind this idea can be reduced to the fact that the error in the state estimate can be assumed to be much smaller than the state vector itself [37]. Thus, the extended Kalman filter can be used to estimate the errors of the states instead of directly observing the states themselves. The state error to be observed can be defined by the difference between the actual value of the state and the observation of the state itself (*Equation 2.72*). The error state dynamics can be linearized in continuous time domain similar to state itself as in *Equation 2.73*. The $F(t)$ and $G(t)$ matrices can be calculated by taking partial derivative of nonlinear system dynamic functions "f" and "g" with respect to state and noise vector itself (*Equations 2.74 & 2.75*).

$$\delta x(t) = x(t) - \hat{x}(t) \tag{2.72}$$

$$\delta \dot{x}(t) = F(t)\delta x(t) \; + \; G(t)w(t) \tag{2.73}$$

$$F(t) = \left. \frac{\partial f(x,t)}{\partial x} \right|_{x=\hat{x}(t)} \tag{2.74}$$

$$G(t) = \left. \frac{\partial g(w,t)}{\partial x} \right|_{w=w(t)} \tag{2.75}$$

Similar to state error linearization, the measurement innovations become much smaller than the actual measurements as the state estimate becomes more accurate [37]. Thus, the linearization of the innovation term becomes possible around the state estimate. The measurement innovation term is defined as the difference between the actual measurement and the estimate of that measurement (*Equation 2.76*). Similar to $F(t)$ and $G(t)$ matrices, the $H(t)$ matrix can be calculated by taking partial derivative of nonlinear measurement dynamic function "h" with respect to state vector itself (*Equation 2.77*).

$$\begin{aligned} \delta z(t) &= z(t) - h(\hat{x}(t),t) \\ &= h(x(t),t) \; + \; v(t) \; - \; h(\hat{x}(t),t) \\ &= H(t)\delta x(t) \; + \; v(t) \end{aligned} \tag{2.76}$$

$$H(t) = \left. \frac{\partial h(x,t)}{\partial x} \right|_{x=\hat{x}(t)} \tag{2.77}$$

Similar to the discrete Kalman filter, the problem of the extended Kalman filter can be defined as to estimation of the nonlinear process that can be modeled by the linear difference equation in *Equation 2.78* from the measurements innovation of the form *Equation 2.81*. The discrete-time state transition matrix (Φ_k) is calculated by *Equation 2.79* and the discrete system noise distribution matrix is defined by *Equation 2.80* [37].

$$\delta x_{k+1} = \Phi_k \delta x_k + \Gamma_k w_k \tag{2.78}$$

$$\Phi_k = \text{exp}(F_k \tau) \tag{2.79}$$

$$\Gamma_k w_k = \int\limits_{t_k - \Delta t}^{t_k} \text{exp}\left(F_k(t_k - \tau)\right) G_k \; w(\tau) \; d\tau \tag{2.80}$$

$$\begin{aligned} \delta z_k^- &= z_k - H_K \hat{x}_k^- \\ &= H_K \delta x_k^- + v_k \end{aligned} \tag{2.81}$$

The *the system noise covariance matrix* ($Q_{D,k}$) for the linearized discrete time system can be found by *Equation 2.82* [51] where the Q_k is equal to covariance of noise vector w_k and can be found by *Equation 2.56* and the *the measurement noise covariance matrix* (R_k) matrix for the linearized system can be found similar to LKF (*Equation 2.57*).

$$\begin{aligned} Q_{D,k} &= E\{\Gamma_k w_k {w_k}^T {\Gamma_k}^T\} \\ &\approx \Phi_k \; G_k \; Q_k \; {G_k}^T \; {\Phi_k}^T \; \Delta t \end{aligned} \tag{2.82}$$

The estimation of the current state error from the knowledge of prior state error estimate and the measurement innovations can be achieved similar to discrete Kalman Filter as in *Equation 2.83* where the K_K is the optimal gain matrix K_k that minimizes the mean-square estimation error (*Equation 2.64*) [46].

$$\begin{aligned}\delta\hat{x_k} &= \delta\hat{x_k}^- + K_k(z_k^- - H_k\hat{x_k}^-) \\ &= \delta\hat{x_k}^- + K_k\delta z_k^-\end{aligned} \tag{2.83}$$

In order to calculate the error covariance for the state error estimates, the error term can be defined as the difference between the actual state error and the state error estimate (*Equation 2.84*). The error covariance matrix can be defined as in *Equation 2.85*. The error covariance matrix update for EKF is similar to that of LKF and can be written in terms of prior measurement error covariance and Kalman gain as in *Equation 2.65*.

$$e_k^- = \delta x_k - \delta\hat{x_k}^- \tag{2.84}$$

$$P_k^- = E\{e_k^- {e_k^-}^T\} = E\{(\delta x_k - \delta\hat{x_k}^-)(\delta x_k - \delta\hat{x_k}^-)^T\} \tag{2.85}$$

The error covariance matrix of the extended Kalman filter is propagated similarly to LKF as in *Equation 2.86*. The difference arises from the definition of noise elements within the discrete time state dynamics.

$$P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_{D,k} \tag{2.86}$$

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Introduction

The main aim of the study is the estimation of vehicle signals for advanced driver-assistance system (ADAS) applications such as anti-lock braking system (ABS), electronic stability program (ESP), and adaptive cruise control (ACC). The signals to be estimated can be divided into two parts; vehicle navigation solution and tire contact variables. In the scope of this study, an extended Kalman filter structure is proposed to estimate the position, velocity, and attitude of the vehicle using IMU and GPS measurements. Then, a method for observing the tire slip variables and lateral and longitudinal contact forces is proposed using the estimated navigation solution, steering angle sensor, and wheel speed encoder measurements. The overall block diagram of the study and the expected outputs are given in *Figure 3.1.* In the given figure, the rounded squares represent the sensors used, and the rectangles represent the proposed algorithms with their respective sections. Moreover, the block in the far right represents observed variables in the scope of the study in *Figure 3.1.*

3.2 Kinematic Model of the Vehicle

Kinematics equations can be used to represent the movement of a body or a vehicle in a predefined space. In the scope of this study, the kinematics equation consists of position and velocity dynamics with respect to the inertial frame and attitude dynamics with respect to the local navigation frame. *Equations 2.23, 2.20 and 2.11* allow the 6-DOF kinematic model of a vehicle to be modeled as in *Equation 3.1.* The study

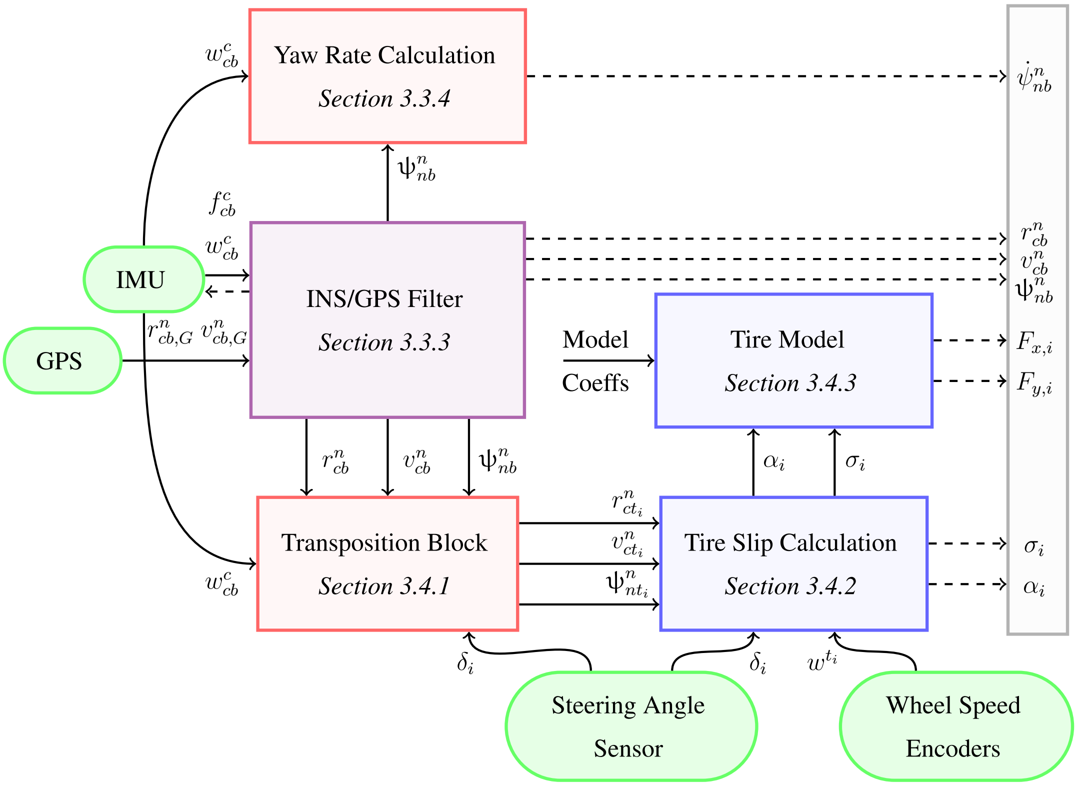

Figure 3.1: Overall Block Diagram of the Study

aims to observe these variables that represent the kinematics of the vehicle with the help of Kalman filtering.

$$\begin{aligned} \dot{r}^n_{cb} &= v^n_{cb} \\ \dot{v}^n_{cb} &= a^n_{cb} \\ \dot{C}^n_b &= C^n_b w^n_{nb}\times \end{aligned} \tag{3.1}$$

3.3 Estimation of Vehicle Navigation Solution

In this study, an extended Kalman filter solution is proposed to observe the navigation solution of the vehicle. This section begins with the introduction of the sensors used for the filtering. Then, the implementation of the inertial navigation solution is introduced. Lastly, a loosely coupled INS/GPS filter architecture is proposed to fuse the inertial navigation solution produced from IMU and GPS solution received from the GPS receiver.

3.3.1 Sensors

The sensors used in autonomous vehicles can be divided into exteroceptive and proprioceptive sensors [52]. The exteroceptive sensors measure the state of surroundings, such as mapping and temperature, while the proprioceptive sensors measure the vehicle states, such as velocity and position. This section introduces IMU and GPS receiver, the proprioceptive sensors on the vehicle used for the proposed Kalman filter.

Inertial Measurement Unit (IMU):

The inertial Measurement Unit (IMU) is a sensor that consists of a couple of (one for each axis) accelerometers, gyroscopes, and necessary electronic components. The IMU is used to measure the translational and rotational movements of an object or body. [53]. The output of a typical IMU consists of a specific force and angular rate of its body. In the scope of the study, it is assumed that the IMU is rigidly attached to the vehicle and measures the angular rate and linear acceleration of the vehicle with respect to the inertial frame. The performance specifications of the IMU used for the study are given in *Table 3.1*. The performance specifications in *Table 3.1* were selected considering the performance of some of the IMUs used in the automotive industry [54, 55, 56, 57] and performance of tactical grade IMUS's [37].

Table 3.1: IMU Performance Specifications

Specification	**Symbol**	**Value**
Sampling Frequency	f_{IMU}	100 Hz
Accelerometer Static Bias Error Std.	σ_{bas}	5 mg
Gyroscope Static Bias Error Std.	σ_{bgs}	10 deg/hr
Accelerometer Random-walk Error Std.	σ_{arw}	0.06 $ft/s/\sqrt{hr}$
Gyroscope Random-walk Error Std.	σ_{grw}	0.125 $deg/\sqrt{hr}$

GPS Receiver:

Global Navigation Satellite Systems (GNSS) is a general term for radio signal based positioning systems that use satellite signals to produce 3-D position and time information of an object on Earth [37, 58]. Global Positioning System (GPS), developed by the United States of America, is the most used and recognized GNSS [59]. GLONASS (Russia), BeiDou (China), and Galileo (EU) are the other GNSS examples that offer global coverage [60].

GPS receivers can produce a fixed position of an object anywhere on Earth as long as they can receive radio signals from at least four GPS satellites [38]. GPS receivers are commonly used for localization and navigation applications due to their high accuracy position solutions. Even the low-cost signal-frequency GPS receivers can achieve position accuracy of about 3 m [52]. However, the GPS receiver outputs cannot be used directly for navigation solutions due to their weaknesses, such as low sampling frequency and limited measurement bandwidth. In the scope of the study, it is assumed that the GPS is rigidly attached to the vehicle and outputs the position and velocity of the vehicle with respect to the inertial frame. The performance specifications of the GPS receiver used for the study is chosen considering [61, 56] and is given in *Table 3.2*.

Table 3.2: GPS Receiver Performance Specifications

Specification	**Symbol**	**Value**
Sampling Frequency	f_{GPS}	$5\ Hz$
Position Bias Error Std.	$\sigma_{GPS,pb}$	$0.25\ m$
Position Drift Error Std.	$\sigma_{GPS,pd}$	$1.5\ m$
Velocity Error Std.	$\sigma_{GPS,v}$	$0.1\ m/s$

3.3.2 Inertial Navigation Solution

The navigation solution produced from the IMU that is attached (*or strapped*) to the body is sometimes referred by *"Strapdown"* Inertial Navigation Solution [62]. However, due to their nature, IMU outputs contain some errors, such as those mentioned in *Section 2.2.5*. The errors in the IMU measurements cause the inertial navigation solution error to grow exponentially. For instance, a bias error in the acceleration measurement causes a linear growth in velocity solution error and quadratic growth in the position solution error [37, 62]. The INS output can be produced at a frequency that is only limited by the sampling frequency of the IMU. Although the navigation solution produced using only Strapdown INS algorithms have a very high output rate, the solution is prone to large errors. Therefore, in order to produce higher quality navigation solutions, the INS solution is generally corrected using other sensor data such as GPS solution, radar output, or images [63, 64, 58]. The simplified block diagram for the inertial navigation algorithm used for the proposed methodology is given in *Figure 3.2*.

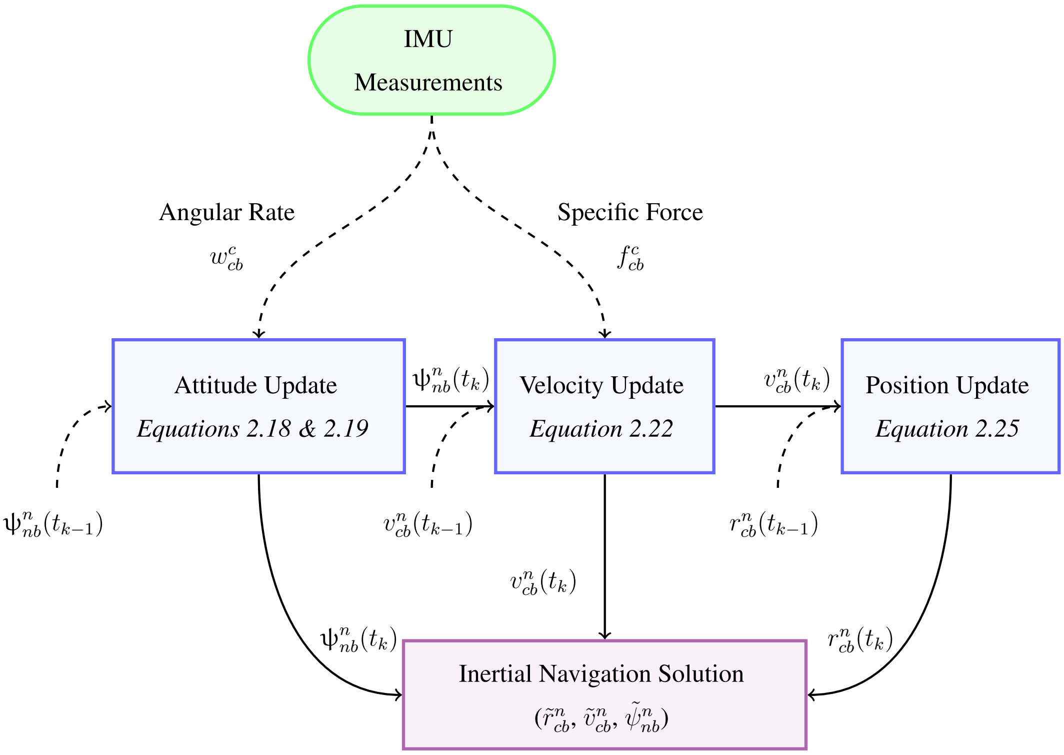

Figure 3.2: Strapdown Inertial Navigation Algorithm

3.3.3 Loosely Coupled INS/GPS Filter

Although the inertial navigation solution offers a high-frequency, high-bandwidth navigation solution, its output is too erroneous to be used for autonomous vehicle applications. On the contrary, the GPS offers a low-frequency, low-bandwidth navigation solution with minor, bounded errors. The advantages and disadvantageous of both navigation solutions make them ideal candidates to be fused by a filter. This section proposes a loosely coupled extended Kalman filter architecture to fuse the inertial navigation solution and GPS measurements. The general structure for the proposed filtering solution is given in *Figure 3.3*. The figure summarises the relationship between the sensors, the INS algorithm, and the propagation and update phases of the filter. The dashed lines in *Figure 3.3* represents the error state estimates of the filter.

While constructing an extended Kalman structure, state propagation, error covariance propagation, and measurement updates can be executed at different frequencies [62, 37]. For the proposed EKF, the state propagation is executed at 100-Hz, which is the sampling frequency of the IMU. The measurement update is executed as soon as a new measurement is available; thus, it is executed at 5-Hz. Furthermore, the error covariance is updated with the accumulation of 5 IMU data; thus, it is applied at 20-Hz.

3.3.3.1 Structure of the State Vector

The navigation solution to be estimated consists of position, velocity, and attitude. The dynamics of these quantities and their errors are highly nonlinear. Moreover, thanks to INS algorithm, there are rough estimates of the navigation solution before the filtering process begins. Therefore, for the INS/GPS filter, an extended Kalman filter can be used to estimate the error states instead of the states themselves.

With the state error estimates, the inertial navigation solution can be corrected to have a high-frequency, high-bandwidth, low-error navigation solution. Moreover, the bias errors in the IMU measurements can be estimated to calibrate the IMU. Considering these limitations, a state vector for the proposed EKF can be constructed to include the position, velocity, attitude, and acceleration bias, gyroscope bias. Thus, the state vector and error state vector to be estimated by the proposed INS/GPS Kalman filter are given in *Equation 3.2*.

$$x = \begin{bmatrix} r^n_{cb} \\ v^n_{cb} \\ \psi^n_{cb} \\ b_a \\ b_g \end{bmatrix} \quad , \quad \delta x = \begin{bmatrix} \delta r^n_{cb} \\ \delta v^n_{cb} \\ \delta \psi^n_{cb} \\ \delta b_a \\ \delta b_g \end{bmatrix} \tag{3.2}$$

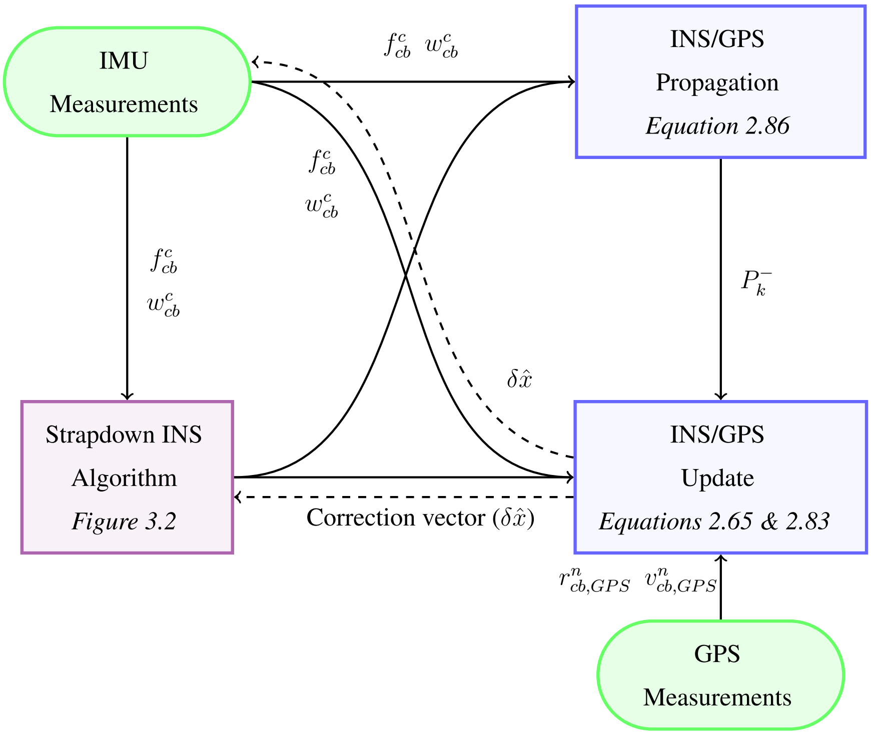

Figure 3.3: Loosely Coupled INS/GPS Filter Algorithm

3.3.3.2 Error State Dynamics

The error state dynamics can be modeled by a continuous nonlinear differential equation of form *Equation 2.68*. The differential equation can be written in a vector form as in *Equation 3.3*. In this section, the construction of nonlinear functions f_i and g_i is introduced.

$$\delta\dot{x} = \begin{bmatrix} \delta\dot{r}^n_{cb} \\ \delta\dot{v}^n_{cb} \\ \delta\dot{\psi}^n_{cb} \\ \delta\dot{b}_a \\ \delta\dot{b}_g \end{bmatrix} = \begin{bmatrix} f_1(\dot{x},t) + g_1(w,t) \\ f_2(\dot{x},t) + g_2(w,t) \\ f_3(\dot{x},t) + g_3(w,t) \\ f_4(\dot{x},t) + g_4(w,t) \\ f_5(\dot{x},t) + g_5(w,t) \end{bmatrix} \tag{3.3}$$

- **Position Error Dynamics:**

  From *Equation 2.23*, the derivative of the vehicle position in local navigation frame with respect to inertial cartesian frame is calculated by *Equation 3.4*. Position error can be defined using *Equation 2.26*. Then, the position error dynamics can be defined as in *Equation 3.5*. By making necessary simplifications, position error dynamics can be represented by *Equation 3.6*.

$$\dot{r}^n_{cb} = v^n_{cb} \tag{3.4}$$

$$\begin{aligned} \delta\dot{r}^n_{cb} &= \tilde{\dot{r}}^n_{cb} - \dot{r}^n_{cb} \\ &= \tilde{v}^n_{cb} - v^n_{cb} \end{aligned} \tag{3.5}$$

$$\delta\dot{r}^n_{cb} = \delta v^n_{cb} \tag{3.6}$$

- **Velocity Error Dynamics:**

  From *Equation 2.20*, the derivative of the vehicle linear velocity in local navigation frame with respect to inertial cartesian frame is calculated by *Equation 3.7*. Where f^b_{cb} corresponds to specific force measurement and g^n corresponds to gravitational acceleration. Velocity error can be defined using *Equation 2.26*. Then, the velocity error dynamics can be defined as in *Equation 3.9*. By making necessary simplifications, velocity error dynamics can be represented by *Equation 3.10*.

$$\dot{v}^n_{cb} = f^n_{cb} + g^n \tag{3.7}$$

$$a \times b = -b \times a \tag{3.8}$$

$$\begin{aligned}\delta\dot{v}_{cb}^{n} &= \dot{\tilde{v}}_{cb}^{n} - \dot{v}_{cb}^{n}\\ &= \tilde{C}_b^n \tilde{f}_{cb}^{b} - C_b^n f_{cb}^{n} + \tilde{g}^n - g^n\\ &= [I + \delta\Psi_{nb}^{n}]C_b^n \tilde{f}_{cb}^{b} - C_b^n f_{cb}^{n} + \delta g^n\\ &= C_b^n \tilde{f}_{cb}^{b} - C_b^n f_{cb}^{b} + \delta\Psi_{nb}^{n} C_b^n \tilde{f}_{cb}^{n} + \delta g^n \qquad (3.9)\\ &\approx C_b^n \delta f_{cb}^{b} + \delta\Psi_{nb}^{n} C_b^n f_{cb}^{n} + \delta g^n\\ &\approx C_b^n \delta f_{cb}^{b} - C_b^n f_{cb}^{n} \delta\psi_{nb}^{n} + \delta g^n\\ &\approx C_b^n \delta f_{cb}^{b} - C_b^n f_{cb}^{b} \delta\psi_{nb}^{n}\end{aligned}$$

$$\delta\dot{v}_{cb}^{n} \approx C_b^n \delta f_{cb}^{b} - f_{cb}^{n} \delta\psi_{nb}^{n} \tag{3.10}$$

- **Attitude Error Dynamics:**

  Attitude error can be related to DCM as in *Equation 3.12* using *Equation 2.29*. Where $\delta\Psi_{nb}^{n}$ represents the skew-symmetric form of attitude vector $\delta\psi_{nb}^{n}$ (*Equation 3.11*). The derivative of Euler error (*Equation 3.13*) can be simplified using *Equation 2.11*. Since both sides are skew-symmetric (*Equation 2.13*), the vector form of Euler error dynamics can be found as in *Equation 3.14*.

$$\delta\Psi_{nb}^{n} = \delta\psi_{nb}^{n}\times \tag{3.11}$$

$$\delta\Psi_{nb}^{n} = \tilde{C}_b^n (C_b^n)^T + I_3 \tag{3.12}$$

$$\begin{aligned}\delta\dot{\Psi}_{nb}^{n} &= \dot{\tilde{C}}_b^n C_b^{nT} + \tilde{C}_b^n \dot{C}_b^{n\,T}\\ &= (\tilde{C}_b^n \tilde{\Omega}_{cb}^{b}) C_b^{nT} + \tilde{C}_b^n (C_b^n \Omega_{cb}^{b})^T\\ &= \tilde{C}_b^n (\Omega_{cb}^{b} + \delta\Omega_{cb}^{b})(C_b^n)^T - \tilde{C}_b^n (C_b^n \Omega_{cb}^{b})(C_b^n)^T \qquad (3.13)\\ &= [I + \delta\Psi_{nb}^{n}] C_b^n \delta\Omega_{cb}^{b} (C_b^n)^T\\ &\approx C_b^n \delta\Omega_{cb}^{b} (C_b^n)^T\end{aligned}$$

$$\delta\dot{\psi}_{nb}^{n} = C_b^n \delta w_{cb}^{b} \tag{3.14}$$

- **Accelerometer Bias Error Dynamics:**

  The static part of accelerometer bias error (δb_{as}) is assumed to have a time-invariant characteristic; thus, its derivative is zero. Whereas the dynamic part (δb_{ad}) can be modelled as exponentially correlated fixed-variance first-order Markov processes [37]. With an approximation the accelerometer bias error dynamics can be written as *Equation 3.15*.

$$\delta\dot{b}_a \approx b_{ad} \tag{3.15}$$

- **Gyroscope Bias Error Dynamics:**

  The static gyroscope bias error, δb_{gd} is assumed to have a time-invariant characteristic [37]; thus, its derivative is zero. Whereas the dynamic part (δb_{ad}) can be modelled as exponentially correlated fixed-variance first-order Markov processes [37]. With an approximation the accelerometer bias error dynamics can be written as *Equation 3.16*.

$$\delta\dot{b}_a \approx b_{gd} \tag{3.16}$$

Therefore, the continuous time error state dynamics for the loosely coupled INS/GPS filter can be represented by *Equation 3.17*.

$$\delta\dot{x} \approx \begin{bmatrix} \delta v^n_{cb} \\ C^n_b \delta f^b_{cb} - f^n_{cb} \delta\Psi^n_{nb} \\ C^n_b \delta w^b_{cb} \\ b_{ad} \\ b_{gd} \end{bmatrix} \tag{3.17}$$

