CHALMERS
UNIVERSITY OF TECHNOLOGY

Measuring wheel rotational speed using inertial sensor modalities

Master thesis in Systems, Control and Mechatronics

Akshay S Aravind
Guna Sekhar Karri

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master Thesis in Systems, Control and Mechatronics

Measuring wheel rotational speed using inertial sensor modalities

Akshay S Aravind
Guna Sekhar Karri

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology
Gothenburg, Sweden 2024

Measuring wheel rotational speed using inertial sensor modalities
Akshay S Aravind
Guna Sekhar Karri

Supervisor: Alireza Marzbanrad, Volvo Technology
Examiner: Mats Jonasson, Mechanics and Maritime Sciences

Master Thesis 2024
Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Visualization of a truck in IPG TruckMaker with IMUs mounted on the rim of it's front left wheel traveling in a highway setting.

Typeset in LaTeX
Gothenburg, Sweden 2024

Measuring wheel rotational speed using inertial sensor modalities
Akshay S Aravind
Guna Sekhar Karri
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Chalmers University of Technology

Abstract

Vehicle motion state estimation is a critical component of automotive control systems, providing the foundation for effective vehicle motion management. Estimating wheel rotational states like rotational angle, speed and acceleration of the wheel and improving the roll and pitch estimation of the road is a major part of this process. In this study, wheel speed estimation is achieved using inertial sensors mounted on the wheel of a truck. The accelerometer data from the inertial measurement units (IMUs) generated in high fidelity simulation environments is processed to estimate the desired states.
Rigid body kinematics and axis transformations are used to relate sensor measurements and our states of interest. The model is verified and tested in various driving and road conditions using simulation environments like IPG TruckMaker and Modelon Impact. Later an Extended Kalman Filter is used for estimation.
A wheel slip controller is also developed in Simulink and embedded it into TruckMaker, and estimated desired states using the Extended Kalman Filter. Additionally, the developed estimator is embedded into TruckMaker for live, closed-loop, simulation testing.
Additionally, using a quaternion representation of roll and pitch angle of the wheel, a non-rotating IMU on the wheel hub and acceleration of truck w.r.to ground frame; roll and pitch angles of the road in different scenarios were also estimated using an Extended Kalman Filter.
The estimators performed well across different scenarios and vehicle speeds, both high and low. The estimated states were in close agreement with their true values.

Keywords: motion estimation, rigid body dynamics, sensor fusion, state observers, IMU sensors, orientation estimation, nonlinear estimation, Kalman filter, IPG TruckMaker, Modelon Impact.

Preface

This report presents the outcome of our master thesis project carried out at the Department of Mechanics and Maritime Sciences at Chalmers University of Technology during the spring of 2024 in cooperation with Volvo Group Truck Technology in Gothenburg, Sweden.

Acknowledgements

We would like to thank our supervisor Alireza Marzbanrad and our manager Nicolas Andersson, at Volvo GTT, who hired us to work on this thesis. We are grateful for the constant support, guidance and supervision from Alireza Marzbanrad on all technical and engineering aspects throughout our work. We are also thankful for Murat Kumru's deep diving insights and guidance into our work. We are grateful to our examiner Mats Jonasson for his support and valuable feedback.
We are grateful to Alexander Hägglund, David Andersson and Shreyas Kogalur at IPG Automotive Sweden for providing us with licenses and technical support with TruckMaker. We thank Mohamed Takkoush at VAS and Peter Sundstrom at Modelon for their support with Modelon Impact.
We would also like to thank Johan Runeby and Lucas Haglund for their support with the Radar thesis project and also for their company. We thank the entire motion estimation team for their valuable support. It has been a great learning experience for us professionally and personally. Finally, we thank our family and friends for their constant support, which has kept us motivated and helped us in our work.

Akshay S Aravind, Guna Sekhar Karri, Gothenburg, June 2024

Contents

List of Figures

List of Tables

Nomenclature

Below is the nomenclature of indices, abbreviations, parameters, and variables that have been used throughout this thesis.

$MEMS$	MicroElectronic Mechanical Systems
INS	Inertial Navigation Systems
IMU	Inertial Measurement Unit
$AWOK$	Accelerometer-based Wheel Odometer for Kinematics
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
MIL	Model In Loop
SIL	Software In Loop
HIL	Hardware In Loop
x, y, z	Cartesian Coordinate Axes
$\{B\}$	Body frame
$\{W\}$	World frame
T	Transformation matrix
t	translation vector
$R_{x,\varphi}$	Rotation of reference frame by angle φ about x-axis
$R_{y,\theta}$	Rotation of reference frame by angle θ about y-axis
$R_{z,\psi}$	Rotation of reference frame by angle ψ about z-axis
$\mathbf{v}_A$	Velocity at point A
$\mathbf{a}_A$	Acceleration at point A
O	Point at center of wheel hub
P	Point on rim of the wheel where rotating IMU is placed
a_{o_x}	Acceleration measured at point O along x-axis in fixed inertial frame
a_{o_y}	Acceleration measured at point O along y-axis in fixed inertial frame
a_{o_z}	Acceleration measured at point O along z-axis in fixed inertial frame
a_{p_x}	Acceleration measured at point P along x-axis in rotating wheel rim
a_{p_y}	Acceleration measured at point P along y-axis in rotating wheel rim
a_{p_z}	Acceleration measured at point P along z-axis in rotating wheel rim
ω	Angular velocity of the wheel
θ	Angular displacement of the wheel
α	Angular acceleration of the wheel

Nomenclature

r_1	Distance of IMU on rim from wheel center in radial direction
a_{p_θ}	Tangential acceleration measured by rim IMU
a_{p_r}	Radial acceleration measured by rim IMU
K	Observer gain
x	State vector
$\hat{x}$	Estimated state vector
$\tilde{x}$	State estimation error
y	Output/Measurement vector
q_k	White Gaussian noise for motion model in state estimation
Q_k	Covariance of white Gaussian noise for motion model in state estimation
r_k	White Gaussian noise for measurement model in state estimation
R_k	Covariance of white Gaussian noise for measurement model in state estimation
$\hat{x}_{k\|k}$	Posterior mean
$\hat{P}_{k\|k}$	Posterior covariance
$\hat{x}_{k+1\|k}$	Predicted state vector
$\hat{P}_{k+1\|k}$	Predicted covariance
$\hat{x}_{k+1\|k+1}$	Updated state vector
$\hat{P}_{k+1\|k+1}$	Updated covariance
S_{k+1}	Innovation covariance
g	Acceleration due to gravity
δ	Rotation angle of unit vector
e_v	Vector of unit length
q	Unit Quaternion
Φ	Roll angle
Θ	Pitch angle
Ψ	Yaw angle
t	Time
T	Sampling time
v_k	White Gaussian noise for motion model in quaternion estimation
M_k	Covariance of white Gaussian noise for motion model in quaternion estimation
w_k	White Gaussian noise for measurement model in quaternion estimation
$h(x)$	Non-linear function
h'	Gradient of $h(x)$
b	Axes misalignment in accelerometer model
ξ_1	White noise drift in accelerometer model
s	Scale variable in accelerometer model
ξ_2	Random walk drift in accelerometer model
Δ_e	Environmental drift noise in accelerometer model
d	Quantization model in accelerometer model
ϵ_0	Initial position of rim IMU
μ	Mean
σ	Standard deviation