3.3.3.3 System Dynamics Matrix

The system dynamics matrix (F_k) can be found by taking the partial derivative of *Equation 3.17* with respect to state errors (*Equation 2.74*). The discrete-time system dynamic matrix for the loosely-coupled INS/GPS filter can be written as in *Equation 3.18*. The discrete-time transition matrix (Φ_k) can be computed using *Equation 2.79*.

$$F_k = \begin{bmatrix} 0_3 & I_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & -f^n_{cb} & C^n_b & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & C^n_b \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \end{bmatrix} \tag{3.18}$$

3.3.3.4 Process Noise Covariance Matrix

The process noise vector from dynamic accelerometer and gyroscope biases is defined in the form of *Equation 3.19*. Then, the noise distribution matrix (G_k) can be found by the taking partial derivative of *Equation 3.17* with respect to the process noise vector (*Equation 2.75*). The discrete-time noise distribution matrix for the loosely-coupled INS/GPS filter can be written as in *Equation 3.20*.

$$w = \begin{bmatrix} \delta b_{ad} \\ \delta b_{gd} \end{bmatrix} \tag{3.19}$$

$$G_k = \begin{bmatrix} 0_3 & 0_3 \\ C^n_b & 0_3 \\ 0_3 & C^n_b \\ I_3 & 0_3 \\ 0_3 & I_3 \end{bmatrix} \tag{3.20}$$

The process noise covariance matrix $Q_{D,k}$ can be calculated using *Equation 2.82*. The noise covariance matrix of the w_k is calculated as *Equation 3.21* using *Equation 2.56*. The process noise covariance matrix for the loosely-coupled INS/GPS filter can be written as in *Equation 3.22*, where σ_{bad} $\&$ σ_{bgd} represent the standard deviation of the accelerometer and gyroscope bias instability errors.

$$Q_k = \begin{bmatrix} E\{\delta b_{ad} \delta {b_{ad}}^T\} & 0_3 \\ 0_3 & E\{\delta b_{gd} \delta {b_{gd}}^T\} \end{bmatrix} \tag{3.21}$$

$$Q_k = \begin{bmatrix} \sigma^2_{bad} I_3 & 0_3 \\ 0_3 & \sigma^2_{bgd} I_3 \end{bmatrix} \tag{3.22}$$

3.3.3.5 Measurement Model

Most GPS receivers output not only position and time data but also velocity data [52]. The GPS receiver used in the simulation for this study can also publish the position and velocity of the vehicle. Therefore, the position and velocity solution of the GPS receiver is utilized for the EKF measurement. Since the GPS receiver and IMU coincide in the simulations, the GPS position and velocity are directly used as measurements without transposition. The measurements can be represented in a vector as in *Equation 3.23*, and the estimate of the same measurements can be represented in a vector as in *Equation 3.24*.

$$\tilde{m}_{GPS} = \begin{bmatrix} \tilde{r}^n_{cb,GPS} \\ \tilde{v}^n_{cb,GPS} \end{bmatrix} \tag{3.23}$$

$$\hat{m}_{INS} = h(x,t) = \begin{bmatrix} \hat{r}^n_{cb,INS} \\ \hat{v}^n_{cb,INS} \end{bmatrix} \tag{3.24}$$

The measurement vector z is defined as *Equation 3.25*. From *Equations 2.71 & 3.24* , the measurement estimate can be defined by *Equation 3.26*. Where the $v(t)$ is the measurement noise that can be modeled as white noise. Similarly, from *Equation 2.76*, the measurement innovation can be defined as in *Equation 3.27*.

$$z = \tilde{m}_{GPS} \tag{3.25}$$

$$\hat{z} = \hat{m}_{INS} + v(t) \tag{3.26}$$

$$\delta z = \tilde{m}_{GPS} - \tilde{m}_{INS} - v(t) \tag{3.27}$$

The discrete measurement innovation matrix, thus, is defined as:

$$\delta z_k^- = \begin{bmatrix} \tilde{r}^n_{cb,GPS} - \hat{r}^n_{cb,INS} \\ \tilde{v}^n_{cb,GPS} - \hat{v}^n_{cb,INS} \end{bmatrix} + v_k \tag{3.28}$$

From *Equations 2.77 & 3.24* , the discrete-time measurement matrix H_k is calculated to be *Equation 3.29*. Finally, the measurement noise covariance matrix can be calculated from *Equation 2.57*. The measurement matrix for the INS/GPS filter is given in *Equation 3.30*, where the $\sigma_{GPS,p}$ and $\sigma_{GPS,v}$ represents the GPS position and velocity error standard deviations.

$$H_k = \begin{bmatrix} -I_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & -I_3 & 0_3 & 0_3 & 0_3 \end{bmatrix} \tag{3.29}$$

$$R_k = \begin{bmatrix} \sigma^2_{GPS,p} I_3 & 0_3 \\ 0_3 & \sigma^2_{GPS,v} I_3 \end{bmatrix} \tag{3.30}$$

3.3.3.6 Error Covariance Propagation

As mentioned in *Section 2.4.2*, the error for the EKF error state estimates can be defined as in *Equation 2.86*. Furthermore, the error covariance matrix can be defined as in *Equation 2.85*. Like the general structure of the Kalman filters, the error covariance matrix can be calculated (or propagated) from the knowledge of the previous error covariance matrix and process noise covariance matrix (*Equation 2.86*). Therefore, it is necessary for any Kalman filter designer to define the initial error covariance matrix (P_0). To achieve this, the initial estimate error is defined as in *Equation 3.31*, where the initial error estimate is taken to be zero. Additionally, the error covariance matrix is calculated by *Equation 3.32*, where the δx_0 represents the initial state errors.

$$e_0^- = \delta x_0 - \cancelto{0}{\delta \hat{x}_0} = \delta x_0 \tag{3.31}$$

$$P_0^- = E\{e_0^- {e_0^-}^T\} = E\{\delta x_0 {\delta x_0}^T\} \tag{3.32}$$

Initially, the state errors are assumed to have zero cross-covariance with respect to each other; therefore, the cross terms in the initial error covariance matrix become zero (*Equation 3.33*). The initial error covariance for the loosely-coupled INS/GPS filter is given in *Equation 3.34*. The initial IMU error uncertainties in the P_0 matrix selected considering the IMU performance specifications in *Table 3.1*. Moreover, the initial navigation error uncertainties were considered tuning parameters of the filter and selected through experimentation.

used for the study are given in *Table 3.3*.

$$P_0^- = \begin{bmatrix} E\{\delta r_0 \delta {r_0}^T\} & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & E\{\delta v_0 \delta {v_0}^T\} & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & E\{\delta \psi_0 \delta {\psi_0}^T\} & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & E\{\delta b_{a0} \delta {b_{a0}}^T\} & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & E\{\delta b_{g0} \delta {b_{g0}}^T\} \end{bmatrix} \tag{3.33}$$

$$P_0^- = \begin{bmatrix} \sigma_{r0}^2 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & \sigma_{v0}^2 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & \sigma_{\psi 0}^2 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & \sigma_{bas}^2 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & \sigma_{bgs}^2 \end{bmatrix} \tag{3.34}$$

Table 3.3: Initial State Error Uncertainties

	Symbol	**Value**
Initial Position Error Uncertainty	σ_{r0}	$25\ m$
Initial Velocity Error Uncertainty	σ_{v0}	$2\ m/s$
Initial Attitude Error Uncertainty	$\sigma_{\psi 0}$	$100\ mrad$

3.3.3.7 Measurement Update

With every measurement received from the GPS receiver, the measurement innovation is computed by *Equation 3.28*. Then, update blocks of the Kalman filter can be executed (*Figure 2.13*). Firstly, the Kalman gain is computed using the measurement innovation and *Equation 2.64*. Next, using the calculated Kalman gain, the error state estimate can be computed (*Equation 2.83*). Since, after every measurement update, the error state estimates are used to correct the state estimated, the previous error state estimate is always a zero vector (*Equation 3.38*). Therefore, the error state estimate for the INS/GPS filter becomes *Equation 3.35*. Finally, the error covariance matrix is updated using *Equation 2.65*.

$$\delta\hat{x_k} = K_k \delta z_k^- \tag{3.35}$$

3.3.3.8 State Propagation and Correction

For the position, velocity and attitude states, the states are propagated using the INS algorithm. And, for the accelerometer and gyroscope biases, the state error estimates are accumulated using *Equation 3.36*. After every measurement update, the error state estimates are used to correct the erroneous state estimates. For position, velocity, accelerometer bias, and gyroscope bias estimate, the corrections are made using *Equation 3.37*. Since the attitude is propagated using the quaternions in the INS algorithm, the attitude corrections are applied using *Equation 3.39* [37]. Since every error state estimate is directly used for state corrections, the previous state error estimate is equated to a zero vector (*Equation 3.38*).

$$\hat{x_k} = x_{\hat{k-1}} + \delta\hat{x_k} \tag{3.36}$$

$$\hat{\xi} = \tilde{\xi} - \delta\hat{\xi} \tag{3.37}$$

$$\delta\hat{x_k}^- = 0_{15\times1} \tag{3.38}$$

$$\hat{q} = \tilde{q} + \frac{1}{2}\begin{bmatrix} +\delta\hat{\phi}\tilde{q_1} + \delta\hat{\theta}\tilde{q_2} + \delta\hat{\psi}\tilde{q_3} \\ -\delta\hat{\phi}\tilde{q_0} - \delta\hat{\theta}\tilde{q_3} + \delta\hat{\psi}\tilde{q_2} \\ +\delta\hat{\phi}\tilde{q_3} - \delta\hat{\theta}\tilde{q_0} - \delta\hat{\psi}\tilde{q_1} \\ -\delta\hat{\phi}\tilde{q_2} + \delta\hat{\theta}\tilde{q_1} - \delta\hat{\psi}\tilde{q_0} \end{bmatrix} \tag{3.39}$$

3.3.4 Estimation of the Vehicle Yaw Rate

The calculation of yaw rate is essential for vehicle stability control algorithms such as rollover prevention systems and ESP [65]. Moreover, the yaw plays an important role in path tracking and path following applications, since these kind of algorithms can use controllers to follow the path with desired yaw rate [66, 67]. The gyroscopes in the IMU measure the angular rate of the vehicle. From the knowledge of the attitude of the vehicle and calibrated gyroscope measurements, the yaw rate can be calculated using *Equation 2.17.*

3.4 Tire Dynamics

Forces at the tires can be represented using tire frames $"t_k", \ \ k \in \{1,2,3,4\}$ (*Figure 2.2*). In the scope of this study, the 1^{st} wheel represents the front left, the 2^{nd} wheel represents the front right, the 3^{rd} wheel represents the rear left, and the 4^{th} wheel represents the rear right wheel in the equations. Tire forces F^{t_k} can be decomposed as in *Equation 3.40.* Tire forces, like acceleration, can be represented in body frame or navigation frame using the direction cosine matrices (*Equation 3.41*).

$$F^{t_k} = F^{t_k}_{ct_k} = \begin{bmatrix} F^{t_k}_x \\ F^{t_k}_y \\ F^{t_k}_z \end{bmatrix} \tag{3.40}$$

$$\begin{aligned} F^b_{ct_k} &= C^b_{t_k} F^{t_k}_{ct_k} \\ F^n_{ct_k} &= C^n_b F^b_{ct_k} \end{aligned} \tag{3.41}$$

3.4.1 Transposition of Vehicle Navigation Solution To Tires

The estimated navigation solution in *Section 3.3.3* represents the position, velocity, and attitude of the body frame, which is centered at the IMU. However, in order to calculate the necessary tire variables, the velocity and Euler angles of the body are needed to be transported to the wheel bases. This can be achieved using the equations in *Section 2.2.6.*

Velocities and rotations at the tires can be represented using wheel centered wheel fixed $"w_k", \ \ k \in \{1,2,3,4\}$, and tire frames $"t_k", \ \ k \in \{1,2,3,4\}$ defined for each wheel (*Figure 2.2*). Rotation Matrices from body frame to k^{th} tire frame can be computed as in *Equation 3.42*, where the δ_k represents the steering angle of the k^{th} tire ($\delta_k = \psi^b_{t_k b}$). The steering angle of the front tires can be found using the steering angle sensor measurements. It can be noted that, for rear wheels where the steering angel is equal to zero, transformation matrix $C^b_{t_k}, \;\; kk \in \{3,4\}$ becomes a identity matrix.

$$C^b_{t_k} = \begin{bmatrix} \cos(\delta_k) & -\sin(\delta_k) & 0 \\ \sin(\delta_k) & \cos(\delta_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.42}$$

The position, velocity and the attitude of the vehicle can be transported to wheel center using the *Equations 2.37, 2.40, & 2.36* . The transposition equations from body frame to tire frames is given in *Equations 3.43, 3.44, & 3.45* . The Euler angles can be calculated from DCM as in *Equation 2.3*.

$$r^n_{ct_k} = r^n_{cb} + C^n_b r^b_{bt_k} \tag{3.43}$$

$$\begin{aligned} v^n_{ct_k} &= v^n_{cb} + \cancelto{0}{v^n_{bt_k}} + w^n_{cb} \times r^n_{bt_k} \\ &= v^n_{cb} + C^n_b w^b_{cb} \times (C^n_b r^b_{bt_k}) \end{aligned} \tag{3.44}$$

$$C^n_{t_k} = C^n_b C^b_{t_k} \tag{3.45}$$

3.4.2 Estimation of Tire Slip Variables

Slip variables of each tire can be calculated using the navigation solution of each wheel, speedometer measurements and the steering angle sensor measurements. The navigation solution for each wheel can be obtained by transposition of the estimated navigation solution of the vehicle as discussed at *Section 3.4.1*. Tire navigation solution includes the linear velocity ($v^n_{ct_k}$) and the attitude ($\psi^n_{nt_k}$). Tire linear velocities (v^{t_k}) defined for each tire in its respected tire frame ”t_k”, $\;\; k \in \{1,2,3,4\}$ can be represented as in *Equation 3.46*. Angular velocities w^{t_k} of each wheel can be represented as in *Equation 3.47* and can be measured directly by speedometers at each wheel base.

$$v^{t_k} = v^{t_k}_{ct_k} = C^{t_k}_n v^n_{ct_k} = \begin{bmatrix} v^{t_k}_x \\ v^{t_k}_y \\ v^{t_k}_z \end{bmatrix} \tag{3.46}$$

$$w^{t_k} = w^{t_k}_{t_k w_k} = \begin{bmatrix} w^{t_k}_x \\ w^{t_k}_y \\ w^{t_k}_z \end{bmatrix} \tag{3.47}$$

By the help of linear velocity and angular velocity at the wheel base, the longitudinal slip ratio of each wheel is calculated using *Equation 3.48.* Moreover, side-slip angles for each tire can be calculated using *Equation 3.49.*

$$\sigma^{t_k}_x = \frac{R_{eff} w^{t_k} - v^{t_k}_x}{v^{t_k}_x} \quad , \quad k \in \{1,2,3,4\} \tag{3.48}$$

$$\alpha^{t_k} = \psi^n_{nt_k} - atan2\left(v^n_{et_k,y}, v^n_{et_k,x}\right) \quad , \quad k \in \{1,2,3,4\} \tag{3.49}$$

3.4.3 Estimation of Tire Contact Forces

From the knowledge of longitudinal slip ratio ($\sigma^{t_k}_x$) and side-slip angle (α^{t_k}) at each tire t_k, the lateral and longitudinal contact forces for each tire can be calculated using the *Equations 2.45 & 2.48* . To tune the model parameters, the relations in the *Equations 2.46 & 2.49* was utilized. The $C_{F\alpha}, C_{F\sigma}, E_\alpha$ & $C_{F\alpha}$ parameters were chosen as tuning parameters and determined by experimentation. The model parameters used for the *Magic Formula* of Pacejka in the scope of the study are given in *Table 3.4.* Since each wheel is assumed to be identical in the simulations, the same model parameters are utilized for every tire.

Table 3.4: Pacejka Tire Model Parameters

Model Parameter	**Abbreviation**	**Value**
Longitudinal Stiffness Factor	B_σ	$(8\ F_z)/(C_\sigma\ D_\sigma)$
Longitudinal Shape Factor	C_σ	1.44
Longitudinal Peak Value	D_σ	$\mu\ F_z$
Longitudinal Curvature Factor	E_σ	-1
Lateral Stiffness Factor	B_α	$(6\ F_z)/(C_\alpha\ D_\alpha)$
Lateral Shape Factor	C_α	1.0
Lateral Peak Value	D_α	$\mu\ F_z$
Lateral Curvature Factor	E_α	-1

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Introduction

In order to evaluate the estimation performance of the proposed algorithms, the algorithms were tested in a simulation environment. The experiments were repeated multiple times for different driving scenarios to investigate the estimation performance in detail. In this chapter, firstly, the simulation environment built for the validation of the proposed algorithms is introduced. Then, the control scenario to evaluate estimations is introduced. Then, the detailed analysis of experiment results for the control scenario is explained. Finally, the results for four different driving scenario is presented.

4.2 Simulation Environment

In this section, the simulation environment built for the study is introduced. Firstly, the Robot Operating System 2 (ROS-2), a middleware solution used between the operating system and application, is introduced. Then, Gazebo, the simulation environment that governs the dynamics of the vehicle and sensor data, is introduced. Finally, the implementation of the vehicle model in Gazebo and algorithms in the ROS-2 environment is explained.

4.2.1 ROS-2

The implementation of middleware is essential for designing robotics applications such as autonomous vehicles (AVs) [68]. *Middleware* can be defined as the software layer that lies between the user applications and the operating system. As a hidden translation layer, it is used for data management and communication for distributed applications [69]. It can also be used to bind different services together [70].

The first generation of Robot Operating System (ROS) is one of the most widely used middleware solutions for self-driving cars in research, and industry [71]. Although the features like being open-source, hardware abstraction, message publishing-subscribing structure, and modular package management make ROS-1 a very prominent middleware in research and experiments, due to its design limitations, the ROS-1 is not suitable for real-time processing. Moreover, the need for a central node among ROS nodes risks the reliability of ROS-1 for real-life implementations [71].

The next generation of ROS (ROS-2) is developed by Open Robotics while considering the real-time processing requirements and other problems caused by the central architecture of ROS-1. For ROS-2, the underlying ROS transport system from ROS-1 is replaced by Data Distribution Service (DDS), an industry standard [3]. DDS is a data-centric publish-subscribe programming model for distributed systems. DDS has been applied for various domains, such as defense, automotive and simulation. The software architecture of ROS-2 is given in *Figure 4.1*. The sixth release of ROS-2, Foxy Fitzroy (released on June 5th, 2020), is utilized throughout the study due to being the latest release of the ROS-2 at the beginning of the study.

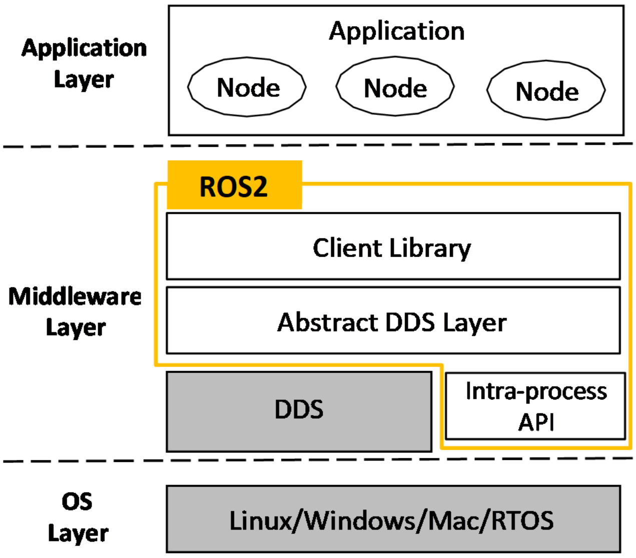

Figure 4.1: ROS-2 Architecture [3]

- **Application Layer:**

  Modular package management is a crucial factor in the popularity of ROS-2 [5]. The application layer of ROS-2 consists of nodes, topics, and services that allow users to divide the algorithms into reusable parts. A ROS-2 node can be summarised as an algorithm package responsible for a single and module purpose. Each node can send and receive data to other nodes via topics, services, actions, or parameters. Topics, another essential part of the ROS-2, act as a bus for nodes to exchange messages. Services are another method for exchanging messages between nodes. The services are based on a call-and-response model versus topics' publisher-subscriber model. Due to the real-time processing requirements of the sensor data, the topics are used instead of services throughout the study.

  ROS 2 can be used to break complex systems into small parts thanks to its modular structure. A developed robotic system such as an autonomous vehicle is comprised of many nodes working together through topics and services [4]. An example ROS graph is given in *Figure 4.2*. The rectangular boxes represent the ROS-2 topics, and the ellipses represent the ROS-2 nodes. The node can subscribe and publish more than one topic simultaneously, as in *Figure 4.2*.

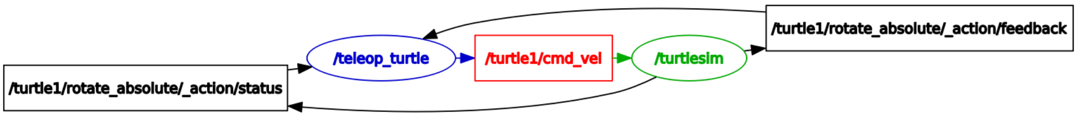
/turtle1/rotate_absolute/_action/status
/teleop_turtle
/turtle1/cmd_vel
/turtlesim
/turtle1/rotate_absolute/_action/feedback

Figure 4.2: An Example ROS-2 Graph [4]

- **Cyclone DDS:**

  Real-time sensor processing requires small delays; thus, ROS-2 nodes are sensitive to communication delays. ROS-2 supports four different DDS vendors in order to provide flexible choices for different use scenarios. The research in [3] concluded that the Cyclone DDS had superior performance to the default DDS (Fast DDS) in terms of latency. Moreover, the default RMW implementation since ROS-2 Galactic (released on May 23rd, 2021) is selected as Cyclone DDS [72]. For the reasons mentioned above, to ensure the study's rapid data transfer requirements, Cyclone DDS is chosen as a DDS layer of ROS-2 Foxy Fitzroy for the study.

- **Quality of Service (QoS) Settings:**

  ROS-2 allows users to tune the communication between nodes by defining Quality of Service (QoS). The tuning allows the ROS-2 communication to be as reliable as TCP or as best-effort as UDP, unlike ROS 1, which primarily only supported TCP [73]. The sensor data profile from the predefined QoS profiles of ROS-2 uses best-effort reliability and smaller queue size, and it is chosen for QoS of the ROS-2 in the study.

4.2.2 Gazebo Classic

The development of the Gazebo simulation environment started as part of a Ph.D. research project in 2004 [74]. The Gazebo provides a multi-robot simulation environment, including dynamics simulation, which is provided by the physics engines. Moreover, Gazebo allows users to choose from different physics engines for different needs. The software architecture behind Gazebo Simulator is given in *Figure 4.3*. A robot model to be simulated in the Gazebo Simulator environment can be defined as a collection of links (bodies), joints, and sensors. While the links represent the inertial, visual, and collision attributes of the robot, the links are responsible for the attachment of the links with each other. The joints inside the simulated robots can be rotational, translational, or combined joints. Furthermore, the sensor objects can be attached to the robot to collect the necessary data for the study. Moreover, as a simulation environment, Gazebo can independently consider gravity, contact forces, and friction.

Two methods can be used to define a model in the Gazebo Simulation environment. The first method is to utilize the Unified Robot Description Format (URDF), which is also compatible with different robotic simulators. Furthermore, the second method is to utilize the Simulation Description Format (SDF), which was first developed as a part of the Gazebo Simulator [75]. Unlike URDF, the SDF can describe everything from the world level down to the robot level, define friction properties, and specify the robot's pose within a world. Considering the mentioned reasons, the SDF is selected to describe the vehicle to be simulated in the study. The newer version of Gazebo, known as Ignition, was renamed the Gazebo as of June 2022. Therefore the Gazebo is now referred to as Gazebo Classic. The Gazebo (Classic) 11.0 was utilized to simulate the vehicle and sensors for the scope of the study.

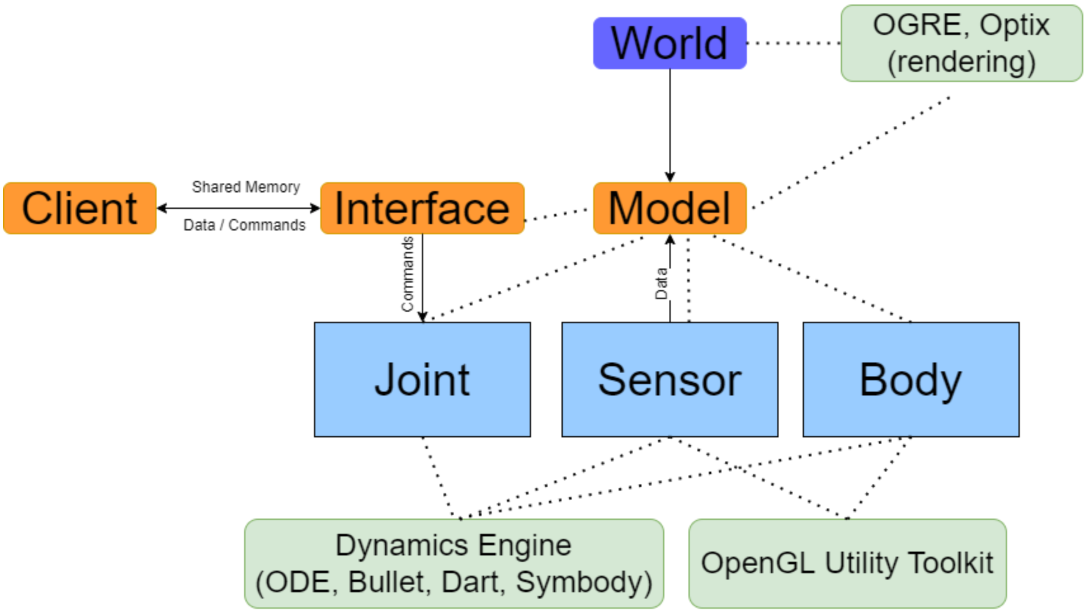

Figure 4.3: Gazebo Architecture [5]

4.2.3 Vehicle Model

The vehicle model in Gazebo is prepared using SDF. The vehicle model consists of links, joints, sensors, plug-ins, and ROS-2 nodes. In this section, the details of the links and joints of the vehicle are presented. Sensors and model plug-ins included in the vehicle model will be discussed in *Section 4.2.4* and ROS-2 nodes included will be discussed in *Section 4.2.5*. The general structure of the links and joints of the vehicle model is given in *Figure 4.6*, where rectangles represent links, and rounded rectangles represent the joints.

As mentioned in the previous section, every link has inertial, visual, and collision attributes in the SDF. The links' inertial attributes and collision attributes (except the friction parameters) are given as constant for the simulation. However, the friction parameters of the collision attribute are supplied to the simulation via a model plug-in. Moreover, lastly, visual attributes are selected from the stock vehicle models visuals of Gazebo (*Figure 4.4*). The selected model visual can be seen in *Figure 4.5*.