M_b	Braking torque
F_x	Longitudinal tire force
J	Wheel moment of inertia
r_r	Effective rolling radius
r_b	Loaded radius at which braking force acts
σ_{d_x}	Slip limit
$\dot{v}_x$	Derivative of the longitudinal velocity at the wheel coordinate system
σ_x	Longitudinal wheel slip
ϕ	Proportional term
s	Sliding surface
a_x	linear acceleration without gravity along x-axis in vehicle body fixed frame
a_y	linear acceleration without gravity along y-axis in vehicle body fixed frame
a_z	linear acceleration without gravity along z-axis in vehicle body fixed frame

1
Introduction

1.1 Background

A key component of vehicle motion estimation process is acquiring precise knowledge of wheel rotational speeds. This information is a critical aspect in implementing advanced control strategies like wheel slip control within vehicles. Wheel slip control focuses on regulating tire slip to improve traction and prevent loss of control, especially in challenging driving conditions like low friction surfaces or emergency braking.
Roll and pitch angle of the truck are also important components of vehicle motion estimation, roll angle of the truck can also be used in various control functions, and slope angle information can be used in powertrain control and demand strategies.

1.2 Related Work

Accurate wheel heading, roll and pitch estimation can be estimated with high resolution, and the result is nearly bias-free using Micro-electromechanical systems (MEMS) sensors [2]. These sensors address challenges such as low-frequency noise and sensor bias, which affect measurement accuracy.
Traditionally, wheel encoders have been used to measure the angular wheel speed by discretizing the wheel circumference. However, the technology has limitations when the wheel slips or skids and measurements can be inaccurate at low speeds due to limited resolution of the sensor both in the time and spatial domain. The usage of accelerometers as continuous wheel encoders overcome these limitations. As accelerometers pose challenges due to noise and techniques like Extended Kalman Filtering have been used to model behavior and minimize errors. Results indicate accurate wheel tracking, with errors below 1% for odometry estimation even in the presence of disturbances, which show the potential of this approach [3].
Introducing Wheel-INS, a complete dead reckoning solution designed to enhance the accuracy and robustness of inertial navigation systems (INS) in wheeled robots. Wheel-INS is based on a wheel-mounted micro-electromechanical system (MEMS) inertial measurement unit (IMU) to achieve these improvements without adding additional component costs. Mounting the IMU to the center of a non-steering wheel offers two significant advantages. Firstly, by utilizing gyroscope outputs, Wheel-INS can calculate wheel speed, thereby replacing traditional odometers and mitigating error drift in the INS. Secondly, the rotation of the wheel enables cancellation of the constant bias error in the inertial sensor, which improves accuracy and reliability

of the system. The maximum position drift of Wheel-INS in the horizontal plane is less than 1.8% of the total traveled distance, reduced by 23% compared to the conventional odometer-aided INS [4].
A multi low-cost INS configuration in land vehicle where two low-cost IMU sensors are mounted on the center of rear wheels of the land vehicle to estimate the vehicle's forward velocity through the gyroscopes located in the perpendicular direction of the wheel. A differential wheel odometry based on the Inertial Measurement Unit (IMU) mounted on the rear wheels is proposed to estimate the vehicle's change of heading [5].
Introducing a low-cost alternative called the Accelerometer-based Wheel Odometer for Kinematics determination (AWOK) system that consists of a single-axis accelerometer mounted radially at the center of a wheel, providing direct distance measurements rather than just velocities. The AWOK system is immune to stochastic errors in MEMS inertial sensors and achieves a remarkable relative accuracy of 0.15% in determining the distance covered by a vehicle [6].

1.3 Objective

In this thesis, estimating the wheelangular speed using IMU will be the major objective. This involves developing a model of the sensor, assessing it's specifications, simulating it under different conditions, processing sensor measurements, quantify the accuracy that can be achieved using the method and to estimate motion states including but not limited to wheel angular speed. By pursuing this path, our goal is to develop process models, design estimators and conduct co-simulation to analyze estimator performance and assess the technology.
Also of importance is the estimation of roll and slope angles of the truck using accelerometer data.

1.4 Scope

- To develop a model of the wheel-mounted inertial sensor embedded in the high-fidelity simulation environment, to simulate sensor outputs under different driving scenarios across various road conditions.
- To verify the accuracy of simulation outputs.
- To engineer and implement a method that processes sensor readings to attain precise wheel rotational speed information with high frequency output.
- To assess the use of different configurations of the sensor(s) for effectively measuring the signals of interest. This involves the investigation of the physical quantities to be observed by the sensor, the number of sensors to be employed, and their specifications, such as sensor output frequency and error characteristics.
- To investigate limitations of the technology.
- Attain roll and pitch information of the vehicle using accelerometers and radar acceleration estimates.

1.5 Outline

The report is organized into the following chapters:

- **Theory:** The theoretical background and analysis utilized in the project are discussed.
- **Methods:** This chapter explains the experimental setup in simulation environments and the methodologies implemented for verifying and estimating the desired states.
- **Results:** This chapter presents the results obtained from various tests through the post-processing of data generated from simulation environment.
- **Discussions:** This chapter discusses the findings and implications of the results obtained. It also contains conclusions, future work and sustainability aspects of our work.

2
Theory

In this chapter, the theoretical concepts used in this thesis will be discussed. This section contains the common terminologies which will be used in further chapters.

2.1 Homogeneous transformation

To fully describe the pose of a rigid object in a three-dimensional space, six dimensions are required: three for its position (x, y and z) and three for its orientation (*roll*, *pitch* and *yaw*). The process of transforming a coordinate frame from one coordinate system to another is known as axis transformation.

2.1.1 2D transformations

To represent a rigid body in 2 dimensions, three coordinates are needed (x, y and θ).
Consider a point **P** whose pose can described by the position and orientation of frame $\{B\}$ w.r.to. frame $\{W\}$ and as shown in Figure 2.1. Frame $\{B\}$ is oriented at an angle θ with frame $\{W\}$. The point **P** is located at W_P in frame $\{W\}$ and at B_P in frame $\{B\}$. With respect to $\{W\}$, the origin of $\{B\}$ is displaced by the vector $t = (x, y)^T$, which represents translation. Frame $\{B\}$ is rotated counterclockwise by an angle θ, which represents rotation R. The combination of both translation and rotation constitutes a transformation of axes.
The coordinates of point **P** in the world frame $\{W\}$ can be computed from its coordinates in the body frame $\{B\}$ using a transformation (rotation and translation) matrix:

$$\begin{aligned} T &= \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \\ T &= \begin{bmatrix} cos\theta & -sin\theta & x \\ sin\theta & cos\theta & y \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2.1}$$

The transformation equation is given by:

$$W_p = T \times B_P \tag{2.2}$$

$$W_P = \begin{bmatrix} cos\theta & -sin\theta & x \\ sin\theta & cos\theta & y \\ 0 & 0 & 1 \end{bmatrix} \times B_P \tag{2.3}$$