Table 4.1: Vehicle Parameters

Coefficient	**Abbreviation**	**Value**
Tire Radius	R_0	0.31265 m
Vehicle Width	w	1.572 m
Vehicle Front Half	a	1.144 m
Vehicle Rear Half	b	1.716 m
Vehicle Mass	m	1326 kg

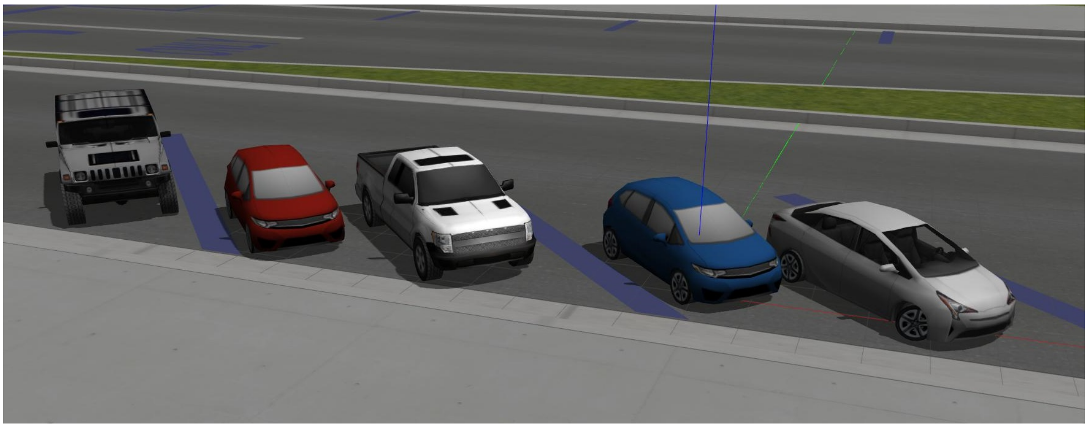

Figure 4.4: Gazebo Vehicle Visuals

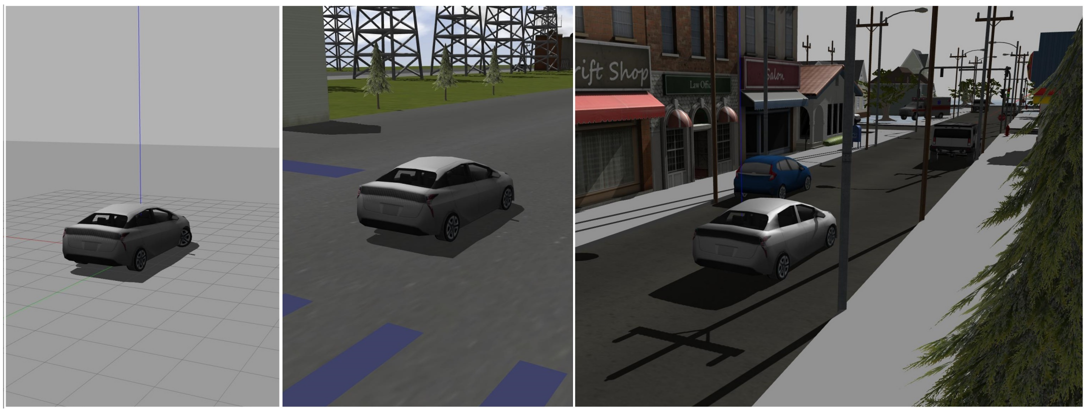

Figure 4.5: Kobra Mk5 in Different Environments

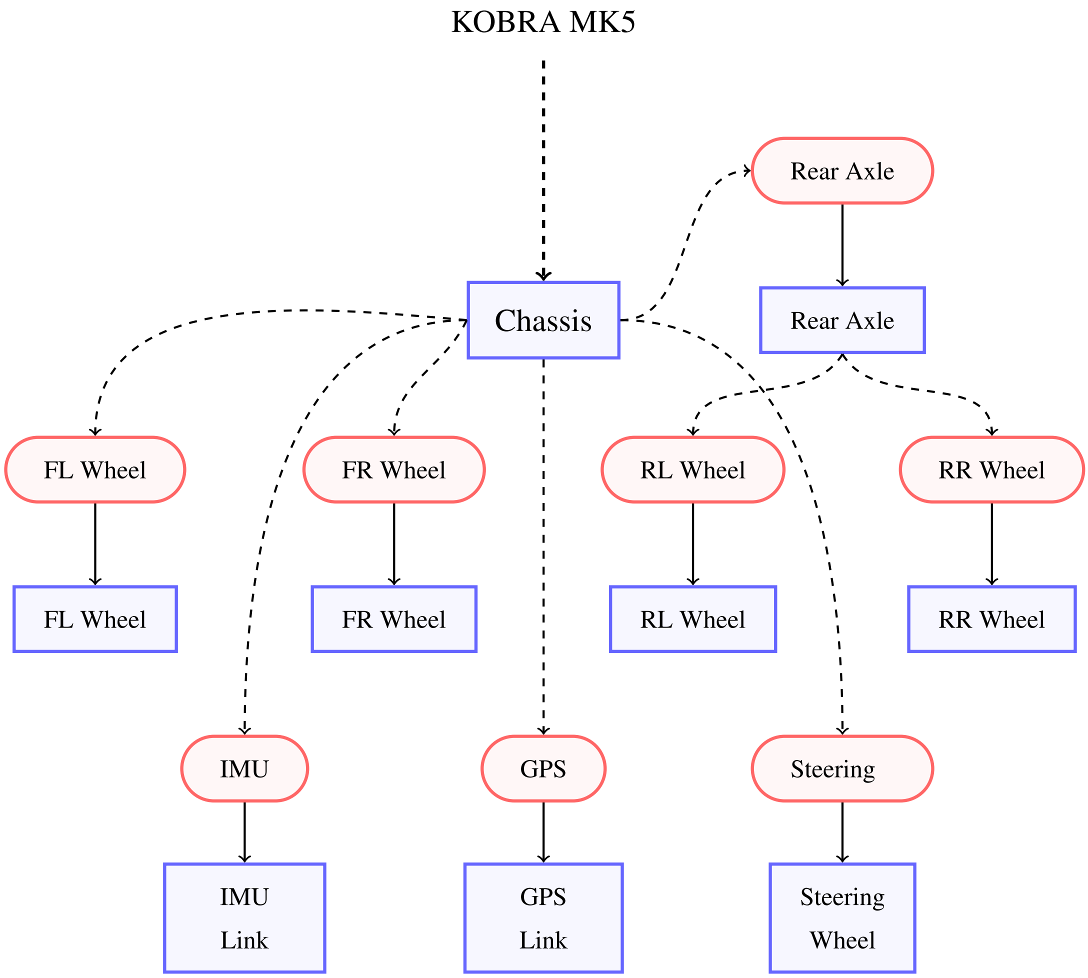

Figure 4.6: Vehicle Model Description

4.2.4 Gazebo Plug-ins

The usage of model plug-ins, allows user to access the background of the simulation. The usage of plug-ins also allows the robot model to communicate with ROS-2 nodes via ROS-2 transport system (DDS). The plug-ins are divided into three classes, model plug-ins, sensor plug-ins, and visual plug-ins. In the scope of the study, only model plug-ins and sensor plug-ins are used.

- **Ackermann Controller Plug-in:**

  The Ackermann controller plug-in utilizes *Equations 2.42 & 2.43* to control vehicle linear velocity and steering angle. The PID controller inside the plug-in is used to send controlled torque to wheel joints. The joints the plug-in controls are given in *Figure 4.7*.

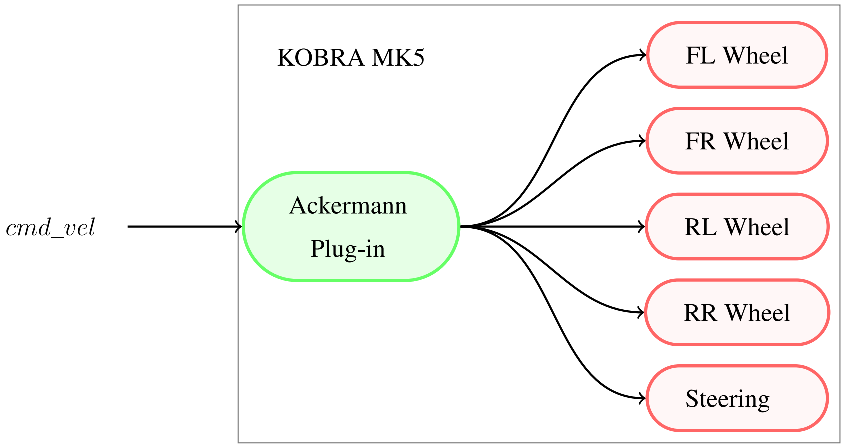

Figure 4.7: Ackermann Drive Plug-in

- **IMU & GPS Sensor Plug-ins:**

  The vehicle description includes an IMU sensor attached to the IMU link located between the rear wheels to publish the IMU Link's linear acceleration and rotation rate as a ROS-2 topic. This plug-in outputs the IMU measurements with errors defined in *Table 3.1*. Furthermore, a GPS receiver is included to publish the position and velocity of the GPS Link attached to the middle of the rear wheels as a ROS-2 topic. The GPS receiver plug-in outputs position and velocity information with errors defined in *Table 3.2*.

- **Joint State Publisher Plug-in:**

  The joint state publisher is used to mimic the wheel encoders and steering angle sensors. This plug-in is attached to the five links specified in *Figure 4.8*. Joint state publisher plug-in publishes the angular velocity of all wheels and steering angle of front wheels as a ROS-2 topic.

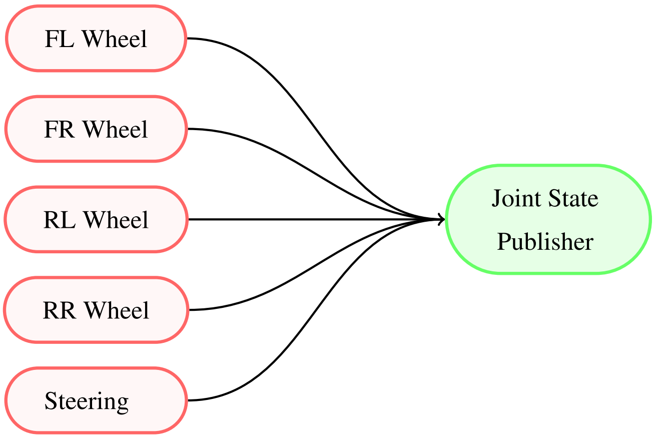

Figure 4.8: Gazebo ROS-2 State Publishers

- **Contact Surface Plug-in:**

  Contact surface plug-in is created to control friction parameters of the collision attributes of wheel links. The variables stated in the *Table 4.2* are supplied to the simulated vehicle as constants. However, the *slip1* and *slip2* parameters of the wheels are dynamically arranged as a function of wheel angular velocity and steering angle.

Table 4.2: Wheel Link Surface Constants

Coefficient	**Abbreviation**	**Value**
Mu Primary	μ_1	0.9
Mu Secondary	μ_2	0.9
Poisson's Ratio [76]	ν	0.5
Elastic Modulus [76]	E	$1e7$

4.2.5 ROS-2 Nodes

The ROS-2 nodes are created to run the proposed algorithm in simulation environment in real time. Three main nodes are created for the study and are introduced in this section. The ROS-2 graph of the stıdy is presented in *Figure 4.9*.

- **Scenario Publisher Node:** The scenario publisher node is crated to publish necessary ROS-2 topics that are subscribed to by the Ackermann Controller plug-in. It publishes desired linear velocity and steering angle of the vehicle.
- **INS/GPS Filter Node:** The INS/GPS Filter node is created to run the algorithm explained in *Section 3.3*. The algorithms in this node can be summarised by the *Figure 3.3*.
- **Tire Force Calculation Node:** The Tire Force Calculation node is created to run the algorithm explained in *Section 3.4*. The algorithms in this node can be summarised by the *Figure 3.1* except the INS/GPS filter part.

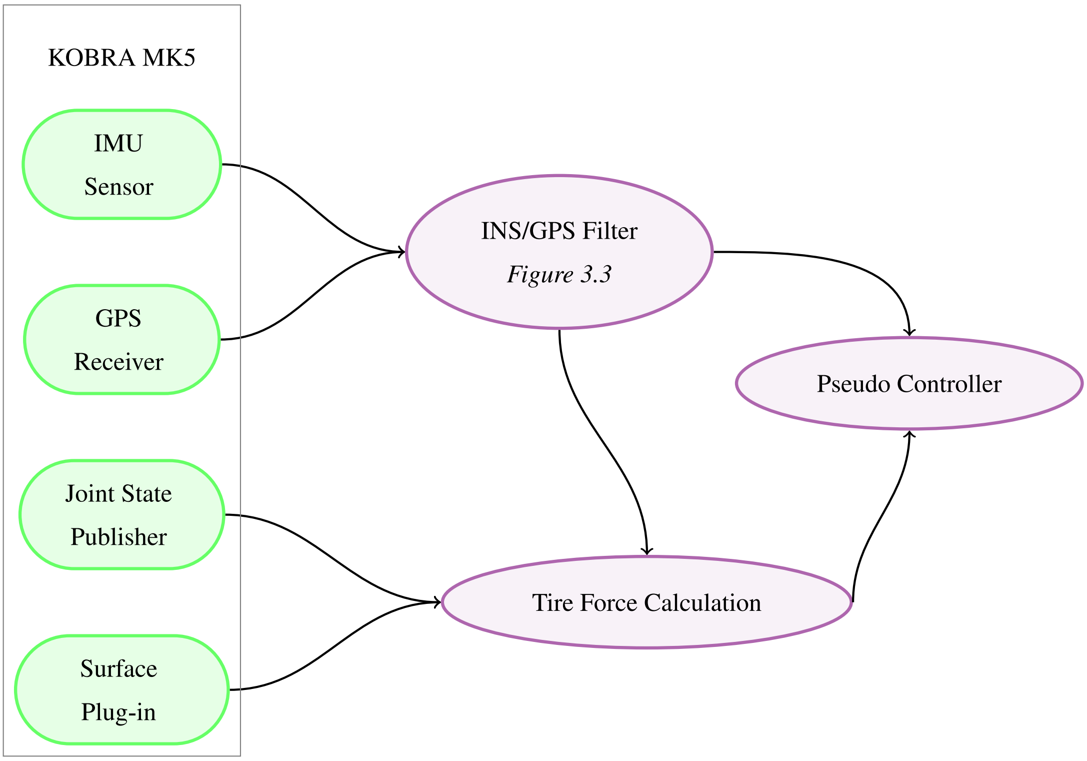

Figure 4.9: ROS-2 Graph

4.3 Driving Scenarios

For the development and validation of the proposed algorithm, simulations were conducted using different driving scenarios. For the performance evaluation, the simulations were repeated multiple times for each scenario. Therefore, driving scenarios that is repeatable and whose results are predictable are selected. In this section, the details of the selected scenarios are introduced.

4.3.1 Scenario 1: The Control Scenario

For a control scenario, a constant velocity constant steering angle scenario is selected in order to create a scenario where both lateral and longitudinal contact forces are constant throughout the scenario. The vehicle inputs that are controlled by the Ackermann drive plug-in are given in *Figure 4.10*. The vehicle route and heading throughout the scenario are given in *Figure 4.11*. As expected, the vehicle followed a circular path thanks to a constant steering angle and velocity.

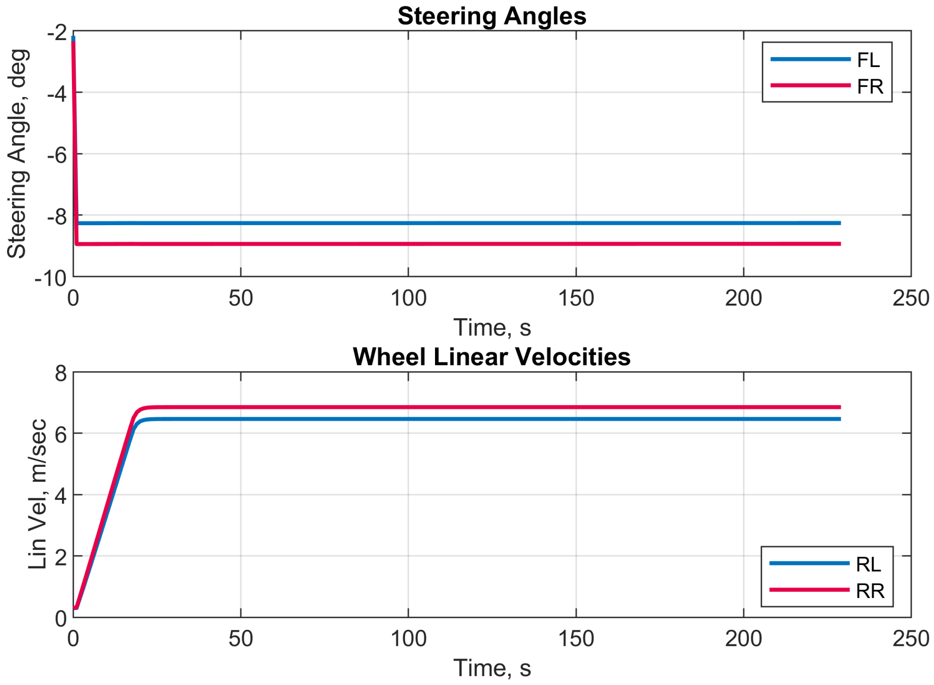

Figure 4.10: Scenario 1 - Vehicle Inputs

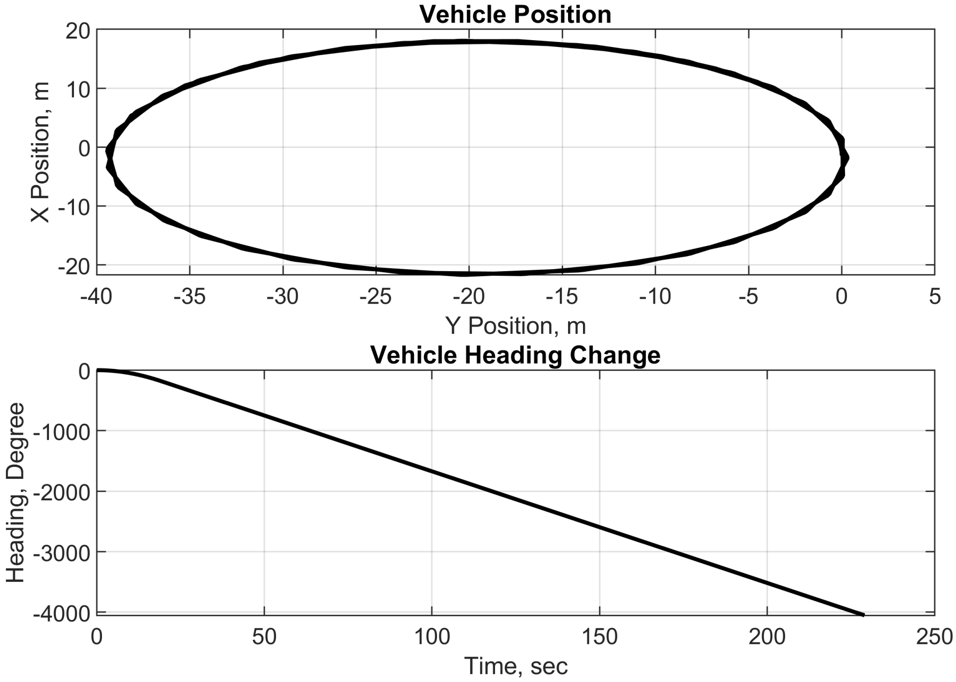

Figure 4.11: Scenario 1 - Vehicle Position

The contact forces generated for the control scenario are given in *Figures 4.12 to 4.15*. The presented figures include the ideal tire forces generated by the simulation, the result of the Pacejka tire model with ideal slip variables as an ideal model, and the estimation result for the tire forces for one simulation run.

It can be observed from the given figures that the initial velocity and attitude errors cause significant errors in the force estimations. However, it can also be observed that force estimations converge to the ideal Pacejka model within a short time, thanks to maneuvers.

As expected from a rear-drive vehicle, constant longitudinal contact forces are observed for the rear wheels (*Figures 4.14 & 4.15*), while small but opposing tire friction forces are observed for the front wheels (*Figures 4.12 & 4.13*). Moreover, constant lateral contact forces are observed for all wheels due to constant rotation at the constant velocity. Thanks to constant output forces, this scenario is utilized to tune the Ackermann drive controller gains and Pacejka tire model parameters.

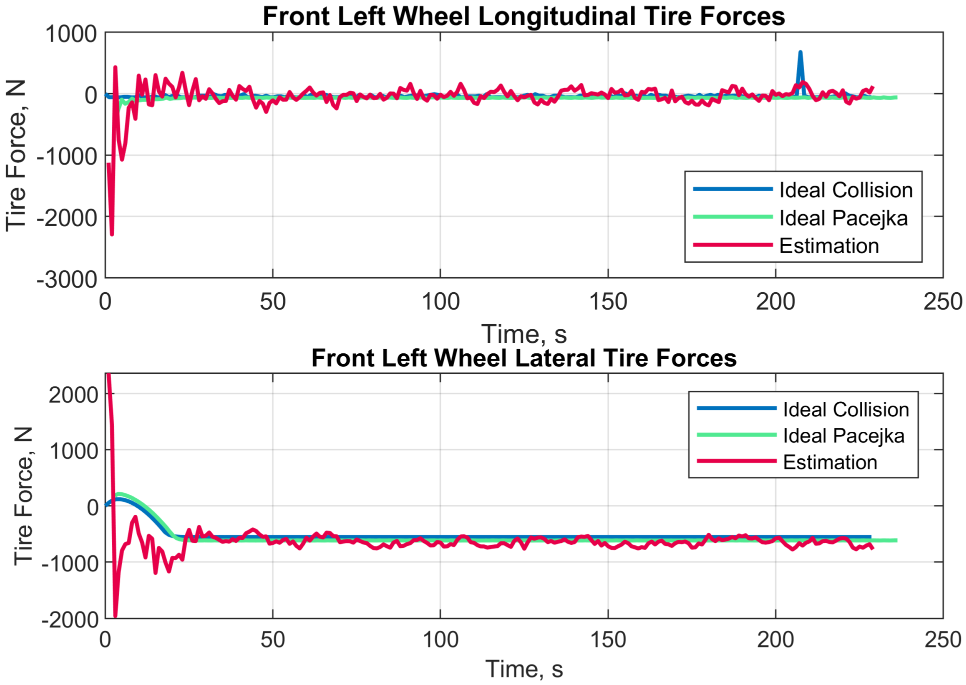

Figure 4.12: Scenario 1 - Front Left Wheel Tire Forces

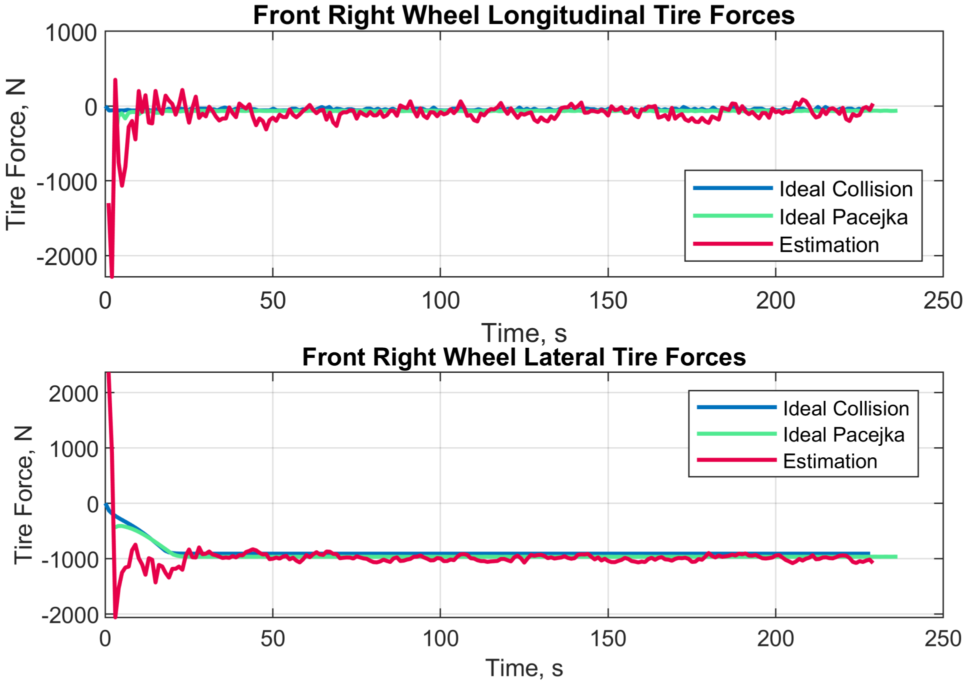

Figure 4.13: Scenario 1 - Front Right Wheel Tire Forces

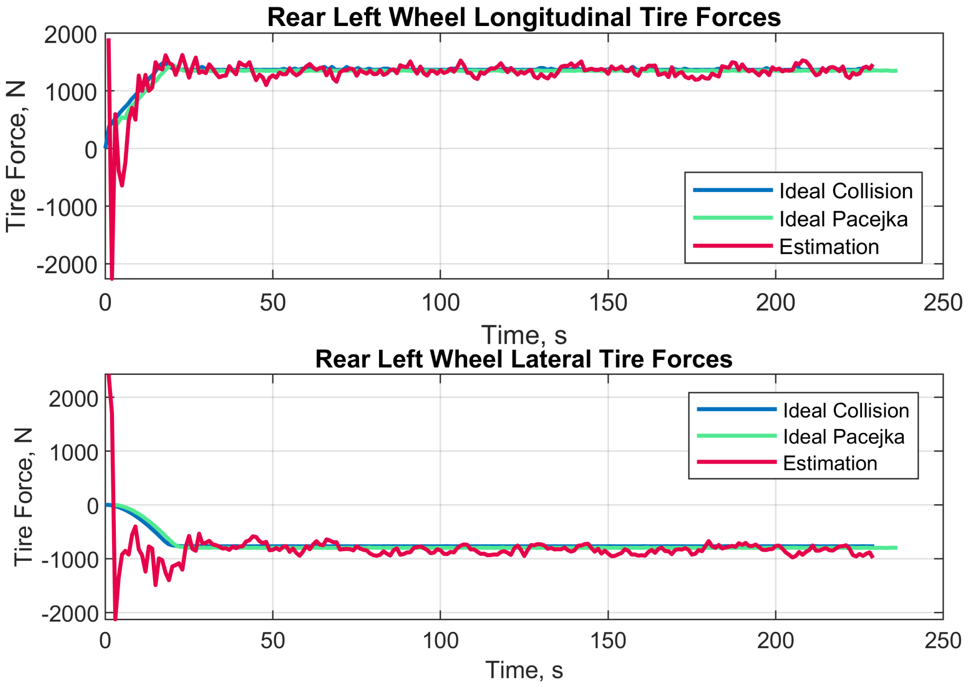

Figure 4.14: Scenario 1 - Rear Left Wheel Tire Forces

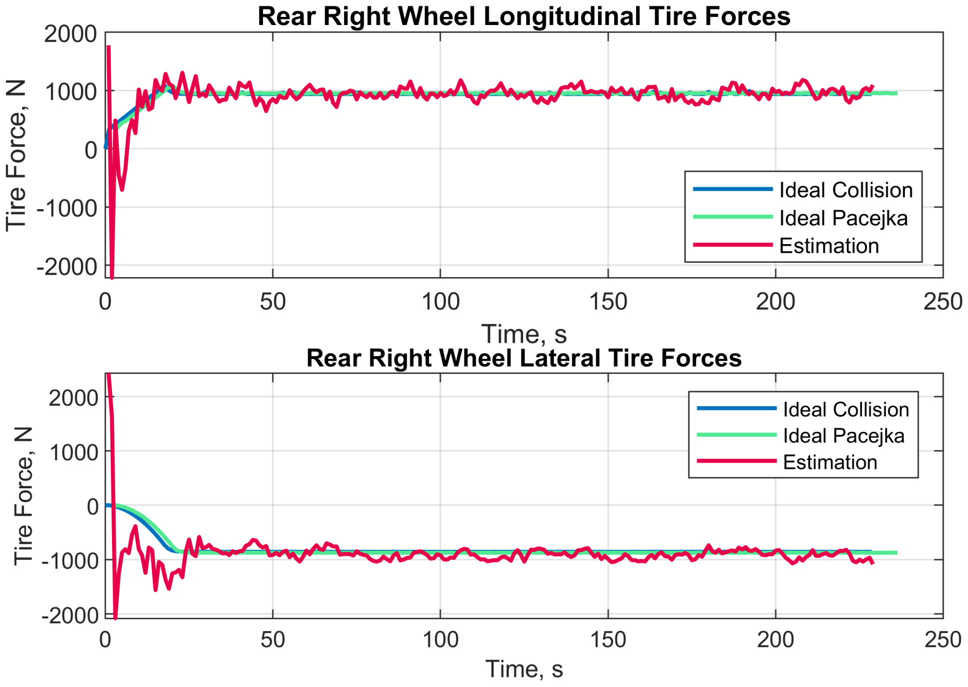

Figure 4.15: Scenario 1 - Rear Right Wheel Tire Forces

4.3.2 Scenario 2: The Performance Scenario

The performance scenario is created to evaluate the estimation performance of the algorithm under various conditions. Unlike other scenarios, the scenario includes cruising in different constant velocities. Moreover, unlike constant turn scenarios, the scenario includes both left and right turns. Additionally, this scenario includes Gaussian noise with $150\ N$ standard deviation on the normal load (F_z) used in the tire model parameters (*Equations 2.46 & 2.49*) unlike other scenarios. The vehicle inputs that are controlled by the Ackermann drive plug-in are given in *Figure 4.16*. The vehicle route and heading throughout the scenario is given in *Figure 4.17*.