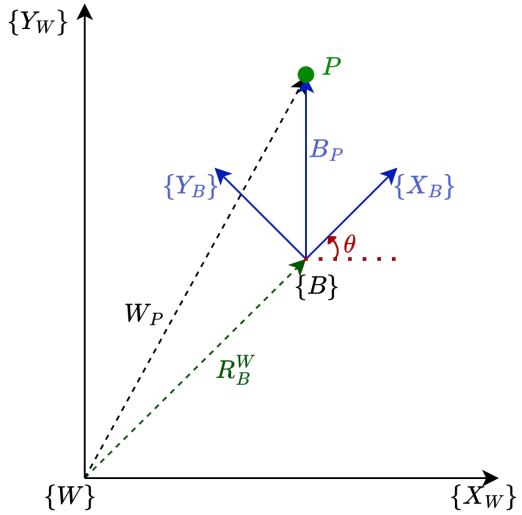


Figure 2.1: 2D Transformation

2.1.2 3D transformations

Similar to 2D transformation, for the 3D transformation of a point **P** from the body frame $\{B\}$ to the world frame $\{W\}$, we add an extra dimension to the rotation matrix $R \in 3 \times 3$ and to the translation vector $t \in 3 \times 1$ the following approach:

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$
$$T = \begin{bmatrix} cos\theta & -sin\theta & 0 & x \\ sin\theta & cos\theta & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

The transformation equation is therefore:

$$W_p = T \times B_P \tag{2.5}$$
$$W_P = \begin{bmatrix} cos\theta & -sin\theta & 0 & x \\ sin\theta & cos\theta & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times B_P \tag{2.6}$$

The rotation R in a 3D transformation can be represented by no more than three rotations called Euler angle rotations.

2.1.3 Euler angle rotations:

Rotation matrices, which are used to rotate vectors in a coordinate frame, can be utilized to apply the transformation. The following are three main rotations along

the x, y and z axes by an angle θ:

$$\mathbf{R_x}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix} \tag{2.7}$$

$$\mathbf{R_y}(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \tag{2.8}$$

$$\mathbf{R_z}(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

The position vectors of coordinate system are multiplied with sequence of the rotation matrices to obtain the desired coordinate frame.
Euler angles represent the rotation of a body in three-dimensional space through a sequence of three rotations about the axes of the coordinate system. These set of Euler angles can be represented as $\zeta = [\varphi \ \ \theta \ \ \psi]^T$.
Consider the rotation matrix that expresses the basic rotation about one of the coordinate axes as a function of a single angle. A generic rotation matrix can then be obtained by composing a suitable sequence of three elementary rotations, ensuring that two successive rotations are not made about parallel axes. This results in 12 distinct sets of angles out of the 27 possible combinations, with each set representing a triplet of Euler angles [7]. One example is ZYX rotation, i.e., $\mathbf{R} = \mathbf{R}_{\mathbf{z},\psi}\ \mathbf{R}_{\mathbf{y},\theta}\ \mathbf{R}_{\mathbf{x},\varphi}$, which means:

- Rotate the reference frame by angle ψ about z-axis ($\mathbf{R}_{\mathbf{z},\psi}$).
- Rotate the reference frame by angle θ about y-axis ($\mathbf{R}_{\mathbf{y},\theta}$).
- Rotate the reference frame by angle φ about x-axis ($\mathbf{R}_{\mathbf{x},\varphi}$).

From rotation matrices mentioned in Equations (2.7) to (2.9),

$$\begin{aligned} \mathbf{R_{zyx}} &= \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\varphi & -sin\varphi \\ 0 & sin\varphi & cos\varphi \end{bmatrix} \\ &= \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\varphi - \sin\psi\cos\varphi & \cos\psi\sin\theta\cos\varphi + \sin\psi\sin\varphi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\varphi + \cos\psi\cos\varphi & \sin\psi\sin\theta\cos\varphi - \cos\psi\sin\varphi \\ -\sin\theta & \cos\theta\sin\varphi & \cos\theta\cos\varphi \end{bmatrix} \end{aligned} \tag{2.10}$$

Assuming,

$$\mathbf{R_{zyx}} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.11}$$

Comparing Equations (2.10) and (2.11), $r_{11} \neq 0$ & $r_{21} \neq 0$ and $\theta \in (0, \pm\ \pi)$, we get:

$$\varphi = tan^{-1}(r_{21}, r_{11}) \tag{2.12}$$

$$\theta = tan^{-1}(-r_{31}, \pm\sqrt{r_{21}^2 + r_{11}^2}) \tag{2.13}$$

$$\psi = tan^{-1}(r_{32}, r_{33}) \tag{2.14}$$

2.2 Relative Acceleration:

Relative acceleration is the rate of change of velocity of one object with respect to another. Consider two points A and B on a rigid body, the relative velocity of two points in a plane motion is given as [8]:

$$\mathbf{v}_A = \mathbf{v}_B + \mathbf{v}_{A/B} \tag{2.15}$$

The rate of change of equation (2.15) gives relative acceleration as:

$$\mathbf{a}_A = \mathbf{a}_B + \mathbf{a}_{A/B} \tag{2.16}$$

The acceleration at point A is gives as the vector sum of acceleration at point B and acceleration of A with respect to B in a plane motion.

2.2.1 Relative acceleration in Rotational motion:

Suppose two points, A and B separated by a distance r lie on a rigid body as in figure 2.2, with the rigid body having translation acceleration a_B and point A performing circular motion about point B. Point A appears to have both normal and tangential components of motion with respect to point B. So, the relative acceleration from equation (2.16) can be written as:

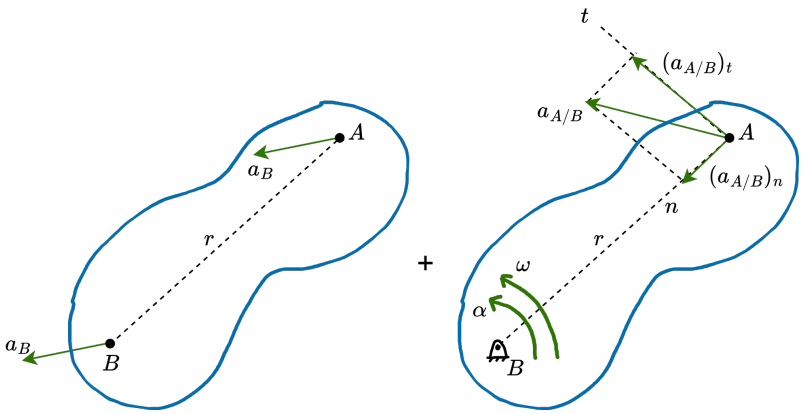


Figure 2.2: Relative acceleration and velocity of a rigid body

$$\mathbf{a}_A = \mathbf{a}_B + (\mathbf{a}_{A/B})_n + (\mathbf{a}_{A/B})_t \tag{2.17}$$

The acceleration components can be written as:

$$(\mathbf{a}_{A/B})_n = \omega \times (\omega \times r)$$
$$(\mathbf{a}_{A/B})_r = \alpha \times r$$

$$\implies \mathbf{a}_A = \mathbf{a}_B + \omega \times (\omega \times r) + \alpha \times r \tag{2.18}$$

Where ω is the angular velocity, $\alpha{=}\dot{\omega}$ is the angular acceleration of the body and "$\times$" is the vector cross product.

2.3 Wheel coordinate system

The wheel coordinate system uses a right-hand rule based X-Y-Z sequence based on the ISO 8855 standard [9]. The orientation of the system is shown in figure 2.3.

The coordinates of the wheel(X_w, Y_w and Z_w) are in a Z-up orientation with origin at the wheel center, X_w is parallel to the direction of travel and the wheel plane, Y_w is parallel to the wheel-spin axis and Z_w is pointing upwards parallel to the wheel plane.

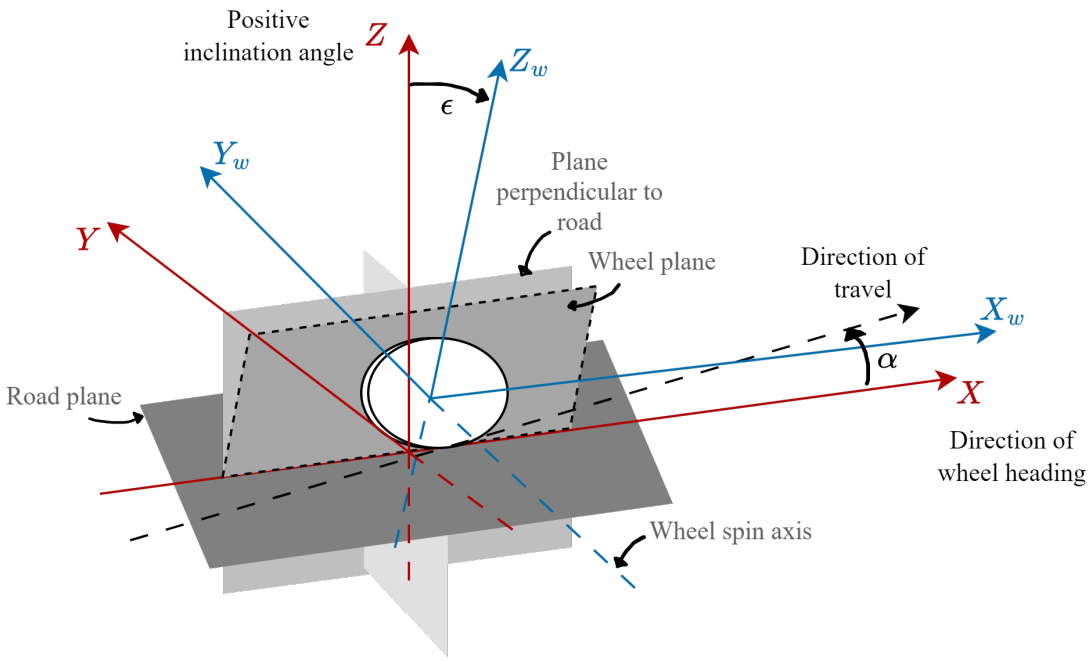


Figure 2.3: Wheel coordinate system

2.4 Experimental Setup

If the motion of the rigid body is described in 3 dimensions (x, y and z), consider two points on the wheel, one at the center of the wheel hub O and the other P at wheel rim separated by distance r_1 as in figure 2.4

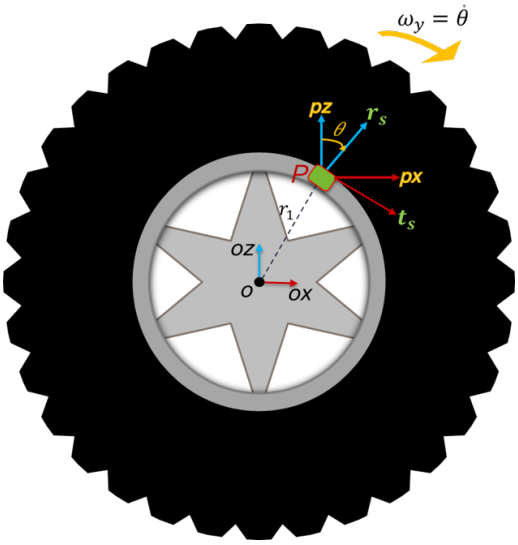


Figure 2.4: Axes of Wheel and IMU

The 3 axes at these points is according to the wheel coordinate system described in section 2.3. These frames are non rotating i.e. they do not rotate with the wheel's rotation, but they follow the topography of road surface the truck and its wheel(s) move on (road elevation and banks). The acceleration of gravity is therefore projected on the x and z axes in these frames.
Given the fact that wheel rim is a rigid body, then the equation (2.18) in 3D is given as follows:

$$\mathbf{a_p} = \mathbf{a_o} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) + \boldsymbol{\alpha} \times \mathbf{r} \tag{2.19}$$

where

$$\mathbf{a_p} = \begin{bmatrix} a_{p_x} \\ a_{p_y} \\ a_{p_z} \end{bmatrix} \quad \mathbf{a_o} = \begin{bmatrix} a_{o_x} \\ a_{o_y} \\ a_{o_z} \end{bmatrix} \quad \boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad \mathbf{r} = \begin{bmatrix} r_1 sin\theta \\ 0 \\ r_1 cos\theta \end{bmatrix} \quad \boldsymbol{\alpha} = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix}$$

Therefore equation (2.19) is:

$$\begin{bmatrix} a_{p_x} \\ a_{p_y} \\ a_{p_z} \end{bmatrix} = \begin{bmatrix} a_{o_x} - r_1 sin\theta\omega_y^2 + \omega_z(\omega_x r_1 cos\theta - \omega_z r_1 sin\theta) + \alpha_y r_1 cos\theta \\ a_{o_y} - \alpha_x r_1 cos\theta + \alpha_z r_1 sin\theta + \omega_y\omega_z r_1 cos\theta + \omega_x\,\omega_y r_1 sin\theta \\ a_{o_z} - r_1 cos\theta\omega_y^2 - \omega_x(\omega_x r_1 cos\theta - \omega_z r_1 sin\theta) - \alpha_y r_1 sin\theta \end{bmatrix} \tag{2.20}$$

An IMU is placed at point P with its own sensor frame consisting of tangential, lateral and radial directions given as t_s, y_s and r_s respectively. This sensor frame rotates with the wheel's rotation and makes an angle θ with the non rotating frame at point P, with θ=0 when the two axes are aligned and increasing clockwise. The IMU on the rim and the sensor frame rotate with the wheel at a speed of $\omega_y = \dot{\theta}$ whereas, the frames at points P and O are non rotating and follow the trajectory of the wheel in 3D space as described earlier.
Therefore, the non rotating frame at P can be transformed to the sensor frame also at point P by a rotation matrix. Hence the acceleration vector from equation (2.20) is reformulated as:

$$\underbrace{\begin{bmatrix} a_{p_\theta} \\ a_{p_y} \\ a_{p_r} \end{bmatrix}}_{\mathbf{a_{ps}}} = \underbrace{\begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix}}_{R_y(-\theta)} \underbrace{\begin{bmatrix} a_{p_x} \\ a_{p_y} \\ a_{p_z} \end{bmatrix}}_{\mathbf{a_p}}$$

$$\begin{bmatrix} a_{p_\theta} \\ a_{p_y} \\ a_{p_r} \end{bmatrix} = \begin{bmatrix} a_{o_x}cos\theta - a_{o_z}sin\theta + \frac{r_1\omega_x^2 sin2\theta}{2} + r_1\omega_x\omega_z cos2\theta - \frac{r_1\omega_z^2 sin2\theta}{2} + r_1\alpha_y \\ a_{o_y} + r_1\omega_x\omega_y sin\theta + r_1\omega_y\omega_z cos\theta - r_1\alpha_x cos\theta + r_1\alpha_z sin\theta \\ a_{o_x}sin\theta + a_{o_z}cos\theta - \frac{r_1\omega_x^2 cos2\theta}{2} - \frac{r_1\omega_x^2}{2} + r_1\omega_x\omega_z sin2\theta - r_1\omega_y^2 + \frac{r_1\omega_z^2 cos2\theta}{2} - \frac{r_1\omega_z^2}{2} \end{bmatrix} \tag{2.21}$$

where a_{p_θ}, a_{p_y} and a_{p_r} are the transformed acceleration axes in tangential, lateral and radial directions respectively. It is assumed that rate of steering angle does not change a lot during truck maneuvers and the rotation along z direction i.e., Rz is ignored.