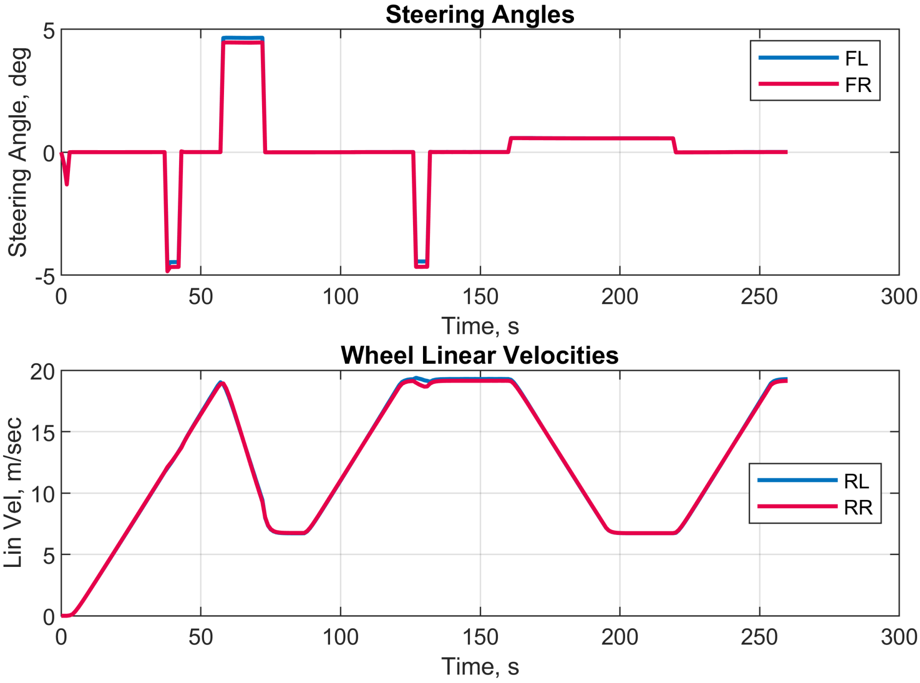

Figure 4.16: Scenario 2 - Vehicle Inputs

The contact forces generated for the performance scenario are given in *Figures 4.18 to 4.21*. The presented figures include the ideal tire forces generated by the simulation, the result of the Pacejka tire model with ideal slip variables as an ideal model, and the estimation result for the tire forces for one simulation run. Similar to first scenario, it can be observed from the given figures that the initial velocity and attitude errors cause significant errors in the force estimations. However, the force estimations converge to the ideal Pacejka model quickly, thanks to maneuvers.

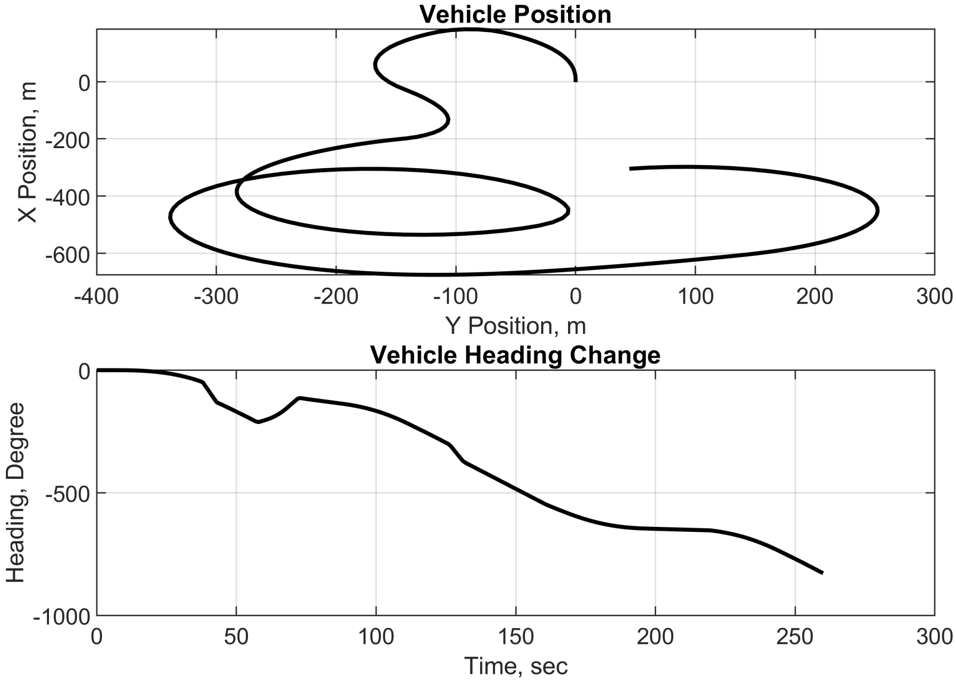

Figure 4.17: Scenario 2 - Vehicle Position

As expected, significant longitudinal tire contact forces are observed for the rear wheels (*Figures 4.14 & 4.15*), while small longitudinal contact forces are observed for the front wheels (*Figures 4.12 & 4.13*). Moreover, it can also be observed from *Figures 4.14 & 4.15* that the ideal longitudinal forces and ideal Pacejka model output diverge from each other at higher velocities. This modeling error causes the longitudinal force estimation errors to grow with higher velocities. Finally, for respected turn positive and negative lateral contact forces are observed at all wheels contacts.

4.3.3 Other Scenarios

A constant acceleration zero steering angle scenario is selected for the third scenario. The vehicle inputs that are controlled by the Ackermann drive plug-in are given in *Figure 4.22*. The vehicle route and heading throughout the scenario are given in *Figure 4.23*. This scenario is selected to investigate the lateral estimation performance without any maneuver. However, it can be observed from *Figure 4.23* that, unlike expectations, the vehicle slowly drifted to the left due to vehicle slip, and that move-

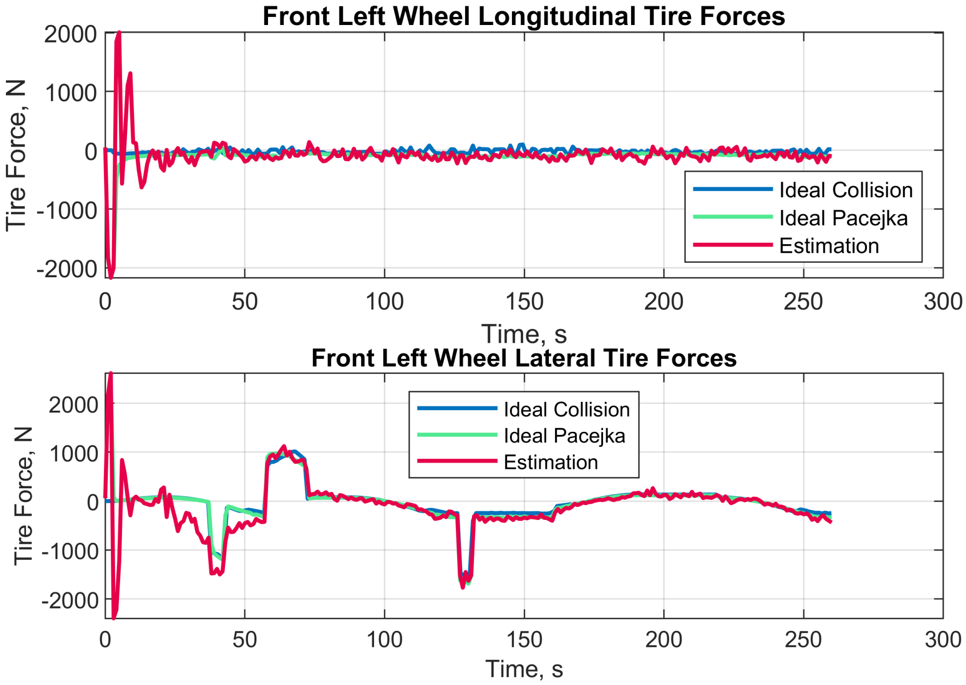

Figure 4.18: Scenario 2 - Front Left Wheel Tire Forces

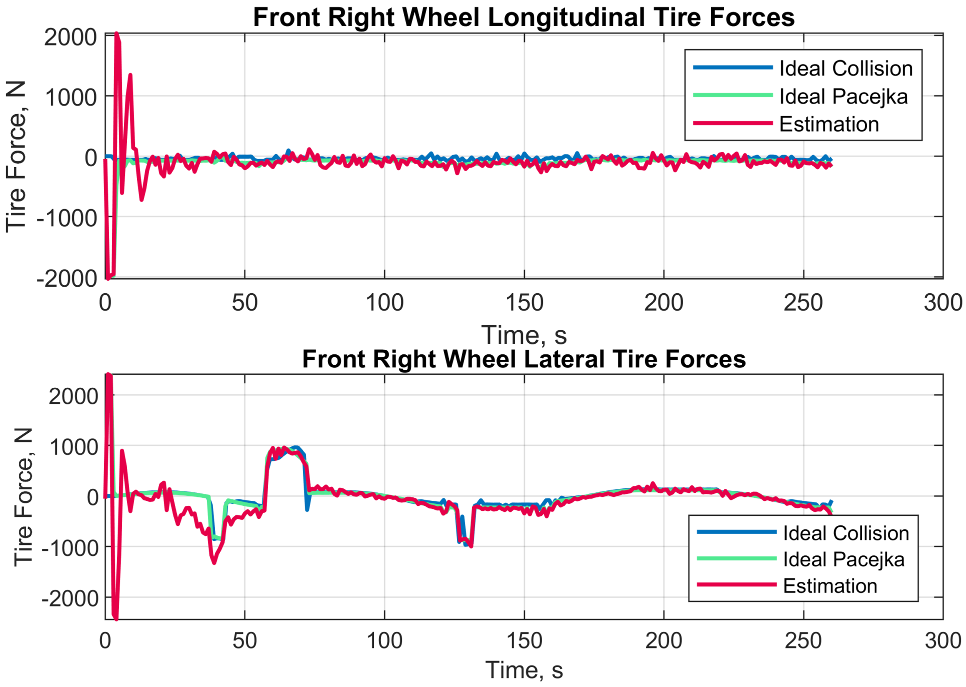

Figure 4.19: Scenario 2 - Front Right Wheel Tire Forces

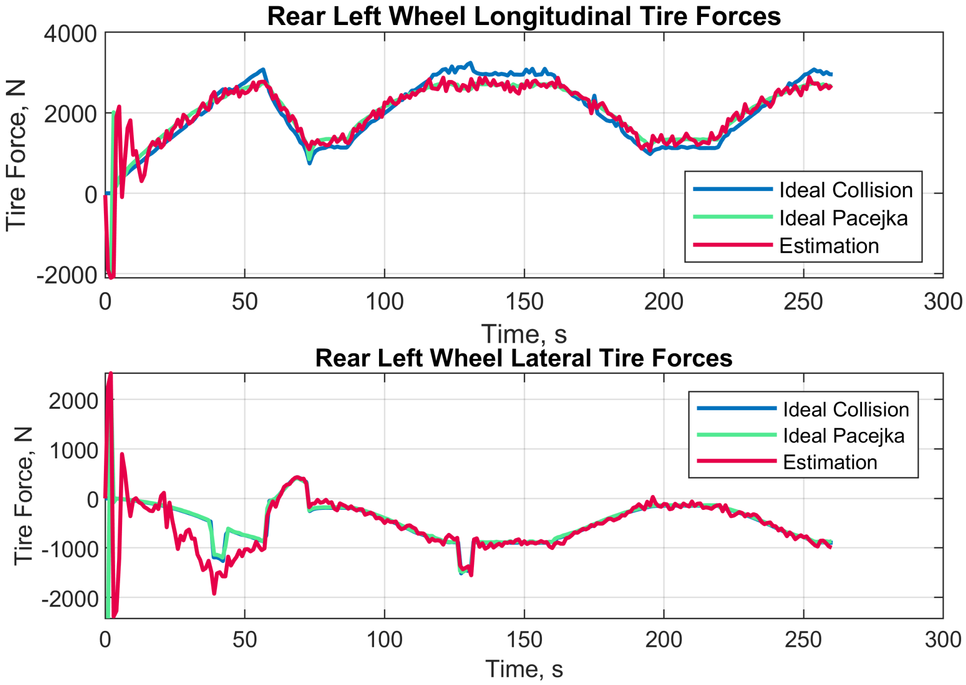

Figure 4.20: Scenario 2 - Rear Left Wheel Tire Forces

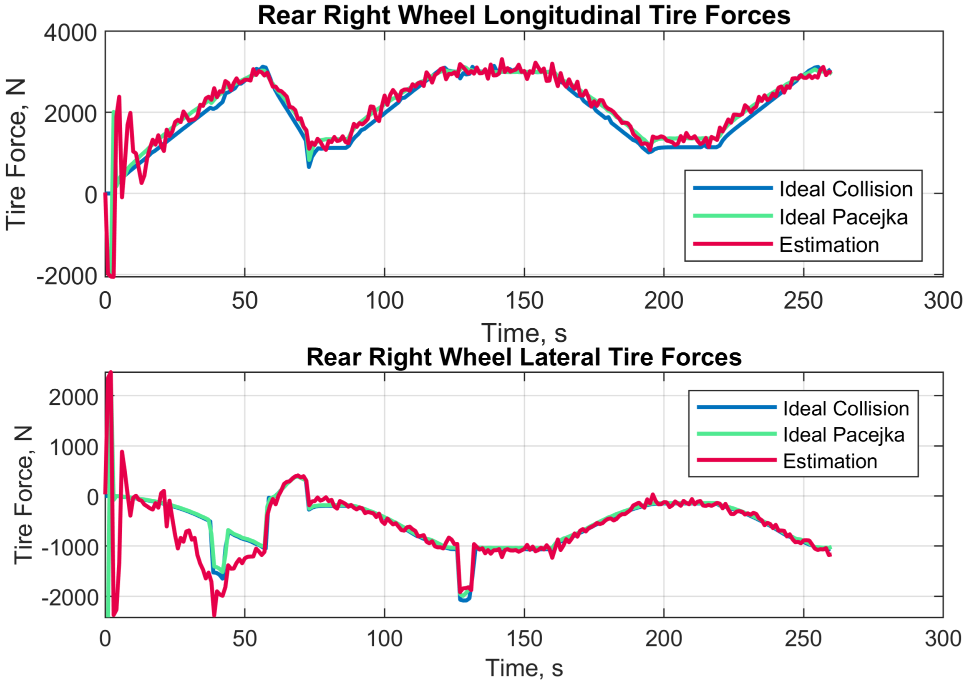

Figure 4.21: Scenario 2 - Rear Right Wheel Tire Forces

ment created a change in the yaw angle. Lateral and longitudinal tire contact forces occurred in the third scenario given in *Figures B.11 to B.14* of *Appendix B*.

For the forth scenario, a constant acceleration constant steering angle scenario is selected. The vehicle inputs that are controlled by the Ackermann drive plug-in are given in *Figure 4.24*. The vehicle route and heading throughout the scenario is given in *Figure 4.25*. Lateral and longitudinal tire contact forces occurred in the fourth scenario given in *Figures B.18 to B.21* of *Appendix B*.

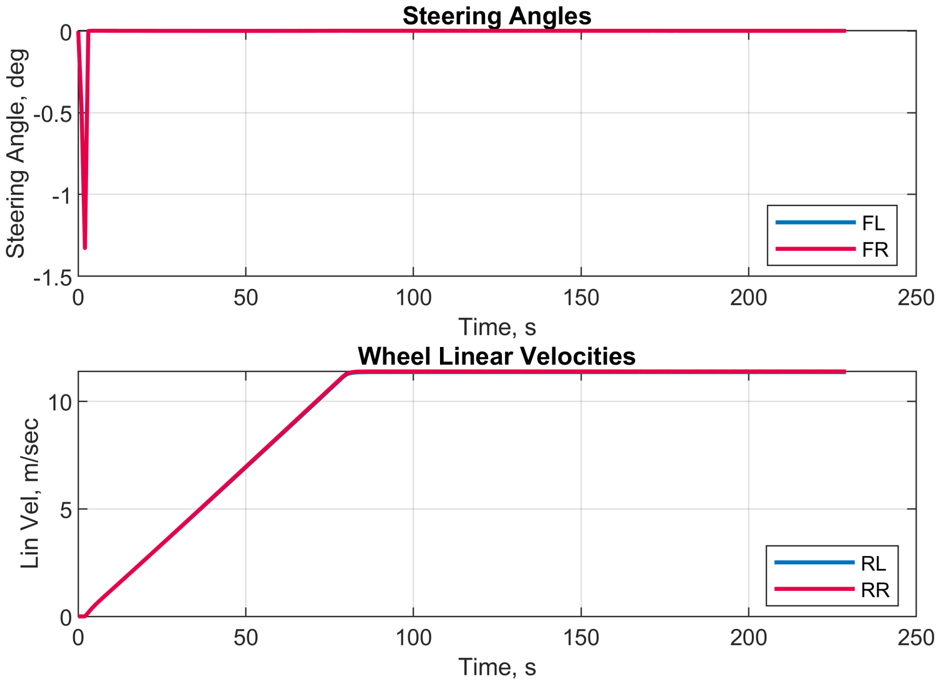

Figure 4.22: Scenario 3 - Vehicle Inputs

4.4 Estimation Quality

In order to be able to evaluate the quality of the estimations, driving scenarios are repeated 100 times as a Monte Carlo analysis. The navigation solution, slip, and force errors are computed using *Equation 4.1*. Where x_m represents the actual value of the state generated by the simulation in the m^{th} run and $\hat{x}_m$ represents the state estimation in the m^{th} run. Then, using the state errors, the mean of the absolute value of state errors for every simulation step are calculated using *Equation 4.2*. Finally,

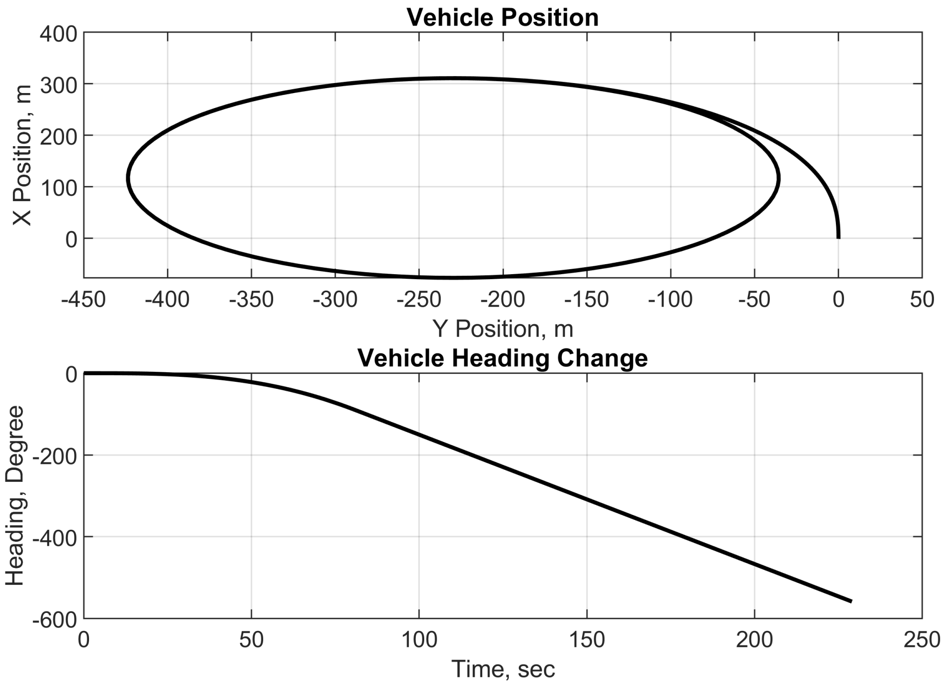

Figure 4.23: Scenario 3 - Vehicle Position

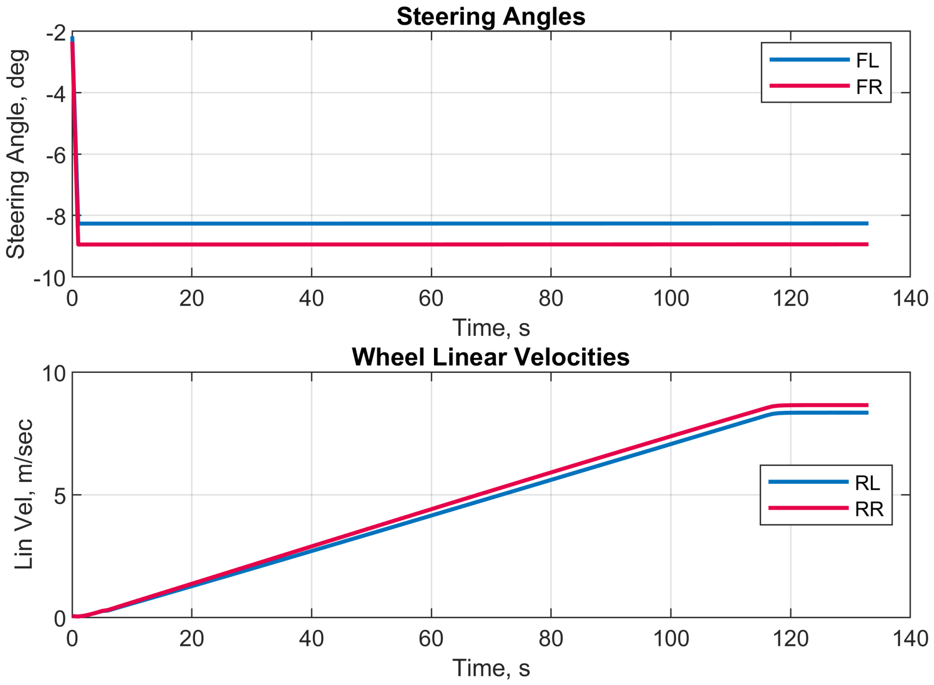

Figure 4.24: Scenario 4 - Vehicle Inputs

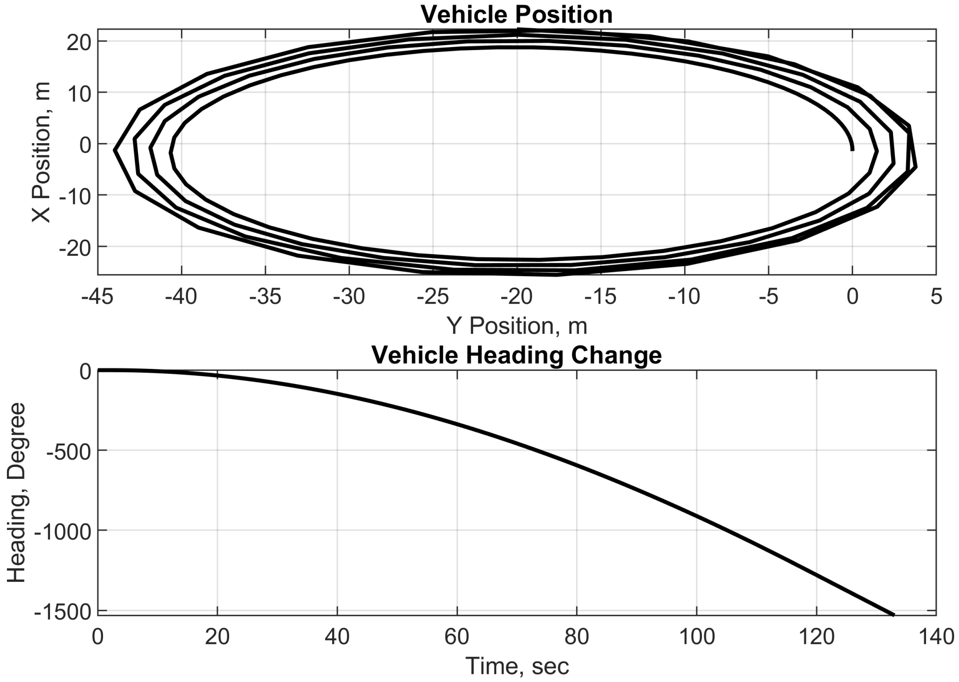

Figure 4.25: Scenario 4 - Vehicle Position

in order to create a evaluation metric, absolute state estimation error mean is defined using *Equation 4.3* where $t_{start} = 120s$ and $t_{end} = 220s$. In this section, absolute estimation error mean values are presented in the time domain. Then, absolute state estimation error means defined in *Equation 4.3* are presented in a table.

$$e_{x,m} = x_m - \hat{x}_m \tag{4.1}$$

$$\sigma_x(t_k) = mean(\mid e_{x,m}(t_k) \mid) \text{ ,where m} \in \text{1:100} \tag{4.2}$$

$$\sigma_x = mean(\ [\ \sigma_x(t_{start})\ :\ \sigma_x(t_{end})\ ]\ ) \tag{4.3}$$

4.4.1 Navigation Solution Estimation Quality

Using the methods defined by the *Equation 4.3*, the position, velocity, and attitude errors are computed for every simulation for every simulation step of Monte Carlo runs. In this section, the mean of absolute errors of 100 Monte Carlo runs for all scenarios are presented. Moreover, the average of absolute error means between 120th and 220th seconds of simulation for every scenario are presented in *Table 4.3*.