2.4.1 Assumptions and model simplification:

For the equation (2.21), certain assumptions are made to simplify the model. Assuming the trucks are not being driven at high lateral accelerations and not high road roll and pitch angle for vast majority of its operation, w_x and w_z and their derivatives α_x and α_z can be assumed to be very small compared to ω_y and α_y or neglected. Also ignoring the lateral acceleration a_{p_y}, the equation (2.21) can be simplified as:

$$\begin{aligned} a_{p_\theta} &= r_1\alpha_y + a_{o_x}cos\theta - a_{o_z}sin\theta \\ a_{p_r} &= -r_1\omega_y^2 + a_{o_z}cos\theta + a_{o_x}sin\theta \end{aligned} \tag{2.22}$$

2.5 Observer based estimation model

State observers or state estimators are systems used to reconstruct internal state(s) $x(t) \in \mathbb{R}^n$ of a modeled system, using measurements of the input $u(t) \in \mathbb{R}^p$ and output quantities of the system $y(t) \in \mathbb{R}^m$.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{2.23}$$

Using an artificial state space representation of the system having same matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{m\times n}$ and $D \in \mathbb{R}^{m\times p}$ as the real system, the outputs $y(t)$ and $\hat{y}(t)$ are compared. The quality of the estimate $\hat{x}$ is assessed by the difference between $y(t)$ and $\hat{y}(t)$ i.e. $y(t)-C\hat{x}(t)-Du(t)$. Using this feedback to the simulation, the observer is written as:

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + K(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= C\hat{x}(t) + Du(t) \end{aligned} \tag{2.24}$$

where K the observer gain is a $n \times m$ matrix. The difference between the true state and the estimated state is called state estimate error

$$\tilde{x}(t) = x(t) - \hat{x}(t) \tag{2.25}$$

From (2.23) and (2.24), we have the state estimate error dynamics:

$$\dot{\tilde{x}}(t) = (A - KC)\tilde{x}(t) \tag{2.26}$$

As all computation is done in discrete time, the representation of equation (2.24) in discrete time is

$$\begin{aligned} \hat{x}_{k+1} &= A\hat{x}_k + Bu_k + K(y_k - \hat{y}_k), \quad \hat{x}_{k=0} = \hat{x}_0 \\ \hat{y}_k &= C\hat{x}_k + Du_k \end{aligned} \tag{2.27}$$

and the state estimate error difference equation

$$\tilde{x}_{k+1} = (A - KC)\tilde{x}_k \tag{2.28}$$

The gain K has to chosen such that the state estimate error dynamics is stable; eigen values for $(A - KC)$ are on the left half plane for continuous time state-space models and inside the unit circle for discrete time state-space models.

2.5.1 Kalman filters

Kalman filter is an optimal, recursive algorithm for state estimation for linear systems with Gaussian noises. The system consists of linear motion and measurement models. The filter works by recursively predicting the object state using the motion model and correcting the state using measurements in the measurement model.
It is common to have noise in the sensors and modeling uncertainties that causes uncertainty in the estimates. The motion and measurement models of the system is made as

$$x_{k+1} = Ax_k + Bu_k + q_k \tag{2.29}$$

where q_k is the white Gaussian noise with co-variance Q_k.

$$y_k = Cx_k + Du_k + r_k \tag{2.30}$$

where r_k are white Gaussian noise with co-variance R_k.
The filter consists of two variables:

- Posterior mean $\hat{x}_{k|k}$ at time k given observations upto and including k.
- Posterior estimate covariance $P_{k|k}$.

The filter has two phases: "Predict" and "Update". In the predict step, the filter uses the state estimate from the current timestep to produce an estimate at next timestep $\hat{x}_{k+1|k}$. In the update step, the difference between measurement prediction and actual measurement information y_{k+1} is multiplied by the optimal Kalman gain K_{k+1} and combined with previous state prediction $\hat{x}_{k+1|k}$ to refine the posterior state estimate $\hat{x}_{k+1|k+1}$.

Predict:

- Predicted state estimate: $\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k$
- Predicted estimate covariance: $P_{k+1|k} = AP_{k|k}A^T + Q_k$

Update:

- Updated state estimate: $\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}(y_{k+1} - C\hat{x}_{k+1|k} - Du_k)$
- Updated estimate covariance: $P_{k+1|k+1} = P_{k+1|k} - K_{k+1}S_{k+1}K_{k+1}^T$
- Kalman gain: $K_{k+1} = P_{k+1|k}C^T S_{k+1}^{-1}$
- Innovation covariance: $S_{k+1} = CP_{k+1|k}C^T + R_{k+1}$

This requires a time-varying gain matrix K_{k+1} that minimizes the variance of the estimation error. The choice of K_{k+1} in the observer is a balance between sensitivity to the measurement disturbances and adaptability to the influence of the system disturbances [10].
Extensions of the Kalman filter for use in nonlinear systems exist, such as Extended Kalman Filter (EKF) or the Unscented Kalman filter (UKF). The nonlinear motion and measurement models are linearized around the prior and predicted values using first order taylor series expansion of the nonliner functions and then Kalman prediction and update steps respectively are applied.

2.5.1.1 Extended Kalman filter

Given a motion model of form

$$x_{k+1} = f(x_k, u_k) + q_k$$

it's first order taylor series approximation around $\hat{x}$ is given as

$$\approx f(\hat{x}_k) + f'(\hat{x}_k)(x_k - \hat{x}_k) + q_k$$

where $f'(\hat{x}_k)$ is the jacobian of f.
Similarly, given a measurement model of form

$$y_k = h(x_k, u_k) + r_k$$

it's first order taylor series approximation around $\hat{x}$ is given as

$$\approx h(\hat{x}_k, u_k) + h'(\hat{x}_k, u_k)(x_k - \hat{x}_k) + r_k$$

where $h'(\hat{x}_k)$ is the jacobian of h.
Since the rigid body motion equations are nonlinear in nature, the EKF is used as the estimation algorithm in this thesis.

EKF Prediction step: This step uses the state estimate at time k to provide an predicted estimate at time $k+1$ using the nonlinear motion model $f(x)$. Similarly the predicted estimate covariance is calculated according to

$$\begin{aligned} \hat{x}_{k+1|k} &= f(\hat{x}_{k|k}, u_k) \\ P_{k+1|k} &= f'(\hat{x}_{k|k})P_{k|k}f'(\hat{x}_{k|k})^T + Q_k \end{aligned} \tag{2.31}$$

EKF Update step: Using the predicted state estimate at time, a measurement y_{k+1} is used to update the prediction. This is achieved by using the difference between the actual measurement and the predicted measurement to correct the prediction. The Kalman gain K_{k+1} decides how much or how little to trust this measurement. Similarly the estimate covariance is also updated according to

$$\begin{aligned} \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_{k+1}(y_{k+1} - h(\hat{x}_{k+1|k}, u_k)) \\ P_{k+1|k+1} &= P_{k+1|k} - K_{k+1} S_{k+1} K_{k+1}^T \\ S_{k+1} &= h'(\hat{x}_{k+1|k}, u_k) P_{k+1|k} h'(\hat{x}_{k+1|k}, u_k)^T + R_{k+1} \\ K_{k+1} &= P_{k+1|k} h'(\hat{x}_{k+1|k}, u_k)^T S_{k+1}^{-1} \end{aligned} \tag{2.32}$$

where S_{k+1} is the innovation covariance.
Equations (2.31) and (2.32) form the basis of the estimator algorithm in this thesis. The motion and measurement model that are used to estimate the states of interest are described in section 3.4 and section 3.8.

2.6 Inertial Measurement Units (IMUs)

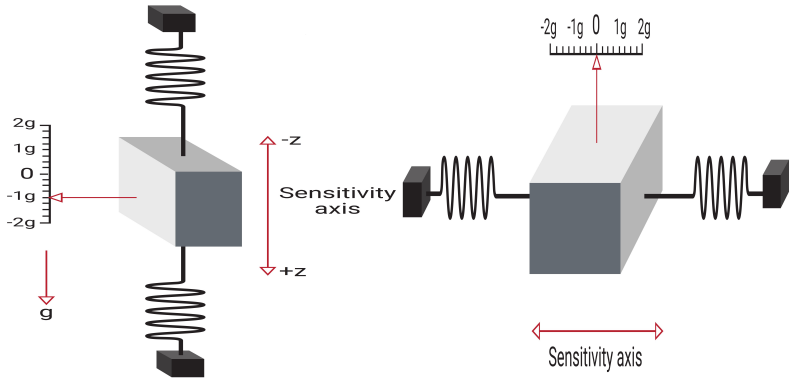
Inertial sensors are used to measure the motion of an object with respect to an inertial frame. An inertial system generally consists of accelerometers and gyroscopes combined to a single system, but can be used individually.
An individual inertial sensor can only sense a measurement along or about a single axis. A three-dimensional sensor consists of three individual inertial sensors mounted together in orthogonal directions called as a 3-axes inertial sensor. Similarly, a 6-axes system consists of a 3-axes accelerometer and a 3-axes gyroscope.

2.6.1 Accelerometers

Accelerometers are sensors that measure both the linear acceleration and local gravity vector, it can provide information both about the change in position and about the inclination of the sensor. A MEMS accelerometer is a mass suspended on a spring, this mass shifts from side to side when the accelerometer is subjected to linear acceleration, the amount of deflection is a function of the acceleration.
If the gravity vector is aligned with the sensitivity axis as in Figure 2.5a, the accelerometer measures the linear acceleration of the body as well as the acceleration due to gravity. If the body is not accelerating, then the accelerometer in case of configuration in 2.5a measures $-1\,g \approx -9.806\,\frac{m}{s^2}$.
A simplified accelerometer measurement model can be given as[11]:

$$y_a = R_n^b(a^n + g^n) + \delta^b + e^b \tag{2.33}$$

where R_n^b is the rotation matrix from the navigation frame n to the body frame b, a^n is the linear acceleration of the sensor (without gravity and its projection), g^n is the gravity vector, bias δ^b and noise e^b in the body frame.



(a) Vertical **(b)** Horizontal

Figure 2.5: Accelerometer model [1]

2.7 Roll and pitch angle of wheel hub

When negotiating a banked road or a elevated road, the wheel will have roll and pitch movements. These, parameters are also defined in the ISO 8855 standard [9]. Using the same wheel coordinate axes from section 2.3, the roll and pitch angles of the wheel are defined as in figure 2.6.

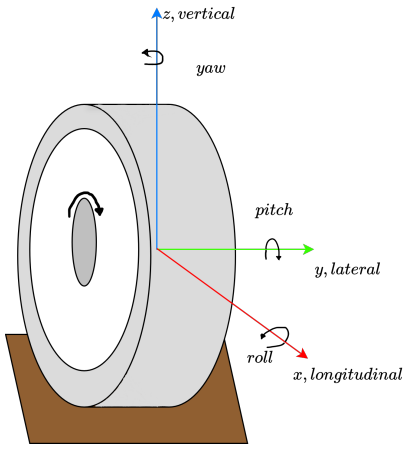


Figure 2.6: Degrees of freedom and ISO coordinate system of wheel

Estimating the roll and pitch angles of the wheel is an orientation estimation problem. Orientation can be described in numerous ways, rotation matrix representation uses 9 values to represent 3 degrees of freedom. Another method uses Euler angles, i.e., a representation using three consecutive rotations around orthogonal axes. However, Euler angle representation is not unique — different sets of Euler angles can correspond to the one rotation matrix, they also suffer from singularity or gimbal lock. Another way is to use a unit length quaternion representation.
In this thesis, quaternions and accelerometers are used for estimates of roll and pitch angle.

2.7.1 Quaternions

Orientation of objects is commonly expressed using unit quaternions. It was first introduced in [12]. A unit quaternion uses a 4DoF representation of the orientation

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} cos(\frac{1}{2}\delta) \\ \sin(\frac{1}{2}\delta) \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} \end{bmatrix} = \begin{bmatrix} cos(\frac{1}{2}\delta) \\ \sin(\frac{1}{2}\delta) e_v \end{bmatrix}, \qquad ||e_v||_2 = 1,\ ||q||_2 = 1$$

where δ is the rotation angle and e_v is a vector of unit length around which the body rotates.
In this thesis, the orientation of the wheel is a time varying quaternion vector which is estimated using the EKF algorithm described in section 2.5.1.1.
Accelerometer measurements are used in the update step of the EKF using the equation (2.33). The rotation matrix in that case can be represented using unit quaternions as follows: [13] (eq-257,page:29)

$$R_n^b = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix} \tag{2.34}$$

The rotation matrix $R_b^n = (R_n^b)^T$ can be represented equivalently in three-angle representation from Euler's rotation theorem, using a ZYX rotation sequence with Φ, Θ and Ψ as the rotations along x(roll), y(pitch) and z(yaw) respectively, the two equivalent rotation matrices are shown below.

$$\begin{aligned} R_b^n &= \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix} \\ &= \begin{bmatrix} cos\Psi cos\Theta & cos\Psi sin\Theta\, sin\Phi - cos\Phi sin\Psi & sin\Psi sin\Phi + cos\Psi cos\Phi sin\Theta \\ cos\Theta sin\Psi & cos\Psi cos\Phi + sin\Psi sin\Theta sin\Phi & cos\Phi sin\Psi sin\Theta - cos\Psi sin\Phi \\ -sin\Theta & cos\Theta sin\Phi & cos\Theta cos\Phi \end{bmatrix} \end{aligned} \tag{2.35}$$

Using algebraic manipulations, the roll and pitch angles of the wheel hub can be found from equation (2.35) as:

$$\begin{aligned} roll\, angle,\ \Phi &= atan2(R_{32}, R_{33}) \\ pitch\, angle,\ \Theta &= atan2(-R_{31}, \sqrt{R_{32}^2 + R_{33}^2}) \end{aligned} \tag{2.36}$$

3
Methods

This chapter deals with the simulation setup, implementation and methods used in IMU sensor location, wheel speed estimation, roll and pitch estimation.