Scenario 1:

The mean of absolute errors of 100 Monte Carlo runs of control scenario for every simulation steps are given from *Figure 4.26* to *Figure 4.28*. As expected, the position solution carry the mean of the GPS position errors given in *Figure B.2*. Moreover, it can be seen from the figures that the states converge rapidly to their steady-state values thanks to maneuvers. Additionally, it can be noticed that the velocity errors became rapidly smaller than GPS velocity errors in *Figure B.3*.

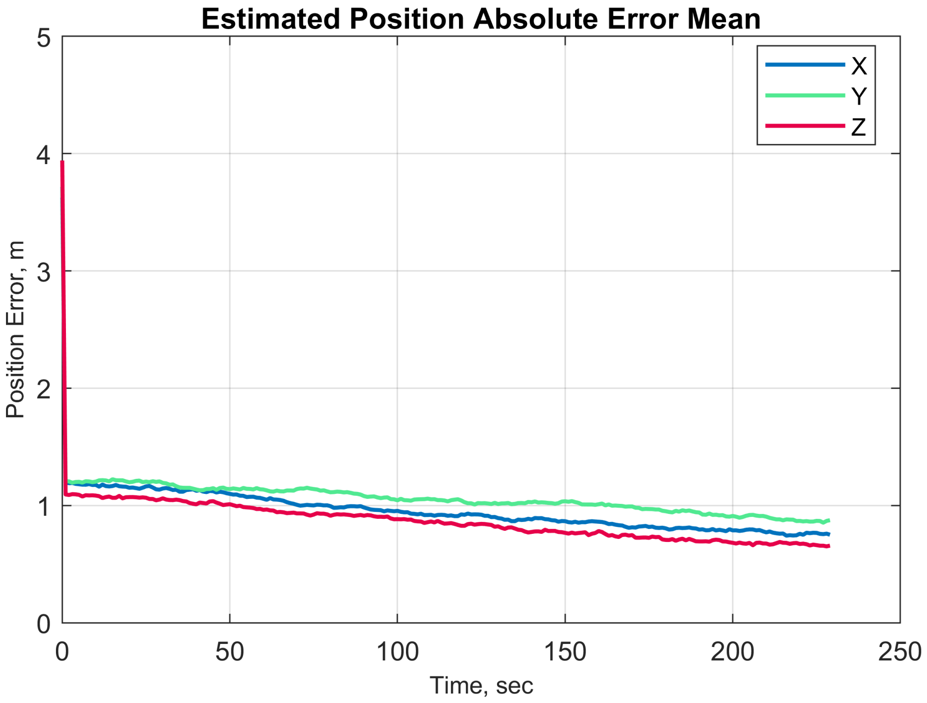

Figure 4.26: Scenario 1 - Vehicle Position Error

Scenario 2:

The mean of absolute errors of 100 Monte Carlo runs of performance scenario for every simulation steps are given from *Figure 4.30* to *Figure 4.32*. Similar to first scenario, the position solution carry the mean of GPS position errors given in *Figure B.2* and all states converge rapidly to their steady-state values thanks to maneuvers and

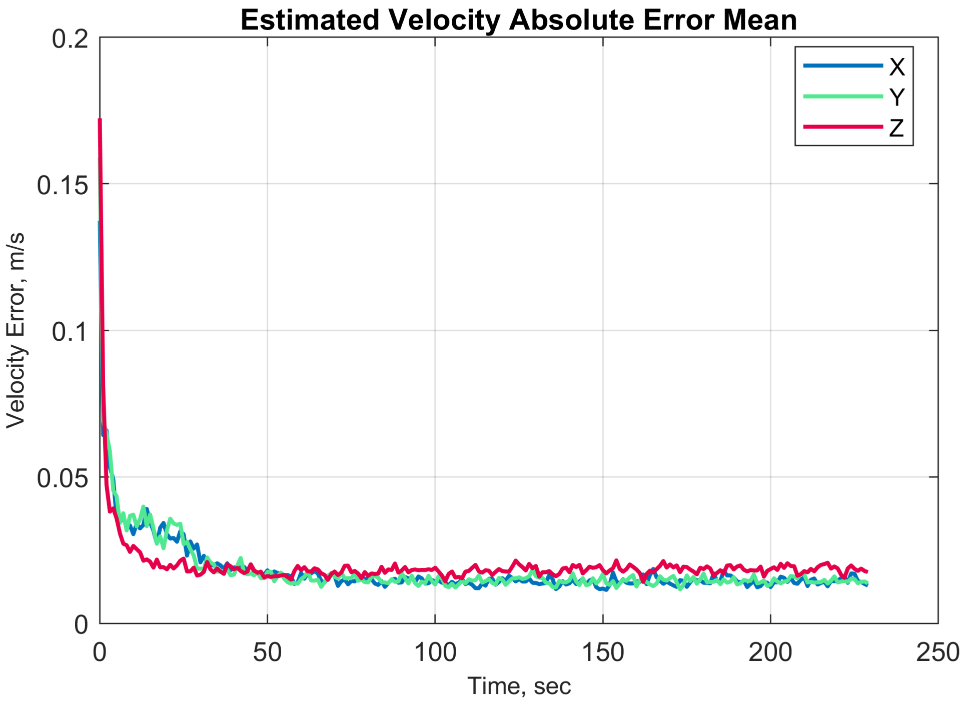

Figure 4.27: Scenario 1 - Vehicle Velocity Error

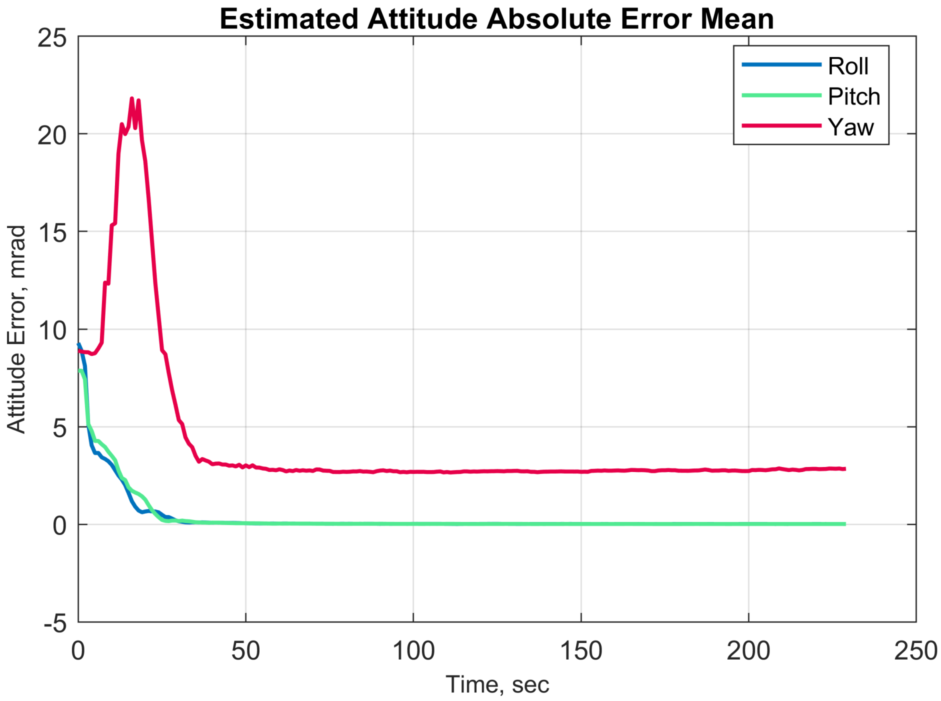

Figure 4.28: Scenario 1 - Vehicle Attitude Error

change of velocity. The actual and estimated position of the vehicle at the start and end of the simulation run is given in *Figure 4.29* to emphasise the position convergence.

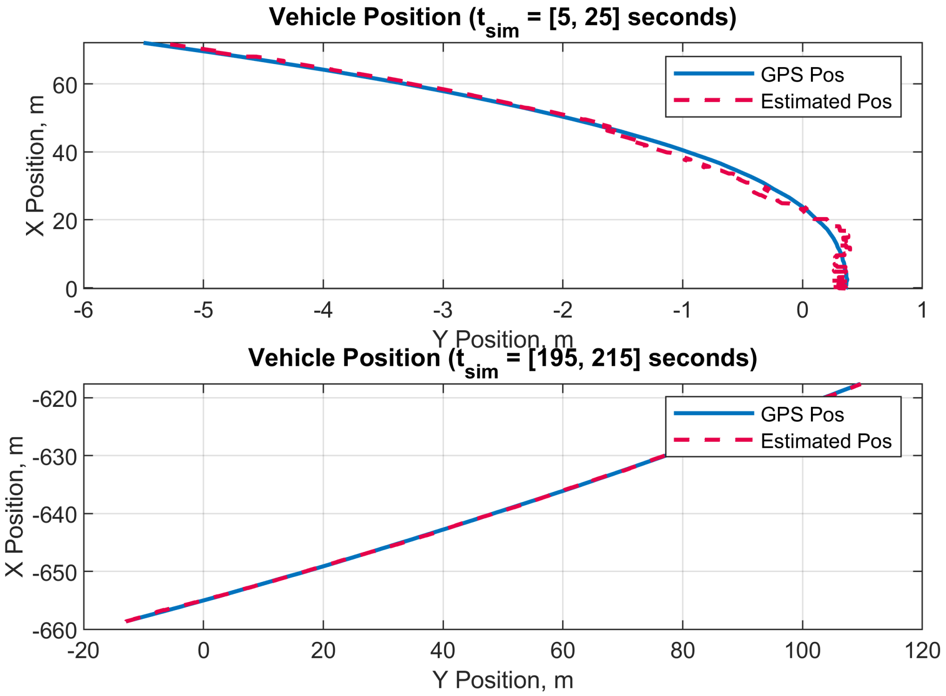

Figure 4.29: Scenario 2 - Vehicle Position Convergence

Table 4.3: Navigation Solution Estimation Error Performance

Estimated Vehicle State	**SCN 1**	**SCN 2**	**SCN 3**	**SCN 4**	**Unit**
Vehicle X-Position	0.833	0.952	0.946	0.953	m
Vehicle Y-Position	0.971	0.897	0.765	0.977	m
Vehicle Z-Position	0.741	0.964	0.846	0.907	m
Vehicle X-Velocity	0.014	0.024	0.019	0.014	m/s
Vehicle Y-Velocity	0.014	0.025	0.020	0.014	m/s
Vehicle Z-Velocity	0.018	0.018	0.017	0.017	m/s
Vehicle Roll Angle	0.016	0.228	0.189	0.020	mrad
Vehicle Pitch Angle	0.015	0.252	0.210	0.019	mrad
Vehicle Yaw Angle	5.402	2.705	6.796	4.330	mrad

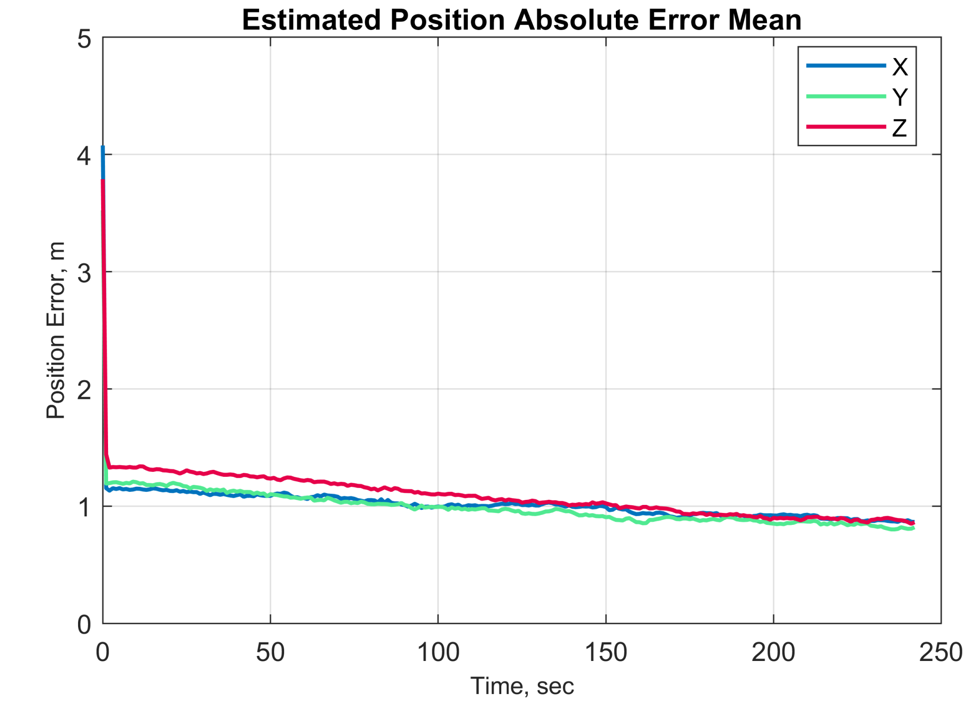

Figure 4.30: Scenario 2 - Vehicle Position Error

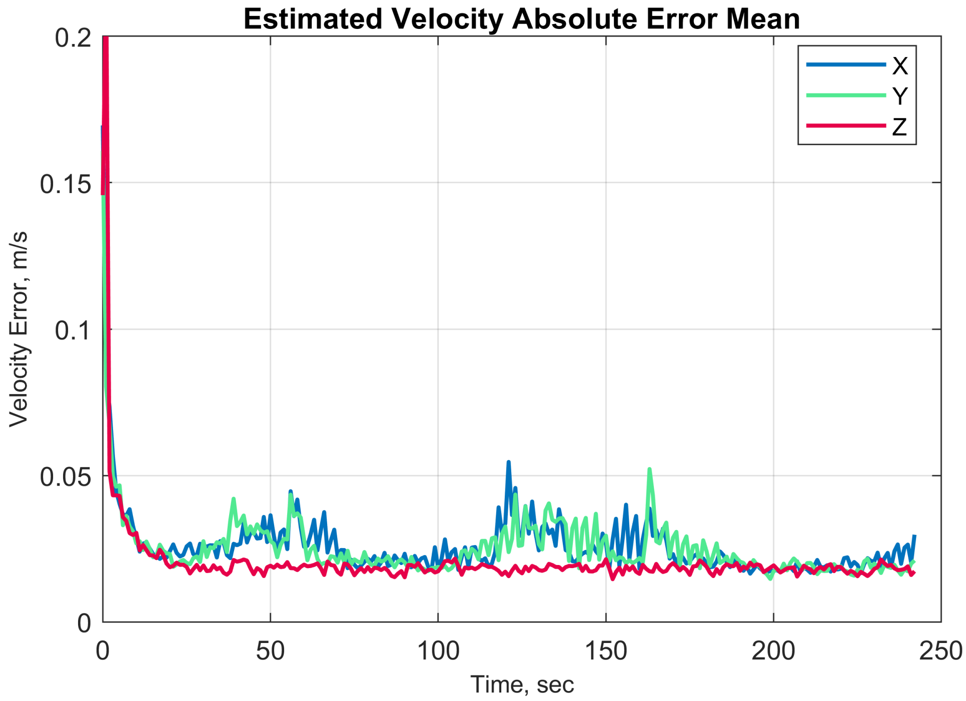

Figure 4.31: Scenario 2 - Vehicle Velocity Error

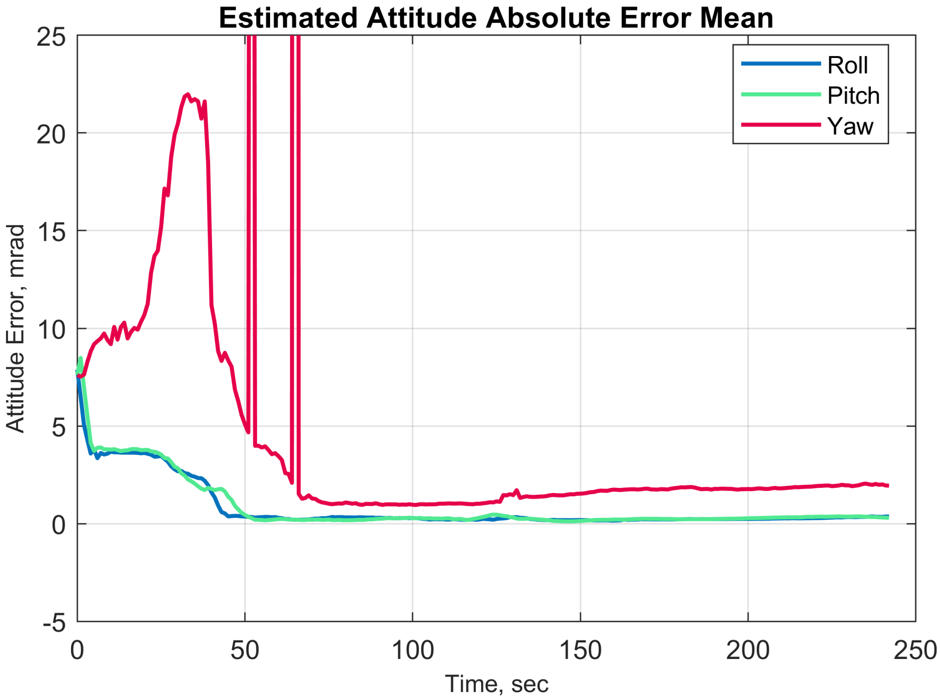

Figure 4.32: Scenario 2: Vehicle Attitude Error

4.4.2 Tire Slip Estimations Quality

Using the methods defined by the *Equation 4.3*, the longitudinal slip ratio and lateral side-slip angle errors are computed for every simulation step of the Monte Carlo runs. The mean of absolute errors of 100 Monte Carlo runs for every simulation steps for the control scenario are given in *Figures 4.33 & 4.34*. The mean of absolute tire slip estimation errors for the performance scenario are given in *Figures B.7 & B.8*.

The error average of absolute error means between 120th and 220th seconds of simulation for every scenario are presented in *Table 4.4*. As expected, the yaw angle error of the navigation solution estimation is reflected in side-slip errors and the largest side-slip angle error is observed during the third scenario that includes limited maneuvers and heading change.

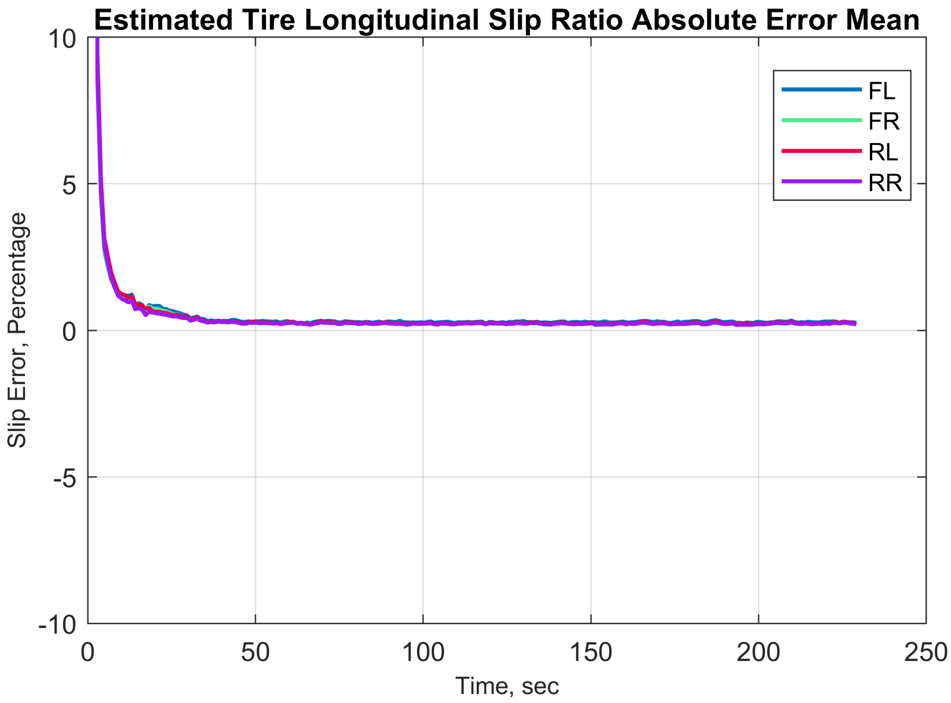

Figure 4.33: Scenario 1 - Longitudinal Tire Slips Errors

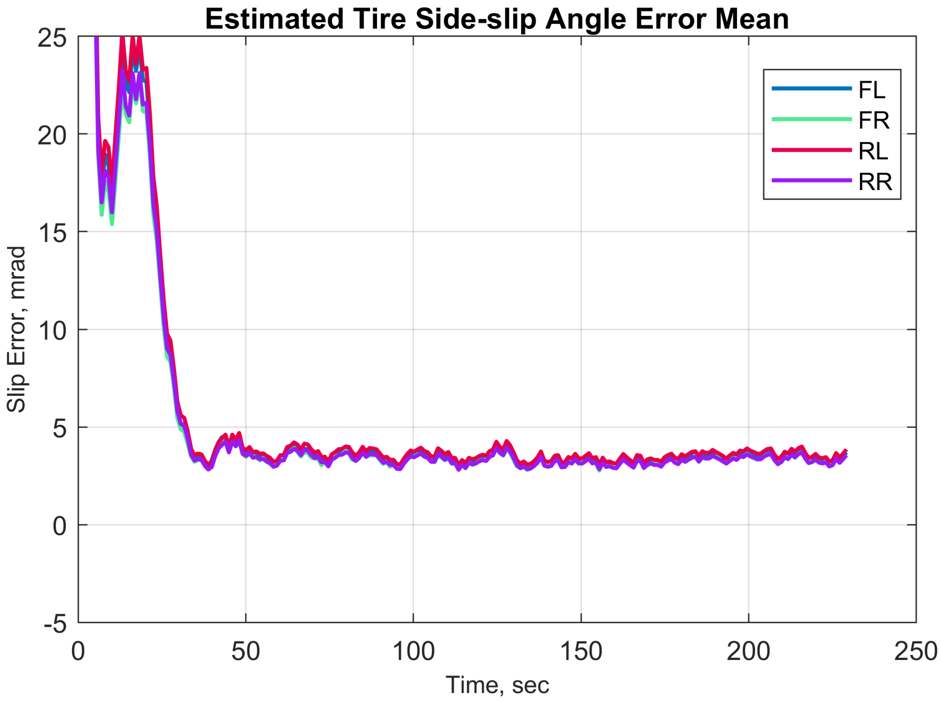

Figure 4.34: Scenario 1 - Lateral Tire Slips Errors

Table 4.4: Tire Slip Estimation Error Performance

Estimated Vehicle State	SCN 1	SCN 2	SCN 3	SCN 4	Unit
FL Longitudinal Slip Ratio	0.283	0.224	0.189	0.250	%
FR Longitudinal Slip Ratio	0.225	0.217	0.186	0.191	%
RL Longitudinal Slip Ratio	0.246	0.232	0.200	0.208	%
RR Longitudinal Slip Ratio	0.223	0.231	0.198	0.187	%
FL Lateral Side-Slip Angle	3.421	2.742	7.012	2.503	mrad
FR Lateral Side-Slip Angle	3.276	2.735	6.956	2.481	mrad
RL Lateral Side-Slip Angle	3.543	2.731	7.013	2.644	mrad
RR Lateral Side-Slip Angle	3.268	2.713	6.956	2.437	mrad

4.4.3 Tire Force Estimations Quality

Similar to navigation and tire slip estimation errors, the longitudinal and lateral tire force estimation errors are computed for every simulation step of the Monte Carlo runs. In this section, the mean of absolute errors of 100 Monte Carlo runs for control and performance scenarios are presented. The mean of absolute tire force estimation errors for other scenarios are given in *Appendix B*. Moreover, the average of absolute error means between 120th and 220th seconds of simulation for all scenarios are presented in *Table 4.5*. As expected, the errors in the side-slip error estimation is reflected in lateral force estimation and the largest lateral force estimation errors are observed during the third scenario that includes limited maneuvers and heading change. Additionally, largest longitudinal force estimation errors are observed during the performance scenario due to modelling errors at higher velocities

Scenario 1:

The mean value of absolute errors of 100 Monte Carlo runs for every simulation steps for the control scenario are given in *Figures 4.35 & 4.36*. It can be seen from the figures that the longitudinal and lateral force estimates converge rapidly to their steady-state values thanks to maneuvers.

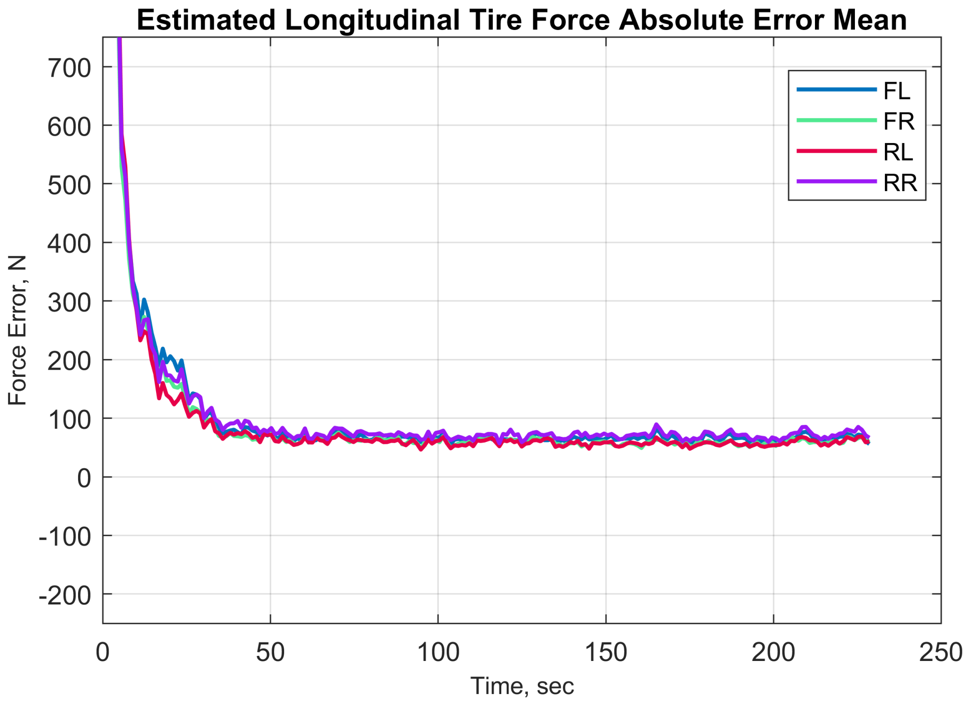

Figure 4.35: Scenario 1 - Longitudinal Tire Force Errors

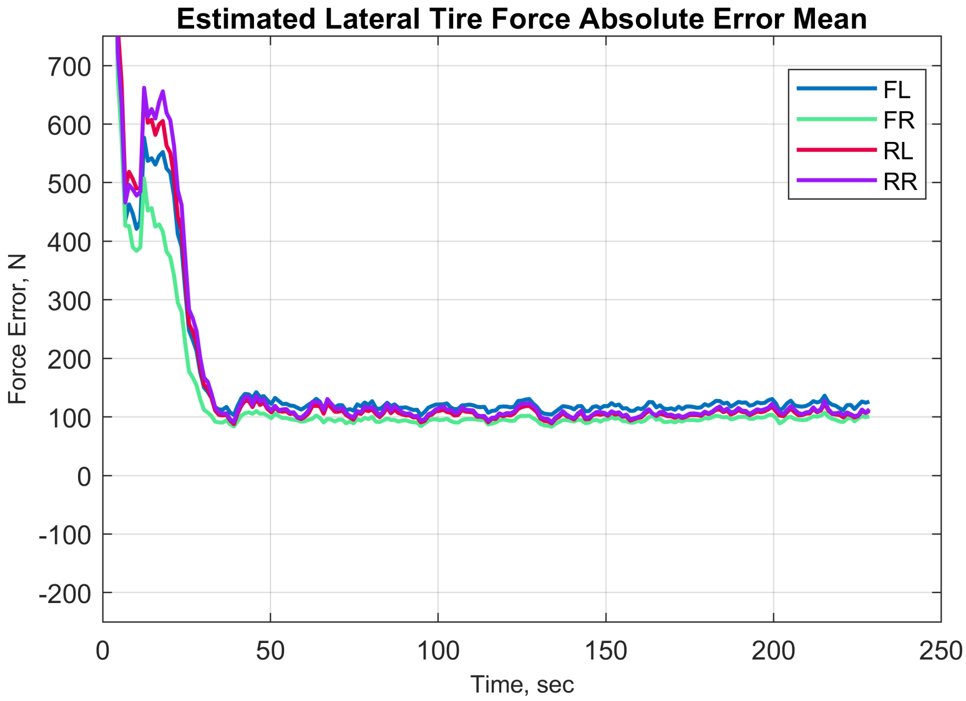

Figure 4.36: Scenario 1 - Lateral Tire Force Errors

Scenario 2:

The mean value of absolute errors of 100 Monte Carlo runs for every simulation steps for the performance scenario are given in *Figures 4.37 & 4.39*. It can be observed from *Figure 4.37* that the difference between ideal collision forces generated by the simulation and the Pacejka tire model output with ideal inputs diverge for higher linear velocities. Therefore, this modelling errors are observed as estimation errors for the longitudinal force estimations. It can be seen from the *Figure 4.38* that the longitudinal force estimations converge to ideal Pacejka model with minor bias. Additionally, the lateral force estimation errors with respect to ideal Pacejka model is given in *Figure B.9*.