3.1 Sensor configuration

Based on concepts explained and derived in sections 2.4 and 2.7, the following IMU sensor configuration is used in this thesis:

- IMUs with three axes i.e. accelerometers that measure accelerations in 3 orthogonal directions are used.
- One IMU is placed on the rim of wheel(s) to measure the tangential (a_{p_θ}), lateral (a_{p_y}) and radial (a_{p_r}) accelerations in $\frac{m}{s^2}$ units.
- On IMU is placed on the hub of the wheel, the sensor frame is non-rotating but follows the trajectory of the center of the wheel along the road surface. The axes of the sensor follow the ISO 8855 convention as described in section 2.3. In Modelon Impact it is not possible to measure acceleration in body coordinates, therefore acceleration in inertial coordinates is recorded. Whereas, in TruckMaker this IMU measures acceleration in body coordinate system.
- The wheel hub IMU follows the ISO 8855 ENU axes convention.
- The measurements from the IMU are generated at 1000 Hz sampling frequency.

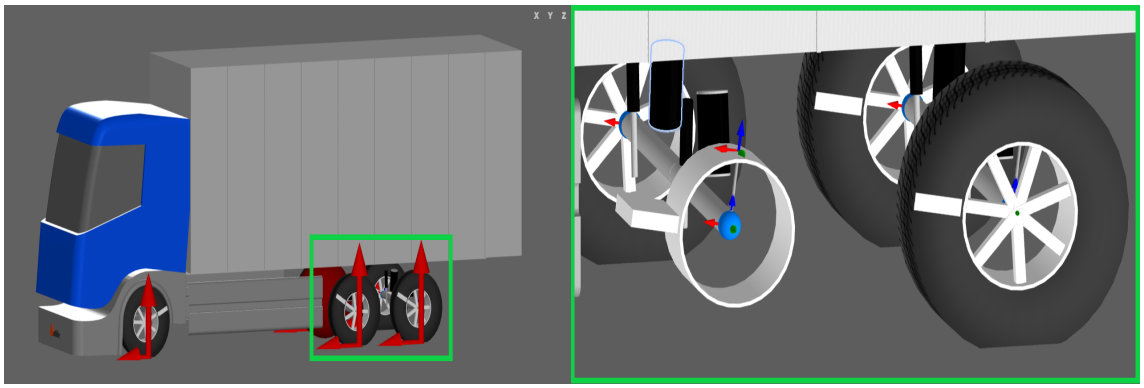
3.2 Simulation environment

In this thesis, a model-based design approach is implemented. Initially, the physics of the problem, the system dynamics, and the estimator models are investigated and examined within a simulation environment using ground truth signals. Subsequently, a sensor model is developed to emulate real sensor measurements and to facilitate the development of the estimator in a Model-In-the-Loop (MIL) environment. Future work includes testing the estimator with real measurement data.

3.2.1 Modelon Impact

Modelon Impact is an Open Web & Modelica-based simulation environment, which offers system-level modeling, simulation, optimization and analysis to drive engineering insight and decision-making. Modelica is a physical modeling language where the physics-based equations are used to make a model in acausal format. Model-

ica is suited for modeling of complex systems which is widely used for model-based system engineering[14].
Modelica-based acausal models are more trustable because they offer a higher fidelity representation of physical systems through equation-based, declarative descriptions. This approach ensures better accuracy, modularity, flexibility, and adaptability, making it easier to model, simulate, and validate complex physical systems compared to causal, algorithmic models. Therefore, in this thesis, in the first step simulations were performed in Modelon Impact environment to conduct investigation and verify the derived equations of motion of theory section.
A built-in truck model is used and a three-axes accelerometer with a rotating frame is mounted on the rim of the truck wheel(s), while an additional three-axis accelerometer with a non-rotating frame is mounted on the hub of the wheel(s) to measure accelerations (gravity+body acceleration or body acceleration). Additionally an absolute angular velocity sensor is mounted on the rim of the wheel(s) to measure the angular velocity of the wheel(s). The setup of sensor frames on wheel(s) can be visualized in figure 3.1.



(a) Truck in Modelon Impact **(b)** Sensor frames in Modelon Impact

Figure 3.1: Modelon Impact visualization

3.2.2 IPG TruckMaker

IPG TruckMaker is a specialized simulation software designed to provide comprehensive capabilities for the development and testing of trucks, buses, and other heavy-duty vehicles. It excels in simulating the complex dynamics of commercial vehicles. It also allows for precise modeling of real-world test scenarios in a virtual environment, enhancing the speed and flexibility of development processes.
With rapid virtual prototyping, TruckMaker provides a high-performance, real-time or faster than real-time capable vehicle model that enables the creation of virtual prototypes early in the development process. The software can integrate real-world data, enhancing the accuracy of simulations. This includes environmental conditions, road surfaces, and traffic scenarios. Real-time simulation is another standout feature of IPG TruckMaker. It supports hardware-in-the-loop (HIL) and software-in-the-loop (SIL) testing, allowing for immediate feedback and iterative development. Integrating IPG TruckMaker into existing development tools is easy due to its many

standard interfaces and support for formats like FMI. Additionally, with interfaces like Matlab, IPG TruckMaker can act as a central integration platform for various tools and systems.
Using TruckMaker, different test scenarios (road and driving scenarios) are modeled, and different IMU sensor configurations are arranged. The IPG Automotive Sweden team provided a customized TruckMaker version in which IMU(s) could be mounted on the wheel(s) of a truck. This truck model was then simulated under different driving and road conditions, to generate ground truth, sensor measurement data and other signals of interest. Along the rim mounted IMU, an additional IMU is placed at the center of the wheel (hub IMU). The sensor frame of the rim IMU is rotating with the wheel, whereas the hub IMU is a non rotating frame, following the trajectory of the wheel in 3D space.
Figure 3.2 shows an IMU mounted on the rim of a truck wheel in TruckMaker.

Figure 3.2: TruckMaker visualization

The rim IMU measures the acceleration (gravity+body acceleration) in the tangential, lateral, and radial directions. These accelerations are a function of the wheel's angular displacement (θ), angular velocity (ω), angular acceleration (α), radial distance (r_1) and acceleration (gravity+body acceleration) at the center of the wheel (a_{o_x} and a_{o_z}) as described in equation (2.22).
The measurements from the IMU are ideal values i.e. without noise, bias or axis misalignment. Figure 3.3 shows measurements from the rim accelerometer and gyroscope from IMU mounted on the rim of a truck moving in a straight line on a flat road accelerating for some time and then driving at constant velocity.