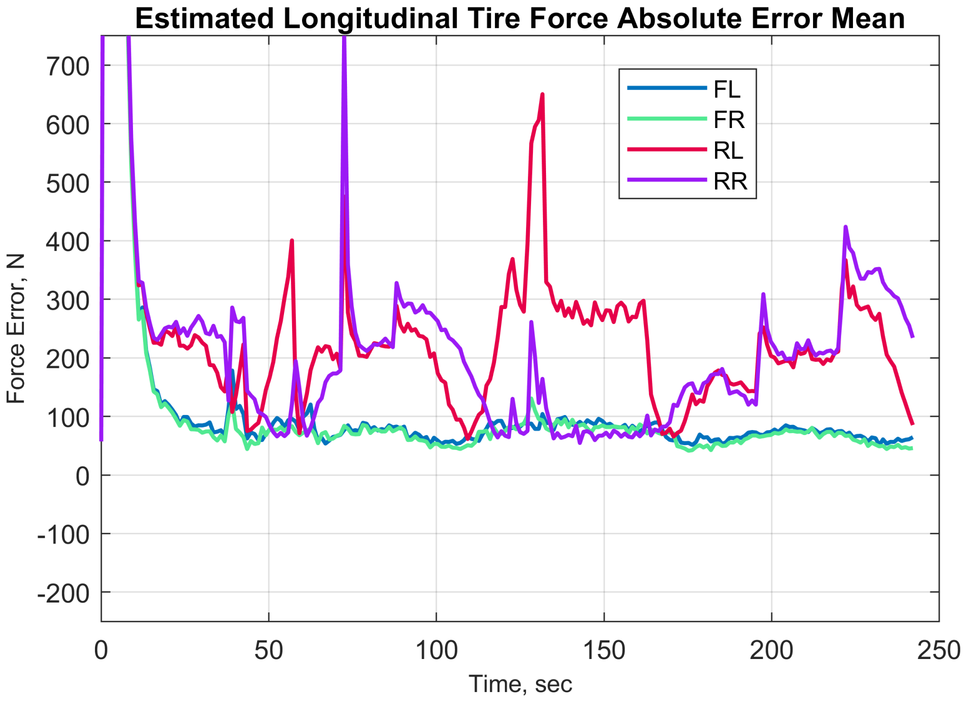

Figure 4.37: Scenario 2 - Longitudinal Tire Force Errors

4.5 Summary

In order to generalize the estimation performance, the Monte Carlo analyses were repeated for four driving scenarios presented in *Section 4.3*. In order to create a generalized performance metric, individual estimation absolute error means of similar

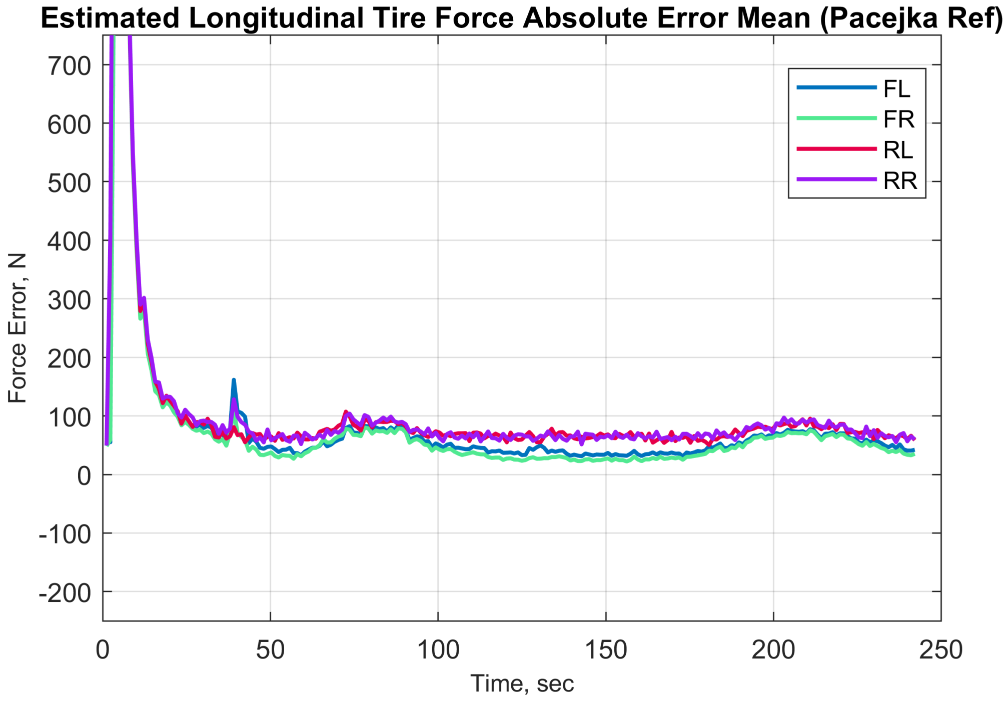

Figure 4.38: Scenario 2 - Ideal Pacejka Model Referenced Longitudinal Tire Force Errors

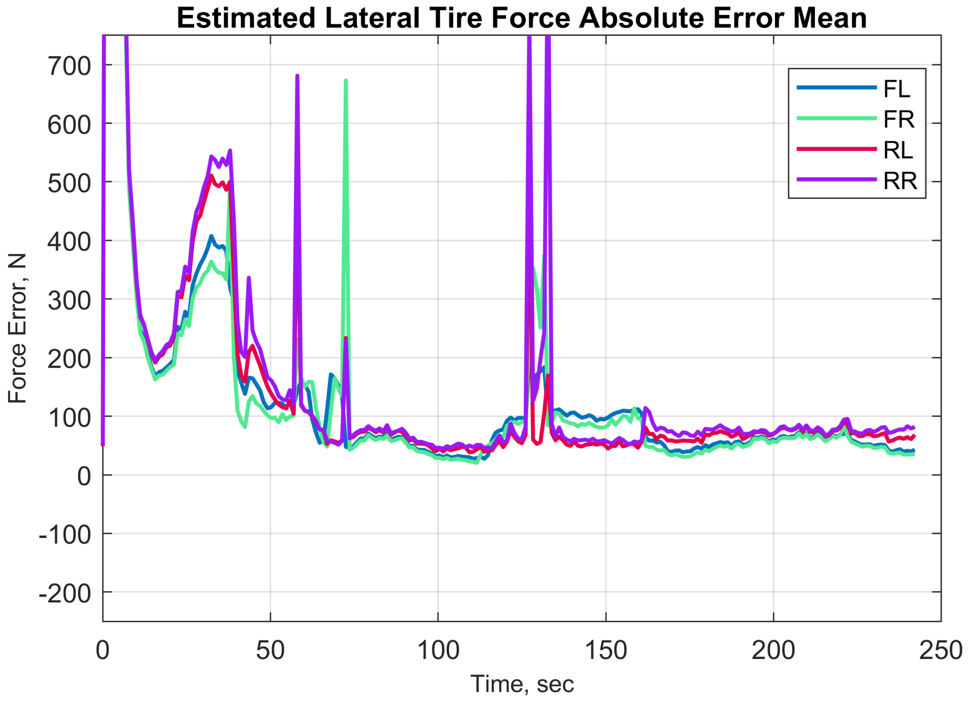

Figure 4.39: Scenario 2 - Lateral Tire Force Errors

Table 4.5: Tire Force Estimation Error Performance

Estimated Vehicle State	**SCN 1**	**SCN 2**	**SCN 3**	**SCN 4**	**Unit**
FL Longitudinal Tire Force	65.129	78.258	51.779	56.048	N
FR Longitudinal Tire Force	58.099	73.591	43.972	48.658	N
RL Longitudinal Tire Force	57.496	229.76	83.037	70.341	N
RR Longitudinal Tire Force	70.599	131.44	65.049	60.491	N
FL Lateral Tire Force	119.16	79.485	165.31	103.00	N
FR Lateral Tire Force	95.765	81.159	145.66	32.241	N
RL Lateral Tire Force	105.13	71.630	222.97	79.059	N
RR Lateral Tire Force	107.98	94.374	238.86	75.112	N

states are grouped. The generalized estimation absolute error mean is defined by *Equation 4.4* where the σ_x represents a vector that group similar states as in *Equation 4.5*. The results for 4 driving scenario is presented in *Table 4.6*.

$$\sigma_X = \mid \sigma_x \mid \tag{4.4}$$

$$f\sigma_x = \begin{bmatrix} \sigma_{x_x} \\ \sigma_{x_y} \\ \sigma_{x_z} \end{bmatrix} \quad \text{or} \quad \sigma_x = \begin{bmatrix} \sigma_{x_{FL}} \\ \sigma_{x_{FR}} \\ \sigma_{x_{RL}} \\ \sigma_{x_{RR}} \end{bmatrix} \tag{4.5}$$

By comparing first and third scenario, it can be concluded that the quantity of maneuvers or changes in the heading highly affects the observability of the attitude estimates. Moreover, the estimation quality of the attitude estimates directly affects the side-slip angle estimates. Similarly, the estimation quality of the side-slip angle directly affects the lateral force estimation quality. Therefore, it can be concluded that, in order to increase the observability of the lateral dynamics in a short period of time, a change in heading is necessary for the proposed algorithm. Similarly, the effect of longitudinal acceleration can be investigated by comparing the first and fourth scenarios. Although both scenarios have similar routes throughout the simulation, the estimation quality of the tire longitudinal and lateral dynamics improves thanks to linear acceleration in the fourth scenario. Finally, the longitudinal tire estimation quality decreases at higher velocities due to modelling errors.

Table 4.6: State Estimation Absolute Error Means For Different Scenarios

Estimated Vehicle State	**SCN 1**	**SCN 2**	**SCN 3**	**SCN 4**	**Unit**
Vehicle Position	1.4783	1.6246	1.4825	1.6389	m
Vehicle Velocity	0.0277	0.0397	0.0334	0.0269	m/s
Vehicle Attitude	5.4027	2.7268	6.8024	4.3310	mrad
Longitudinal Slip Ratio	0.4919	0.4531	0.3872	0.4220	%
Lateral Side-Slip Angle	6.7590	5.4613	13.969	5.0358	mrad
Longitudinal Tire Force	126.12	285.67	125.46	118.81	N
Lateral Tire Force	214.67	164.14	394.11	153.43	N

CHAPTER 5

CONCLUSION

Autonomous driving applications are a crucial task for the safety of vehicles. Implementing these critical applications to a wide range of vehicles with inexpensive sensors and estimating vehicle signals is a demanding task for researchers.

In this thesis, we proposed a Kalman filter-based estimator to observe the vehicle navigation solution, which consists of the position, velocity, and attitude of the vehicle. The proposed Kalman filter is used to combine the high-frequency, high-bandwidth erroneous Inertial Navigation Solution (INS) produced from IMU measurements and low-frequency, low-bandwidth GPS position and velocity solution, which both have bounded errors. Then, the method to observe necessary tire states is proposed using the navigation solution, wheel encoder measurements, and steering angle sensor measurement.

In addition, a simulation environment is built for this study using the Gazebo Classic simulator and ROS-2 middleware. Then a vehicle model is built to create the necessary sensor data for the algorithms. Then the proposed algorithms were implemented as ROS-2 nodes and attached to the vehicle model. Different driving scenarios are created in order to evaluate the estimation quality. The simulations are repeated multiple times for different scenarios as a part of the Monte Carlo analysis. The results showed that the proposed algorithm could estimate the vehicle's position within a 5-meter error, the velocity within the 0.5 cm/s error, and the attitude within the 0.5-degree error. Moreover, the proposed algorithm can produce lateral and longitudinal tire force estimates within 200-N error if there is sufficient change in heading to improve the quality of attitude estimates.

Lastly, the performance of the proposed analysis is evaluated using different driving scenarios. The results showed that the scenario of the driving scenario directly affects the quality of the estimation as expected. It has been observed that the estimation quality of the vehicle attitude estimation increases as the vehicle is exposed to more maneuvers or heading changes. Therefore, with more maneuver, the estimation quality of side-slip angle estimation and lateral contact force estimation increases since they are heavily dependent on the heading of the vehicle. Moreover, it has also been concluded that the observability of the vehicle also increases not just by maneuvers but also with the vehicle's acceleration.

This results demonstrate that the proposed algorithms provide sufficient enough estimations for the required vehicle signals. However, the estimation quality can be improved by future work. The proposed algorithms can be improved by considering the wheel encoder measurements as a measurement to the EKF. The study can also be improved by building a filter that include tire contact force estimation errors as a error state variables. Additionally, the simulations conducted during the study can further be improved by modelling physical disturbances to the wheel joints to simulate the disturbances on the lever arm between IMU and wheel.

REFERENCES

[1] "Ackermann turning geometry illustration." https://commons.wikimedia.org/wiki/File:Ackermann_turning.svg. Accessed: 2022-07-10.

[2] R. N. Jazar, *Vehicle Dynamics Theory and Application*. Springer, 2nd ed., 2014.

[3] Y. Yang and T. Azumi, "Exploring real-time executor on ros 2," in *2020 IEEE International Conference on Embedded Software and Systems, ICESS 2020*, Institute of Electrical and Electronics Engineers Inc., 12 2020.

[4] O. Robotics, "Understanding topics." https://docs.ros.org/en/foxy/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Topics/Understanding-ROS2-Topics.html. Accessed: 2021-03-26.

[5] Z. B. Rivera, M. C. D. Simone, and D. Guida, "Unmanned ground vehicle modelling in gazebo/ros-based environments," *Machines*, vol. 7, 7 2019.

[6] "Newly released estimates show traffic fatalities reached a 16-year high in 2021." https://www.nhtsa.gov/press-releases/early-estimate-2021-traffic-fatalities. Accessed: 2022-07-18.

[7] "Critical reasons for crashes investigated in the national motor vehicle crash causation survey." https://crashstats.nhtsa.dot.gov/Api/Public/Publication/812115. Accessed: 2022-07-18.

[8] M. Papis, T. Dziewoński, and M. Matyjewski, "Preliminary assessment of the advanced driver assistance systems efficiency from the safety point of view," pp. 371–375, 06 2017.

[9] C. Jain, R. Abhishek, and A. Dixit, "Linear control technique for anti-lock brak-

ing system," *International Journal of Engineering Research and Applications*, vol. 4, no. 8, pp. 104–108, 2014.

[10] L. Chen, Z. Li, J. Yang, and Y. Song, "Lateral stability control of four-wheel-drive electric vehicle based on coordinated control of torque distribution and esp differential braking," *Actuators*, vol. 10, no. 6, 2021.

[11] C. K. Song, M. Uchanski, and J. K. Hedrick, "Vehicle speed estimation using accelerometer and wheel speed measurements," in *SAE Technical Papers*, SAE International, 2002.

[12] J. J. Oh and S. B. Choi, "Vehicle velocity observer design using 6-d imu and multiple-observer approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, pp. 1865–1879, 2012.

[13] S. L. Miller, B. Youngberg, A. Millie, P. Schweizer, and C. Gerdes, "Calculating longitudinal wheel slip and tire parameters using gps velocity," in *Proceedings of the American Control Conference*, vol. 3, pp. 1800–1805, Institute of Electrical and Electronics Engineers Inc., 2001.

[14] D. M. Bevly, J. Ryu, and J. C. Gerdes, "Integrating ins sensors with gps measurements for continuous estimation of vehicle sideslip, roll, and tire cornering stiffness," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, pp. 483–493, 12 2006.

[15] M. Hrgetic, J. Deur, V. Ivanovic, and E. Tseng, "Vehicle sideslip angle ekf estimator based on nonlinear vehicle dynamics model and stochastic tire forces modeling," *SAE Int. J. Passeng. Cars-Mech. Syst*, vol. 7, 2014.

[16] J. Dakhlallah, S. Glaser, S. Mammar, and Y. Sebsadji, "Tire-road forces estimation using extended kalman filter and sideslip angle evaluation," in *Proceedings of the American Control Conference*, pp. 4597–4602, Institute of Electrical and Electronics Engineers Inc., 2008.

[17] M. Doumiati, A. C. Victorino, A. Charara, and D. Lechner, "Onboard real-time estimation of vehicle lateral tire-road forces and sideslip angle," *IEEE/ASME Transactions on Mechatronics*, vol. 16, pp. 601–614, 8 2011.

[18] D. Moustapha, V. Alessandro, C. Ali, and L. Daniel, “A method to estimate the lateral tire force and the sideslip angle of a vehicle: Experimental validation,” in *American Control Conference*, 2010.

[19] M. Doumiati, G. Baffet, D. Lechner, A. Victorino, and A. Charara, “Embedded estimation of the tire/road forces and validation in a laboratory vehicle,” tech. rep., 2008.

[20] E. Joa, K. Yi, and Y. Hyun, “Estimation of the tire slip angle under various road conditions without tire–road information for vehicle stability control,” *Control Engineering Practice*, vol. 86, pp. 129–143, 5 2019.

[21] Y. W. Liao and F. Borrelli, “An adaptive approach to real-time estimation of vehicle sideslip, road bank angles, and sensor bias,” *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 7443–7454, 8 2019.

[22] F. Cheli, E. Sabbioni, M. Pesce, and S. Melzi, “A methodology for vehicle sideslip angle identification: Comparison with experimental data,” *User Modeling and User-Adapted Interaction*, vol. 45, pp. 549–563, 2007.

[23] S. H. You, J. O. Hahn, and H. Lee, “New adaptive approaches to real-time estimation of vehicle sideslip angle,” *Control Engineering Practice*, vol. 17, pp. 1367–1379, 12 2009.

[24] K. Jiang, A. Pavelescu, A. C. Victorino, A. Charara, and A. Victorino, “Estimation of vehicle’s vertical and lateral tire forces considering road angle and road irregularity,” in *IEEE Conference on Intelligent Transportation Systems*, pp. 342–347, 2014.

[25] C. Reyes, O. Garcia, P. S. Meirelles, and J. V. Ferreira, “Estimation of longitudinal and lateral tire forces in a commercial vehicle,” in *2015 Workshop on Engineering Applications - International Congress on Engineering, WEA 2015*, Institute of Electrical and Electronics Engineers Inc., 12 2015.

[26] C. D. R. Bautista, P. S. Meirelles, and O. G. Bedoya, “Sideslip angle estimation for ground vehicles,” in *Proceedings of the 23rd ABCM International Congress of Mechanical Engineering*, ABCM Brazilian Society of Mechanical Sciences and Engineering, 2015.

[27] H. Heidfeld, M. Schünemann, and R. Kasper, "Ukf-based state and tire slip estimation for a 4wd electric vehicle," *Vehicle System Dynamics*, vol. 58, pp. 1479–1496, 10 2020.

[28] H. Harry, H. J. Karl, R. Stephan, and G. Frank, "Tire force estimation for a passenger vehicle with the unscentedkalman filter," in *IEEE Intelligent Vehicles Symposium*, IEEEE, 8 2014.

[29] A. C. Hazlett, J. L. Crassidis, D. P. Fuglewicz, and P. Miller, "Differential wheel speed sensor integration with gps/ins for land vehicle navigation," in *AIAA Guidance, Navigation, and Control Conference*, 8 2011.

[30] K. S. Han, E. Lee, and S. Choi, "Estimation of the maximum lateral tire-road friction coefficient using the 6-dof sensor," in *ICCAS 2015 - 2015 15th International Conference on Control, Automation and Systems, Proceedings*, pp. 1734–1738, Institute of Electrical and Electronics Engineers Inc., 12 2015.

[31] C. Sierra, E. Tseng, A. Jain, and H. Peng, "Cornering stiffness estimation based on vehicle lateral dynamics," *Vehicle System Dynamics*, vol. 44, pp. 24–38, 2006.

[32] B. Ozkan, D. Margolis, and M. Pengov, "The controller output observer: Estimation of vehicle tire cornering and normal forces," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 130, pp. 0610021–06100210, 11 2008.

[33] S. Yazid, G. Sebastien, M. Said, and D. Jamil, "Road slope and vehicle dynamics estimation," in *American Control Conference*, American Automatic Control Council, 6 2008.

[34] X. Liao, Q. Huang, D. Sun, W. Liu, and W. Han, "Real-time road slope estimation based on adaptive extended kalman filter algorithm with in-vehicle data," in *29th Chinese Control And Decision Conference (CCDC)*, IEEE, 2017.

[35] G. F. Trommer, "Sensor fusion for land vehicle slope estimation," in *DGON Inertial Sensors and Systems (ISS)*, p. 22, IEEE, 9 2016.

[36] W. K. J. and S. James, *Springer Handbook of Robotics*. Springer International Publishing, 2nd ed., 2016.

[37] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House Publishers, 2nd ed., 2013.

[38] J. A. Farrell, *Aided Navigation : GPS With High Rate Sensors*. McGraw Hill, 1st ed., 2008.

[39] W. S. Flenniken, J. H. Wall, and D. M. Bevly, "Characterization of various imu error sources and the effect on navigation performance," in *18th International Technical Meeting of the Satellite Division of The Institute of Navigation*, pp. 967 – 978, ION, 9 2005.

[40] M. J. L., K. L. G., and B. J. N., *DYNAMICS (Engineering Mechanics)*. John Wiley & Sons, Inc, 8th ed., 2015.

[41] A. Rudolph, "Observations on ackermann's patent movable axles for four wheeled carriages," tech. rep., University of Michigan, 1819.

[42] E. Kaltenegger, "Physical and graphical simulation of an ackermann steered vehicle," tech. rep., TU Wien, 9 2016.

[43] R. Rajesh, *Vehicle Dynamics and Control (Mechanical Engineering Series)*. Springer Science+Business Medi, 2nd ed., 2012.

[44] H. B. Pacejka, *Tire and Vehicle Dynamics*. Elsevier, 3rd ed., 2012.

[45] H. PACEJKA and R. SHARP, "Shear force development by pneumatic tyres in steady state conditions: A review of modelling aspects," *Vehicle System Dynamics*, vol. 20, no. 3-4, pp. 121–175, 1991.

[46] B. R. Grover and H. P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*. Wiley, 4th ed., 2012.

[47] R. E. Kalman, "A new approach to linear filtering and prediction problems," *ASME*, vol. 82, pp. 35–45, 3 1960.

[48] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series, with engineering applications*. The MIT Press, 1st ed., 3 1949.

[49] S. N., "Lecture notes, fall 2009: 3 - the wiener filter," tech. rep., Israel Institute of Technology, Department of Electrical Engineering, 9 2009.

[50] O. A. V and S. R. W., *Discrete-Time Signal Processing*. Pearson New International Edition, 3rd ed., 8 2013.

[51] K. Eyice, "Reducing computational demand of multi-state constraint kalman filter in visual-inertial odometry applications," Master's thesis, Middle East Technical University, 2019.

[52] S. Campbell, N. O'Mahony, L. Krpalcova, D. Riordan, J. Walsh, A. Murphy, and C. Ryan, "Sensor technology in autonomous vehicles : A review," in *2018 29th Irish Signals and Systems Conference (ISSC)*, pp. 1–4, 2018.

[53] D. Česenek, "Inertial measurement unit modeling," Master's thesis, Czech Technical University, 5 2019.

[54] E. Electronic, "Epson inertial measurement unit (imu)." `https://www.epson-electronics.de/electronics/cms/index/465`. Accessed: 2021-09-26.

[55] B. M. Solutions, "Inertial measurement unit)." `https://www.bosch-mobility-solutions.com/en/solutions/sensors/inertial-measurement-unit/`. Accessed: 2021-09-26.

[56] H. AB, "Automotive positioning / precise positioning systems / compact packages)." `https://hexagonpositioning.com/autonomous-x/automotive-positioning/precise-positioning-systems/compact-packages`. Accessed: 2021-09-26.

[57] VectorNav, "Vectornav vn-110 imu/ahrs)." `https://www.vectornav.com/resources/datasheets/vn-110-imu-ahrs`. Accessed: 2021-09-26.

[58] M. Pratap and E. Per, *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2nd ed., 2006.

[59] "Other global navigation satellite systems (gnss)." `https://www.gps.gov/systems/gnss/`. Accessed: 2022-06-16.

[60] "Other global navigation satellite systems (gnss)." `https://www.`

`euspa.europa.eu/european-space/eu-space-programme/what-gnss/`. Accessed: 2022-06-16.

[61] T. S. N. O. (AJM-32), “Global positioning system standard positioning service performance analysis report,” tech. rep., Federal Aviation Administration (FAA), 1 2021.

[62] D. H. Titterton and J. L. Weston, *Strapdown inertial navigation technology*. Institution of Electrical Engineers & American Institute of Aeronautics and Astronautics., 2nd ed., 2004.

[63] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint kalman filter for vision-aided inertial navigation,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3565–3572, 2007.

[64] C. Luo, M. Krishnan, M. Paulik, B. Cui, and X. Zhang, “A novel lidar-driven two-level approach for real-time unmanned ground vehicle navigation and map building,” in *Intelligent Robots and Computer Vision XXXI: Algorithms and Techniques* (J. Röning and D. Casasent, eds.), vol. 9025, pp. 17 – 27, International Society for Optics and Photonics, SPIE, 2014.

[65] R. Rajamani and D. Piyabongkarn, “New paradigms for the integration of yaw stability and rollover prevention functions in vehicle stability control,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 5046–5051, 2012.

[66] L. Tang, F. Yan, B. Zou, K. Wang, and C. Lv, “An improved kinematic model predictive control for high-speed path tracking of autonomous vehicles,” *IEEE Access*, vol. 8, pp. 51400–51413, 2020.

[67] C. Hu, Z. Wang, H. Taghavifar, J. Na, Y. Qin, J. Guo, and C. Wei, “Mme-ekf-based path-tracking control of autonomous vehicles considering input saturation,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5246–5259, 2019.

[68] Y. M. Azumi Takuya and S. Kato, “Exploring the performance of ros2,” in *Proceedings of the 13th International Conference on Embedded Software, EMSOFT 2016*, Association for Computing Machinery, Inc, 10 2016.

[69] "What is middleware?." `https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-middleware/`. Accessed: 2022-04-24.

[70] T. Wu, B. Wu, S. Wang, L. Liu, S. Liu, Y. Bao, and W. Shi, "Oops! it's too late. your autonomous driving system needs a faster middleware," *IEEE Robotics and Automation Letters*, vol. 6, pp. 7301–7308, 10 2021.

[71] H. Cui, J. Zhang, and W. R. Norris, "An enhanced safe and reliable autonomous driving platform using ros2," in *2020 IEEE International Conference on Mechatronics and Automation, ICMA 2020*, pp. 290–295, Institute of Electrical and Electronics Engineers Inc., 10 2020.

[72] O. Robotics, "About different ros 2 dds/rtps vendors." `https://docs.ros.org/en/foxy/Concepts/About-Different-Middleware-Vendors.html`. Accessed: 2021-09-25.

[73] O. Robotics, "About quality of service settings." `https://docs.ros.org/en/foxy/Concepts/About-Quality-of-Service-Settings.html`. Accessed: 2021-09-26.

[74] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 10 2004.

[75] "Gazebo sdformat specification." `http://sdformat.org/tutorials?cat=specification&`. Accessed: 2021-03-24.

[76] D. Rosato and D. Rosato, "3 - design parameter," in *Plastics Engineered Product Design* (D. Rosato and D. Rosato, eds.), pp. 161–197, Amsterdam: Elsevier Science, 2003.

[77] K. Hartani, A. Miloudi, and Y. Miloud, "Electric vehicle stability with rear electronic differential traction," in *EFEEA'10 International Symposium on Environment Friendly Energies in Electrical Applications*, Environment-Friendly Energies and Applications, 11 2010.

Appendix A

ACKERMANN DRIVE GEOMETRY

The relation between the rotation angle of the center wheel and the inner and outer wheel can be computed using the equalities given in *Equation A.1* where the R represents the radius of the turning circle. The relations between steering angles of the front wheels and the steering angle of the imaginary center wheel is given in *Equation 2.42*.

$$\begin{aligned}
\tan \delta_C &= L/R \\
\tan \delta_L &= L/(R + W/2) \\
\tan \delta_R &= L/(R - W/2)
\end{aligned} \tag{A.1}$$

The desired linear velocity of the imaginary middle wheel rotates around the center of the turning center of radius R with the same angular velocity as the inner and outer wheels [77]. Thus, using the equalities in *Equation A.2*, the relation between linear velocities of the rear wheels and linear velocity of the imaginary center wheel can be calculated and snd given in *Equation 2.43*.

$$\begin{aligned}
v_C &= w_V R \\
v_L &= w_V (R + W/2) \\
v_R &= w_V (R - W/2)
\end{aligned} \tag{A.2}$$

Appendix B

DETAILS OF DRIVING SCENARIOS

B.1 Scenario 1

GPS position error for one of the simulation runs of the control scenario is given in *Figure B.2*. The velocity error of the GPS solution for the same run is given in *Figure B.3*.

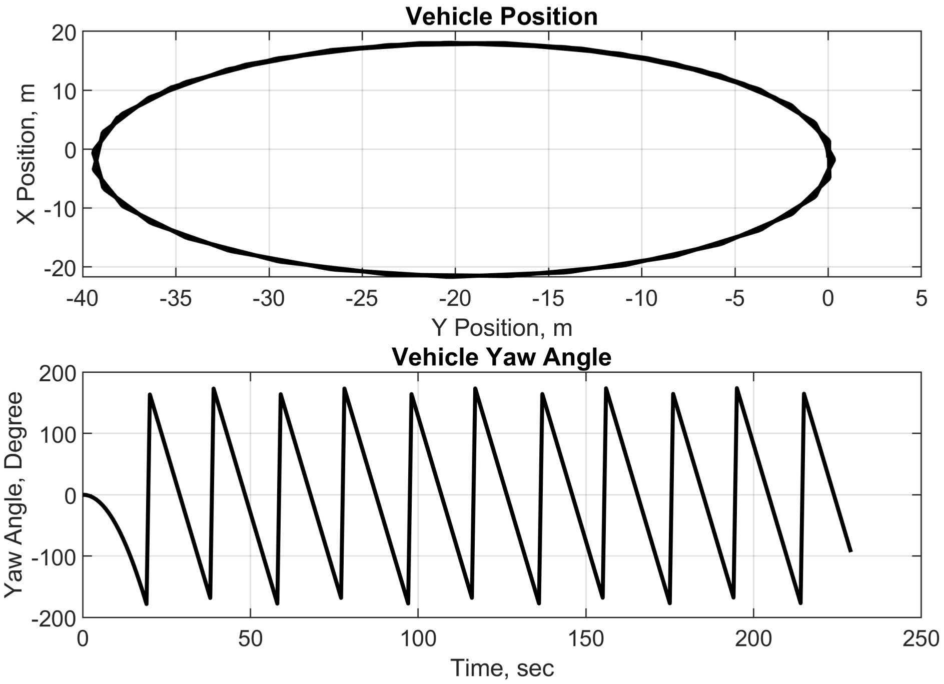

Figure B.1: Scenario 1 - Vehicle Position and Yaw Angle

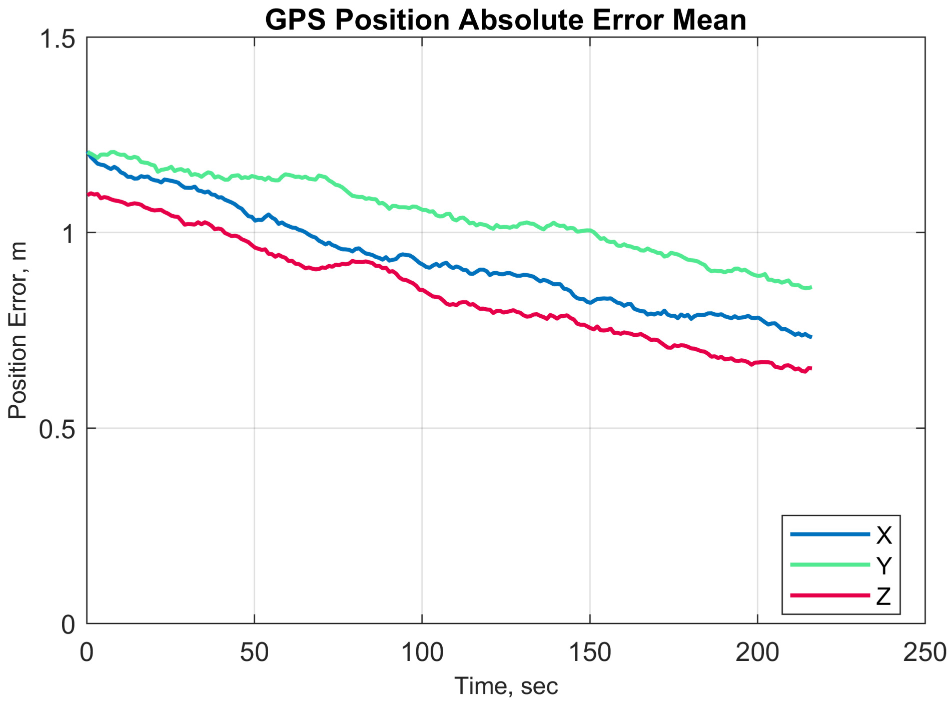

Figure B.2: Scenario 1 - GPS Position Error

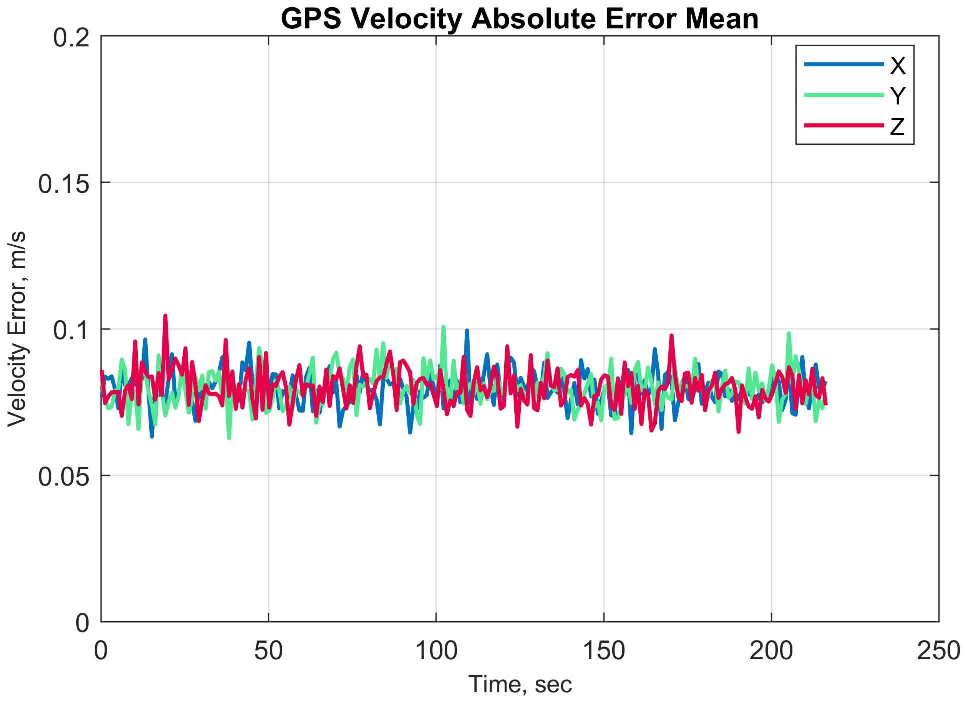

Figure B.3: Scenario 1 - GPS Velocity Error

B.2 Scenario 2

GPS position error for one of the simulation runs of the performance scenario is given in *Figure B.5*. The velocity error of the GPS solution for the same run is given in *Figure B.6*. The mean of absolute tire slip estimation errors of 100 Monte Carlo runs for every simulation steps is given in *Figures B.7 & B.8*.

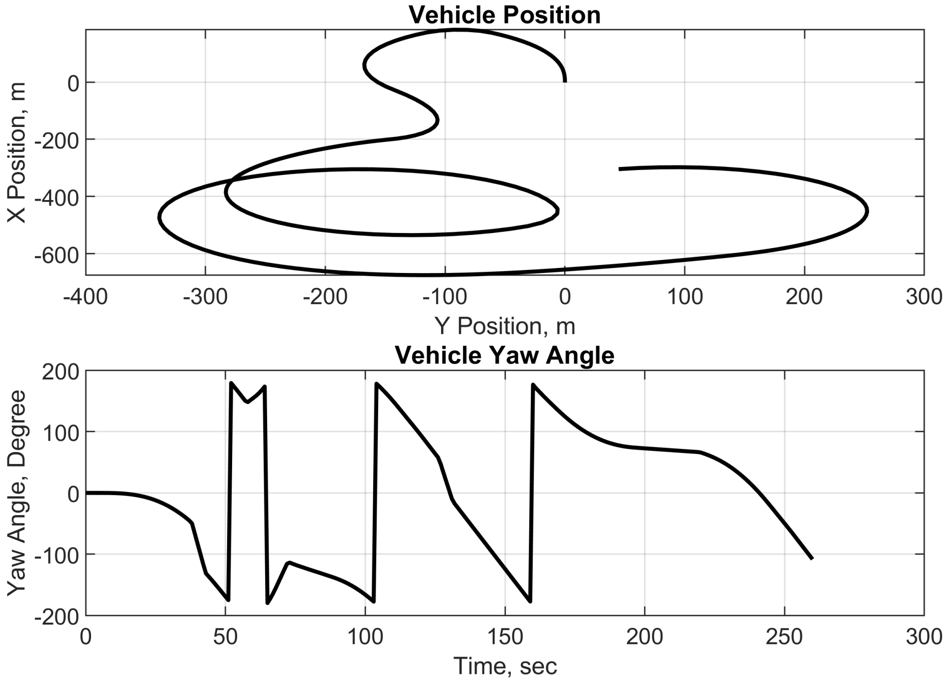

Figure B.4: Scenario 1 - Vehicle Position and Yaw Angle

B.3 Scenario 3

The contact forces generated for the third scenario are given in *Figures B.11 to B.14*. The presented figures include the ideal tire forces generated by the simulation, the result of the Pacejka tire model with ideal slip variables as an ideal model, and the estimation result for the tire forces for one simulation run.

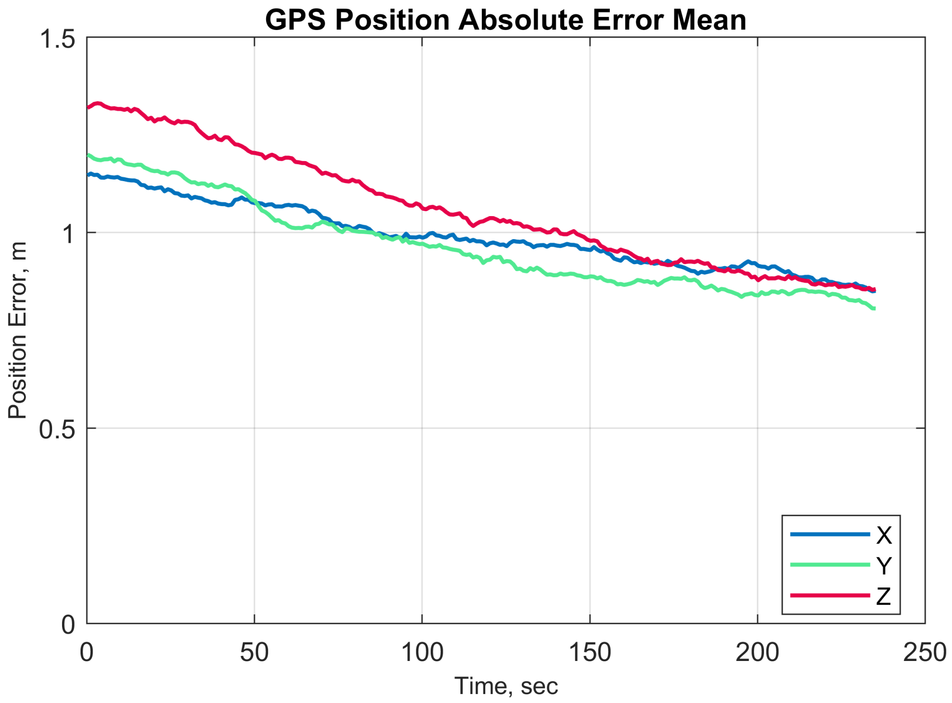

Figure B.5: Scenario 2 - GPS Position Error

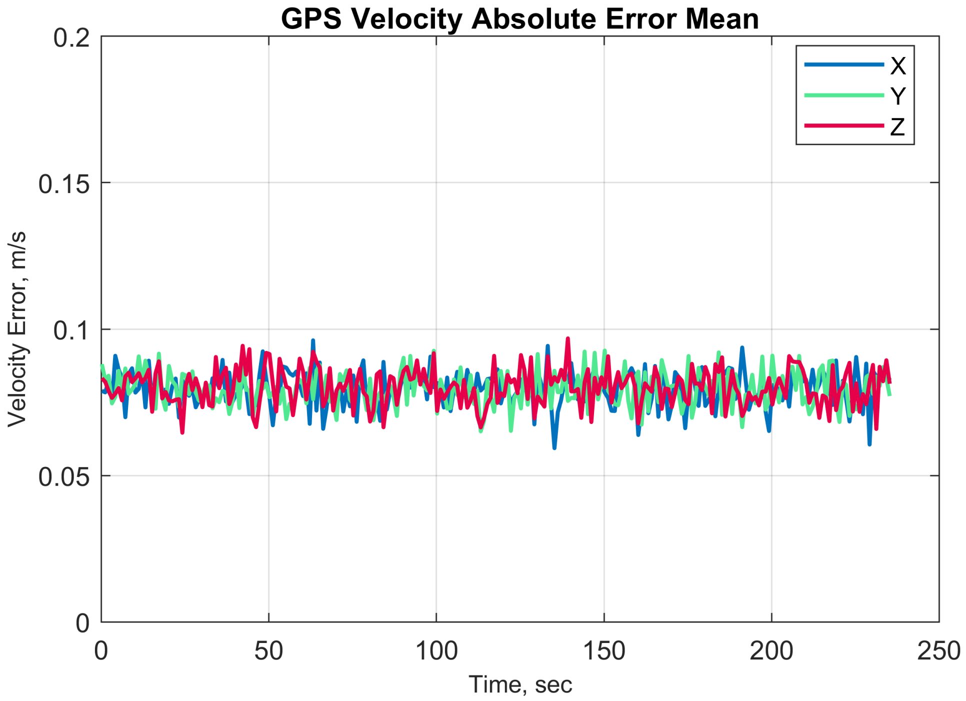

Figure B.6: Scenario 2 - GPS Velocity Error

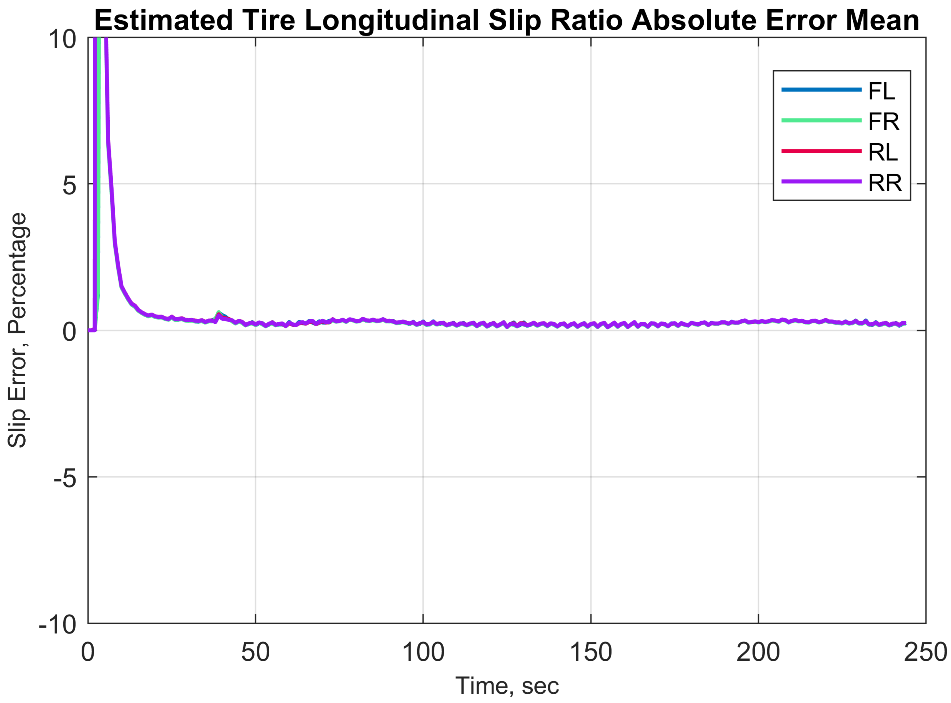

Figure B.7: Scenario 2 - Longitudinal Tire Slips Errors

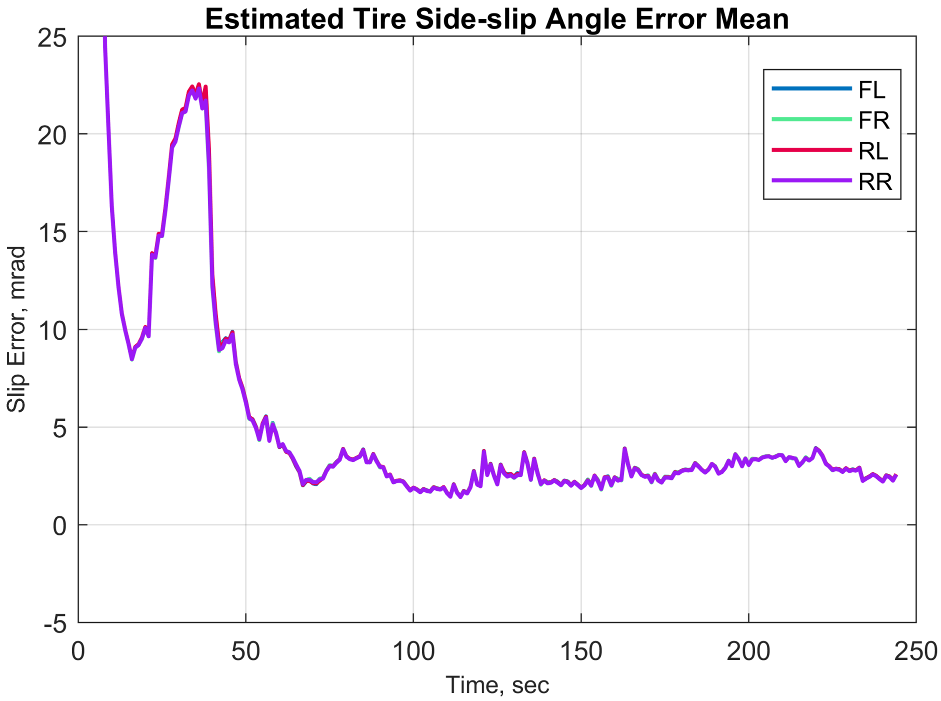

Figure B.8: Scenario 2 - Lateral Tire Slips Errors

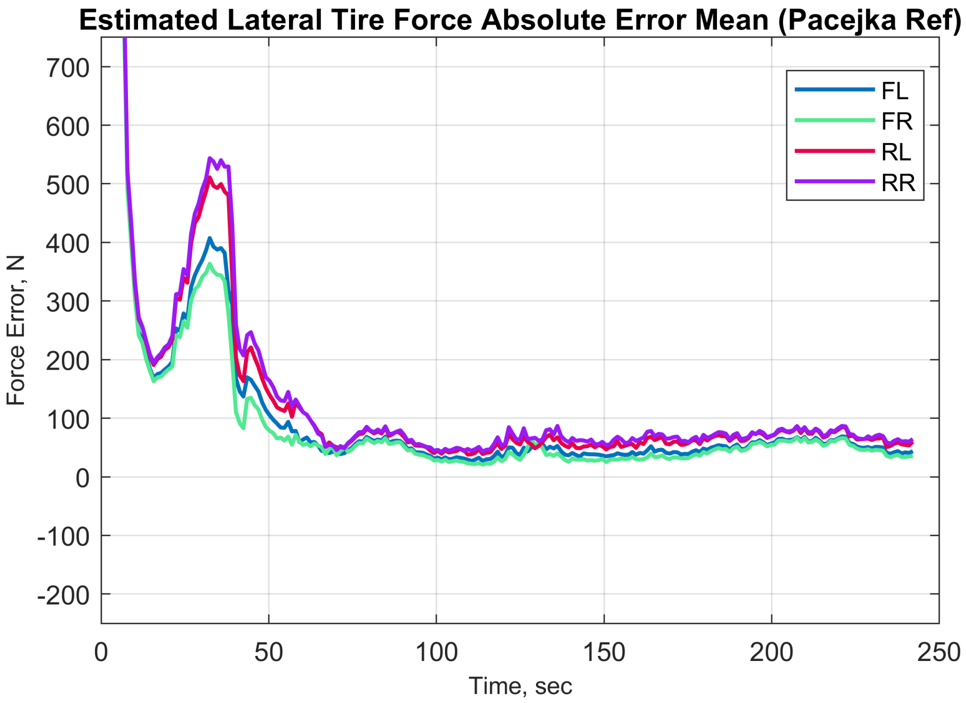

Figure B.9: Scenario 2 - Ideal Pacejka Model Referenced Lateral Tire Force Errors

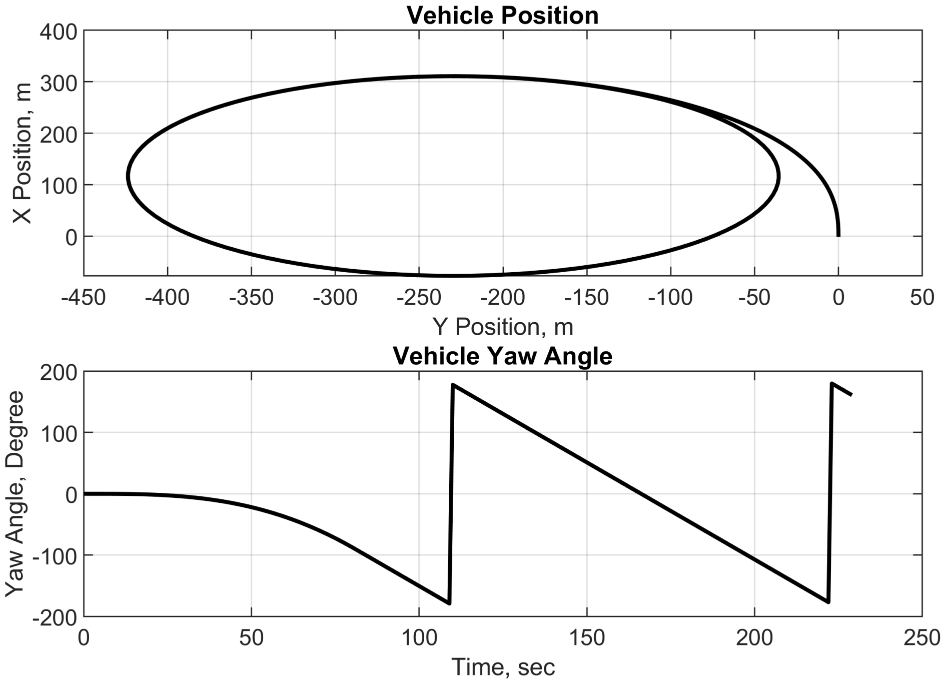

Figure B.10: Scenario 3 - Vehicle Position and Yaw Angle

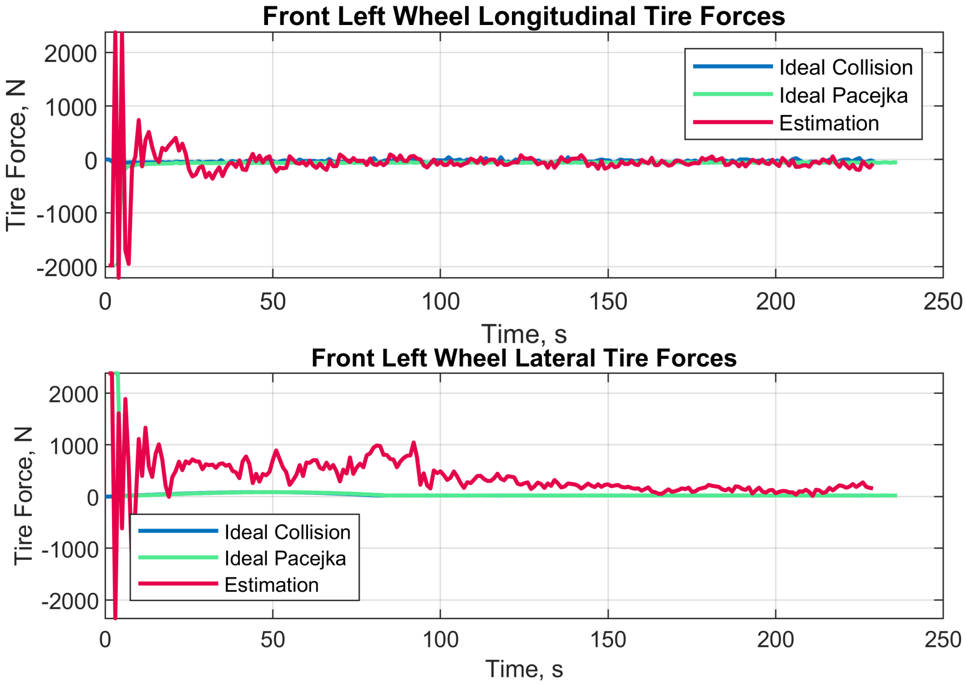

Figure B.11: Scenario 3 - Front Left Wheel Tire Forces

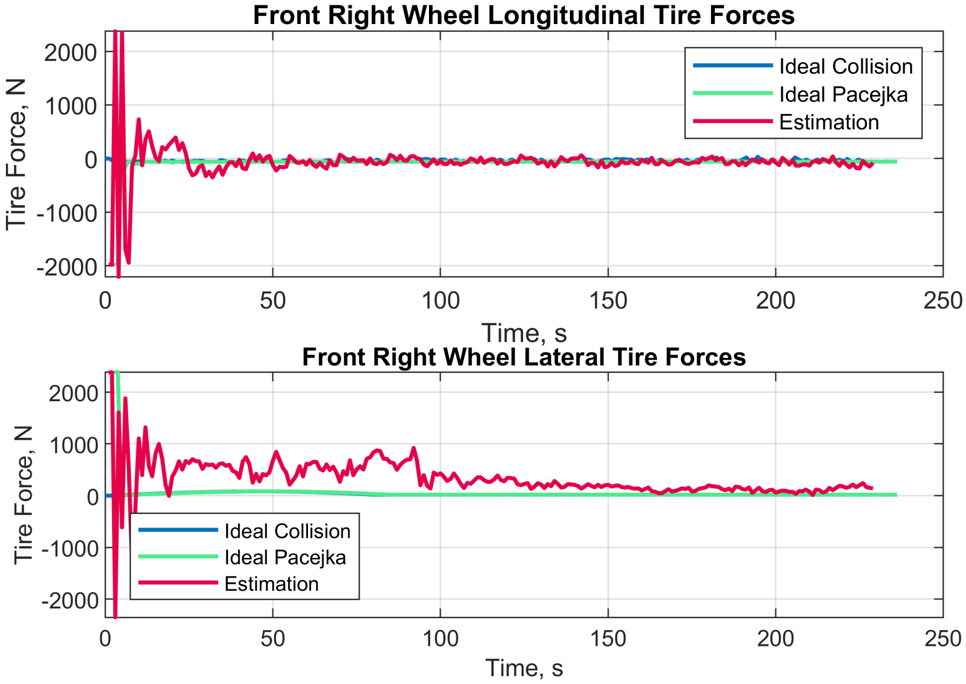

Figure B.12: Scenario 3 - Front Right Wheel Tire Forces

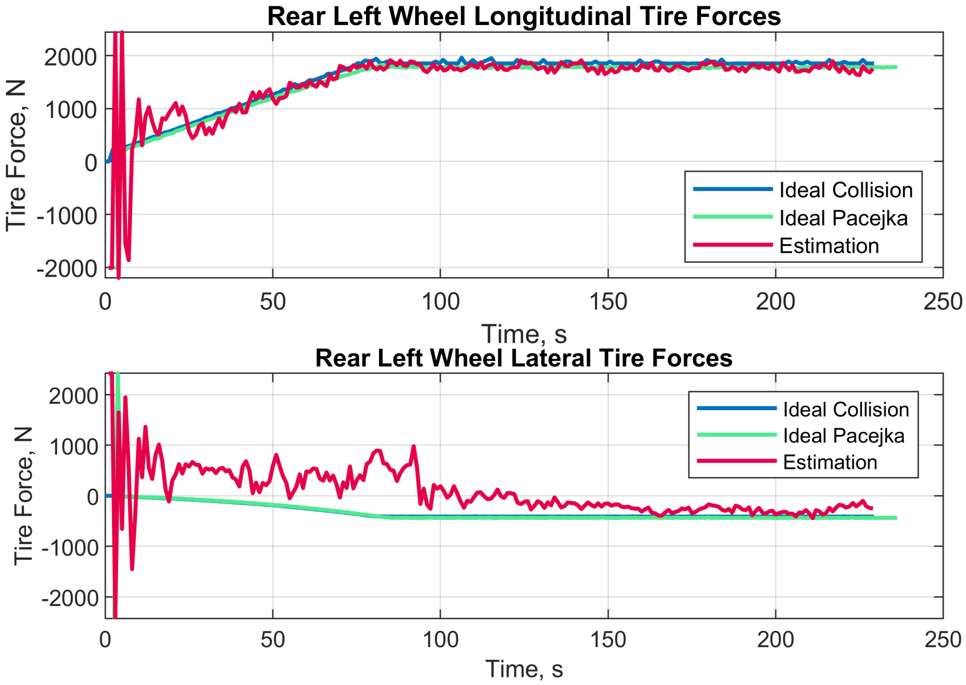

Figure B.13: Scenario 3 - Rear Left Wheel Tire Forces

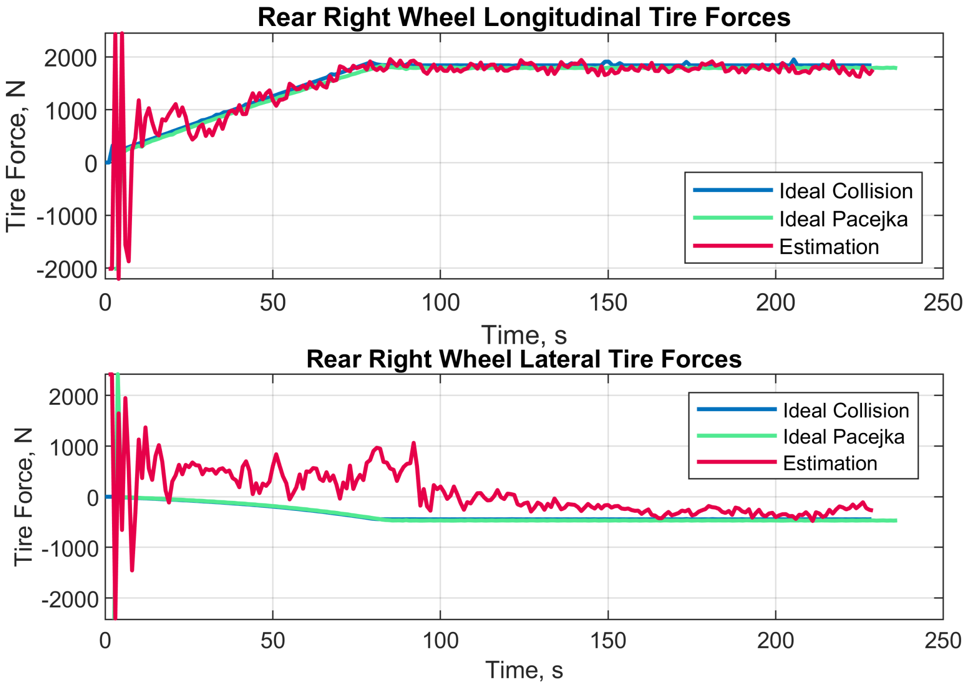

Figure B.14: Scenario 3 - Rear Right Wheel Tire Forces

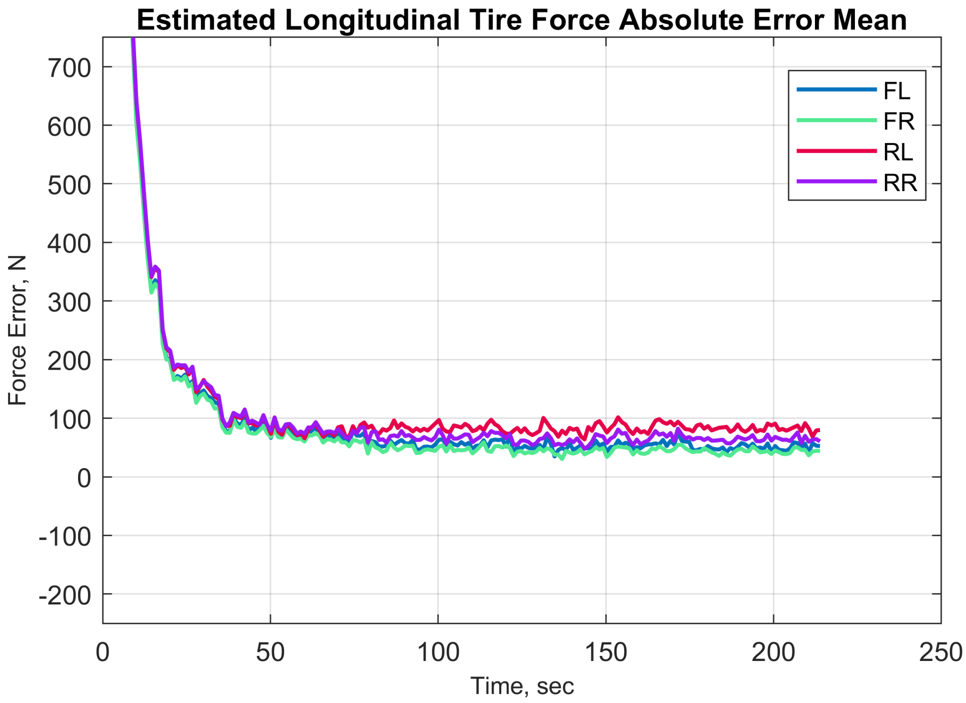

Figure B.15: Scenario 3 - Longitudinal Tire Force Errors

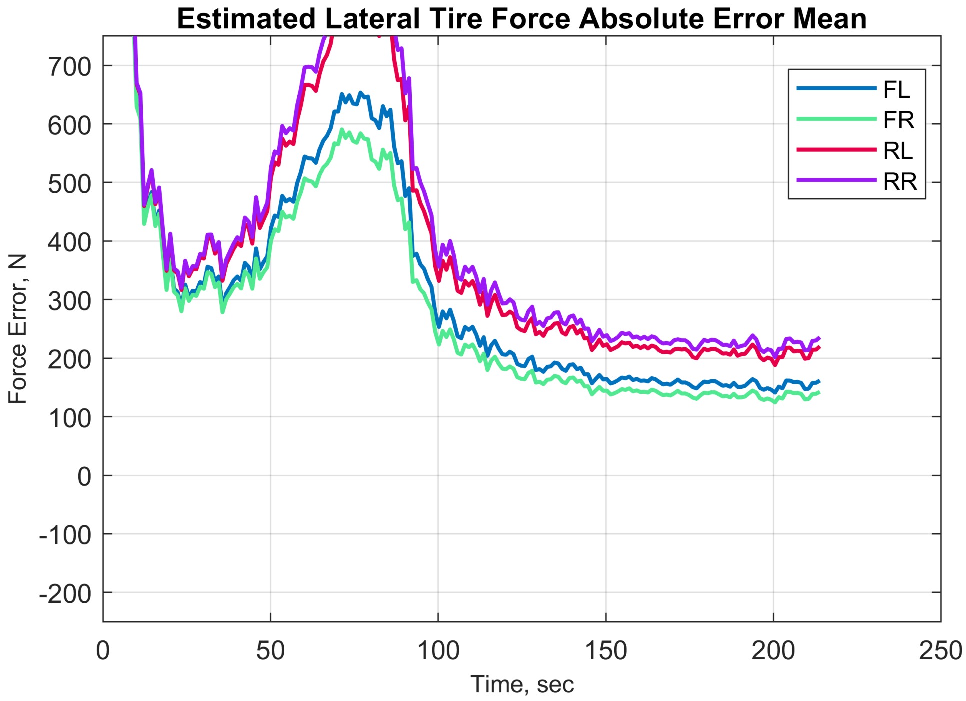

Figure B.16: Scenario 3 - Lateral Tire Force Errors

B.4 Scenario 4

The contact forces generated for the fourth scenario are given in *Figures B.11 to B.14*. The presented figures include the ideal tire forces generated by the simulation, the result of the Pacejka tire model with ideal slip variables as an ideal model, and the estimation result for the tire forces for one simulation run.

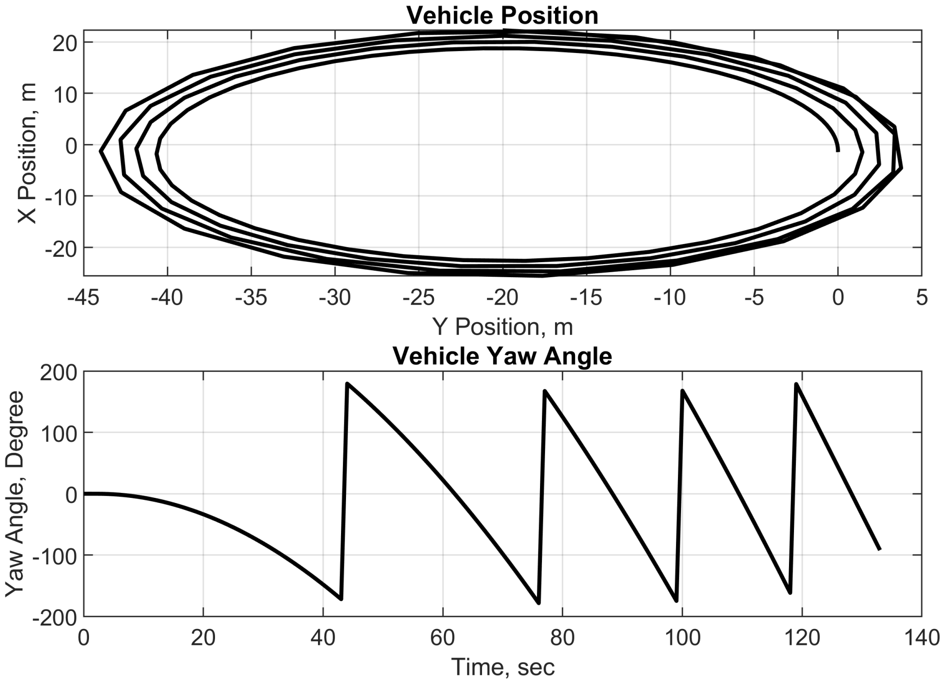

Figure B.17: Scenario 4 - Vehicle Position and Yaw Angle

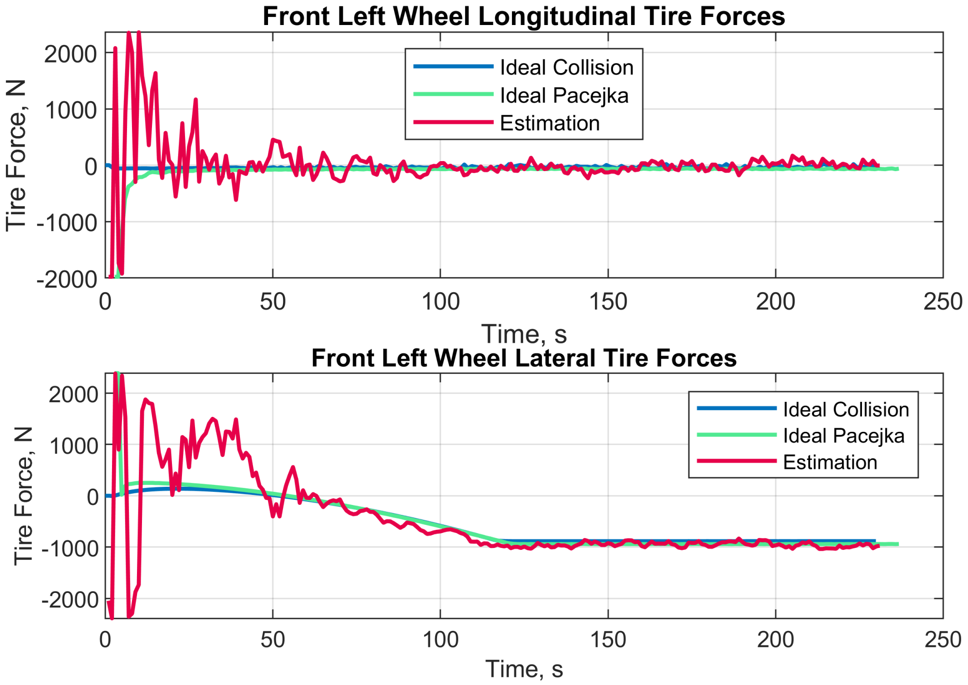

Figure B.18: Scenario 4 - Front Left Wheel Tire Forces

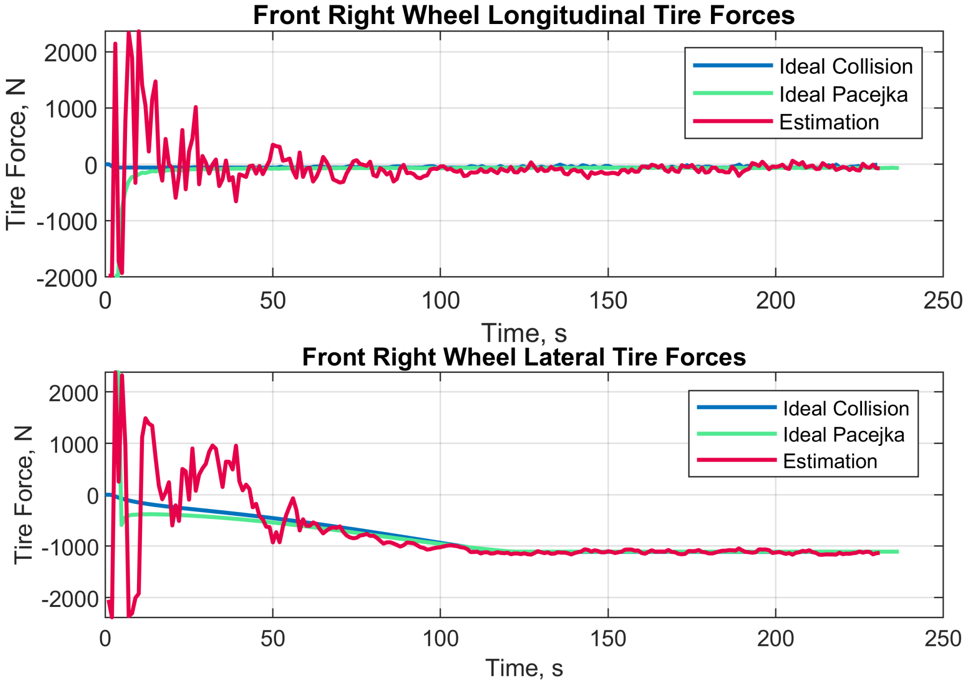

Figure B.19: Scenario 4 - Front Right Wheel Tire Forces

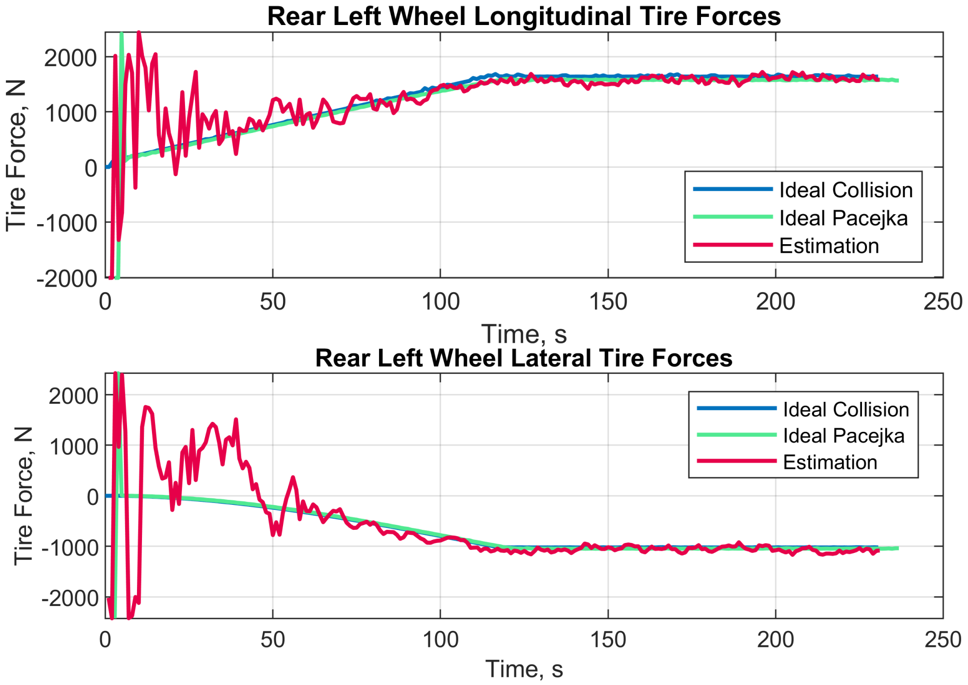

Figure B.20: Scenario 4 - Rear Left Wheel Tire Forces

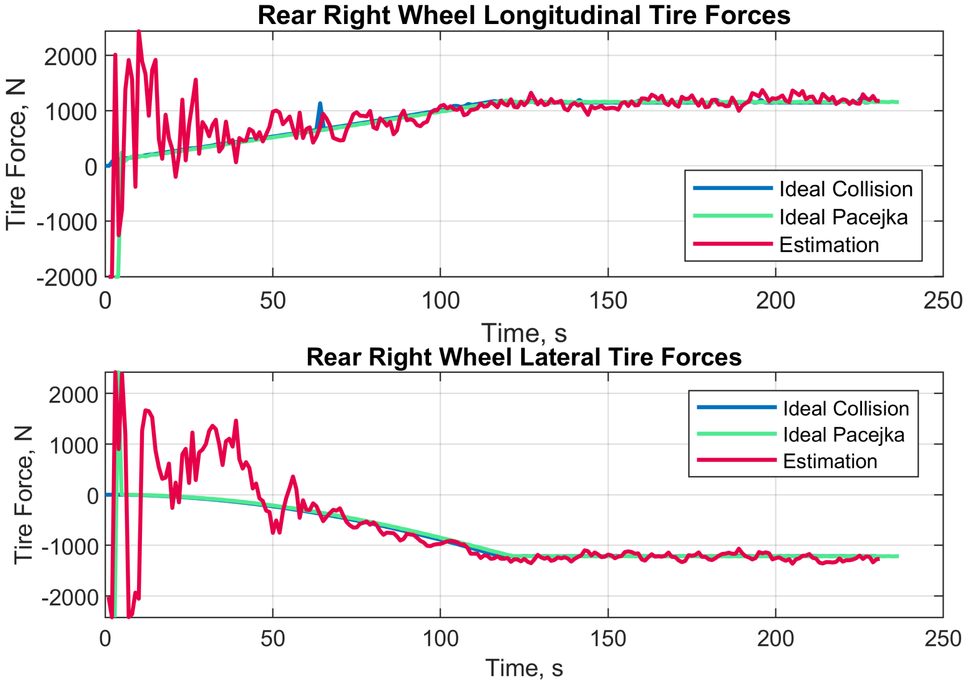

Figure B.21: Scenario 4 - Rear Right Wheel Tire Forces

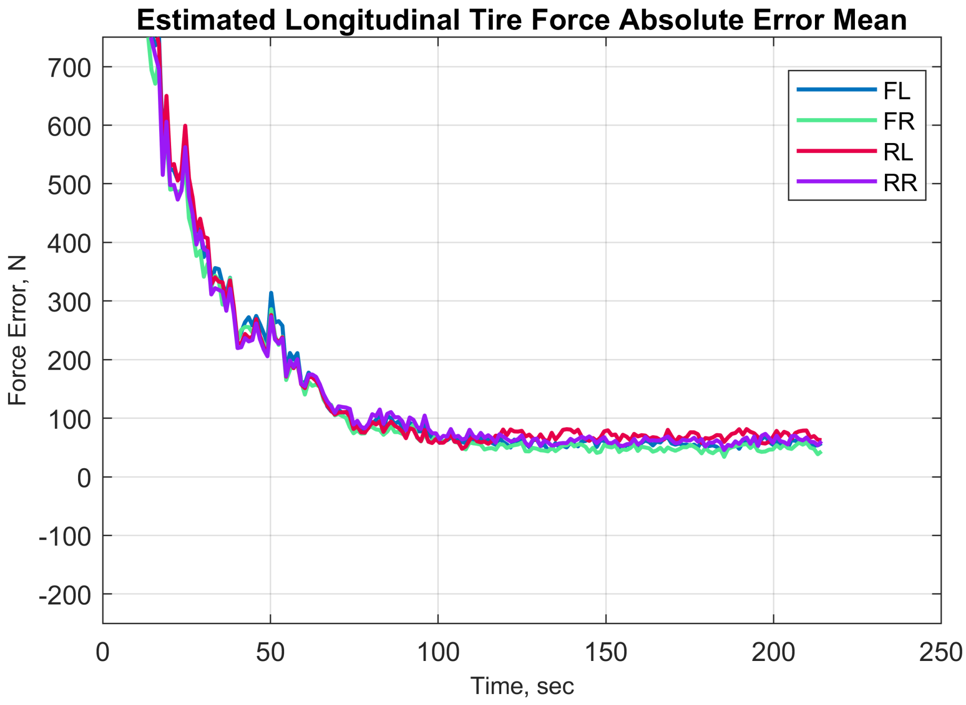

Figure B.22: Scenario 4 - Longitudinal Tire Force Errors

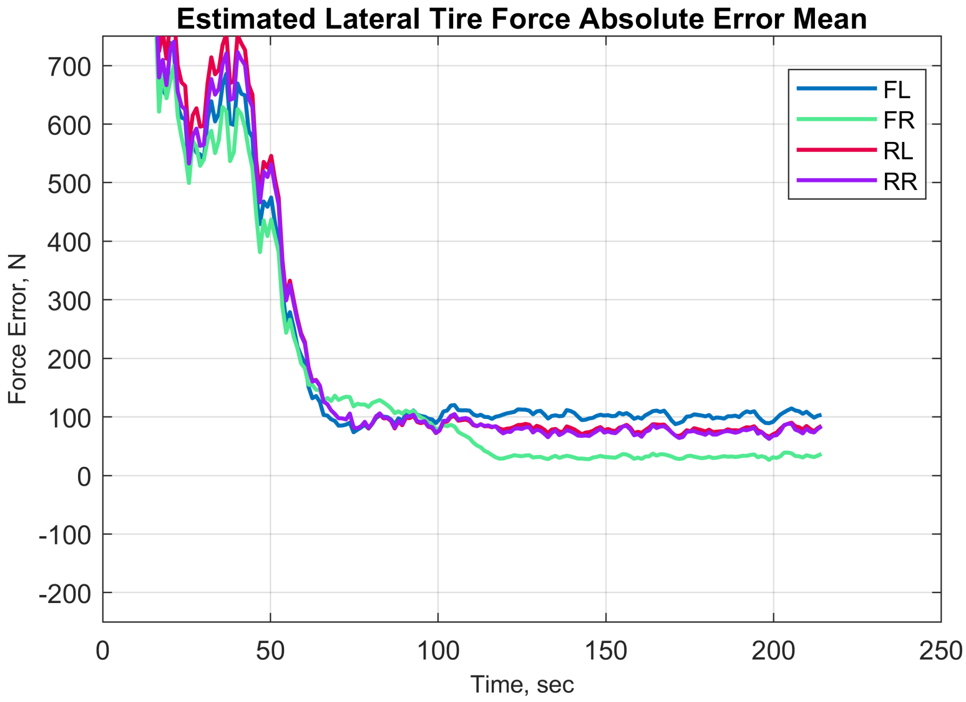

Figure B.23: Scenario 4 - Lateral Tire Force Errors