3.3 Development process

The process started with modeling a rigid body and deriving its equations of motion. These equations were initially verified using Modelon Impact. Following this, various scenarios were tested and the results from IPG TruckMaker were compared with

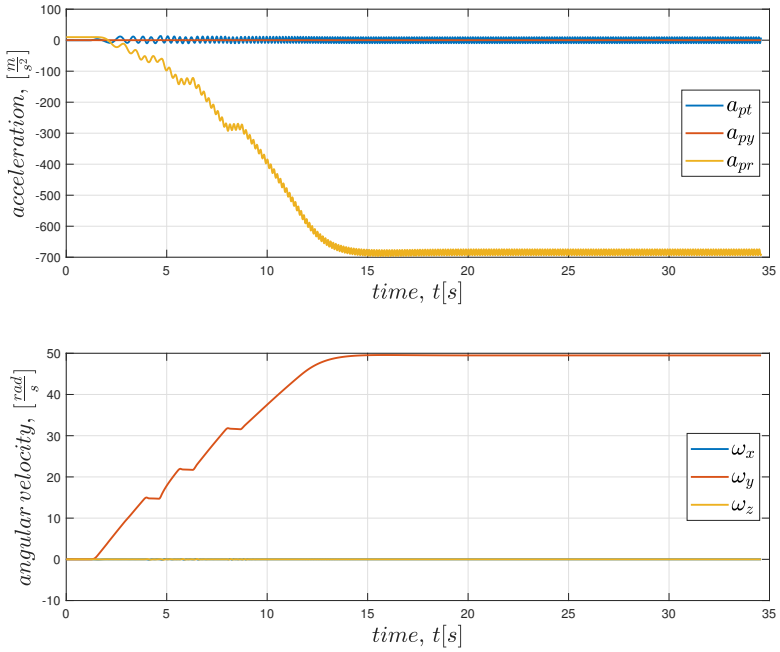


Figure 3.3: Rim IMU measurements on TruckMaker

those from Modelon Impact. After bench-marking the verification results, IPG TruckMaker was chosen for further simulation and development work due to its ease of modeling roads and handling complex scenarios.

3.4 EKF implementation

The simulation environments (Modelon Impact and IPG TruckMaker) generate measurements from the IMUs (rim and hub), these measurements are imported into MATLAB for processing. The Extended Kalman Filter (EKF) is implemented in MATLAB to estimate the desired state variables based on the sensor measurements.

The state variables chosen to be estimated in this thesis are: wheel angular displacement θ, wheel angular speed ω and the wheel angular acceleration α. $x(t) = \begin{bmatrix}\theta(t) & \omega(t) & \alpha(t)\end{bmatrix}^T$.

Estimating the wheel speed is of major importance as it is an input for algorithms like wheel slip control [15]. Providing accurate and fast angular speed estimates can greatly improve performance of these algorithms. Accurate estimates of absolute angular displacement θ is important in autonomous vehicle solutions, where knowing the exact position of wheel can be of great importance. Angular acceleration α is also of importance to vehicle motion algorithms.

3.4.1 Motion model

The motion model chosen for the observer design is a constant angular acceleration ($\dot{\alpha} = 0$) model. This model can be formulated in continuous time domain as follows:

$$\underbrace{\begin{bmatrix}\dot{\theta}(t)\\ \dot{\omega}(t)\\ \dot{\alpha}(t)\end{bmatrix}}_{\dot{x}(t)} = \underbrace{\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix}\theta(t)\\ \omega(t)\\ \alpha(t)\end{bmatrix}}_{x(t)} + \tilde{q}(t) \tag{3.1}$$

The model in equation (3.1) is a linear model and as described in section 2.5.1.1, the model has to be discretized to arrive at form $x_{k+1} = A_k x_k + q_k$.
For a continuous time linear system of form

$$\dot{x}(t) = \tilde{A}x(t) \tag{3.2}$$

it can be shown that

$$x(t+T) = exp(\tilde{A}T)x(t) \tag{3.3}$$

$$\implies A_k = exp(\tilde{A}T) \tag{3.4}$$

where T is the sample time. Using the Taylor series expansion of $exp(\tilde{A}T) = I + \tilde{A}T + \tilde{A}^2\frac{T^2}{2}$ and ignoring higher dimensions, $A_k = A$ can be derived as:

$$A = \begin{bmatrix}1 & T & \frac{T^2}{2}\\ 0 & 1 & T\\ 0 & 0 & 1\end{bmatrix} \tag{3.5}$$

Therefore the discrete time motion model can be described as:

$$\underbrace{\begin{bmatrix}\theta_k\\ \omega_k\\ \alpha_k\end{bmatrix}}_{x_{k+1}} = \underbrace{\begin{bmatrix}1 & T & \frac{T^2}{2}\\ 0 & 1 & T\\ 0 & 0 & 1\end{bmatrix}}_{A} \underbrace{\begin{bmatrix}\theta_{k-1}\\ \omega_{k-1}\\ \alpha_{k-1}\end{bmatrix}}_{x_k} + q_k \tag{3.6}$$

where q_k is the discrete time motion noise with zero mean and covariance Q_k $q_k \sim \mathcal{N}(0, Q_k)$, $Q_k \in \mathbb{R}^{3\times3}$ is the covariance matrix (positive semi-definite and symmetric).

3.4.2 Measurement model

The measurement model for the observer is based on equations derived using the rigid body kinematics of the wheel that relate the accelerometer measurements to the state variables. Therefore, the measurement model can be formulated as:

$$\underbrace{\begin{bmatrix}a_{p_r,k}\\ a_{p_\theta,k}\end{bmatrix}}_{y_k} = \underbrace{\begin{bmatrix}-r_1\omega_k^2 + a_{o_z,k}\,cos\theta_k + a_{o_x,k}\,sin\theta_k\\ r_1\alpha_k + a_{o_x,k}\,cos\theta_k - a_{o_z,k}\,sin\theta_k\end{bmatrix}}_{h(x_k,u_k)} + r_k \tag{3.7}$$

where $r_k \sim \mathcal{N}(0, R_k)$ is the measurement noise and $R_k \in \mathbb{R}^{2\times2}$ is the measurement noise covariance matrix. u_k is the input parameter, $a_{o_x,k}$ and $a_{o_z,k}$ are the input signals from the non rotating hub IMU.
The equation (3.7) is non-linear and $h(x_k, u_k)$ is the non-linear function that relates the state and input variables to the tangential and radial accelerations of the rim IMU.
rSince the EKF algorithm in section 2.5.1.1 uses the jacobian matrix $h'(x_k, u_k)$, it can obtained by partial differentiation of $h(x_k, u_k)$ w.r.to. the state vector x_k, $h'(x_k, u_k) = \frac{\partial h}{\partial x}$. Therefore, the jacobian matrix can be obtained as:

$$\begin{aligned} h'(x_k, u_k) &= \begin{bmatrix} \frac{\partial h}{\partial \theta} & \frac{\partial h}{\partial \omega} & \frac{\partial h}{\partial \alpha} \end{bmatrix} \\ &= \begin{bmatrix} a_{o_x,k}\, cos\theta_k - a_{o_z,k}\, sin\theta_k & -2r_1\omega_k & 0 \\ -a_{o_z,k}\, cos\theta_k - a_{o_x,k}\, sin\theta_k & 0 & r_1 \end{bmatrix} \end{aligned} \tag{3.8}$$

In equations (3.7) and (3.8), if the hub IMU measurements are not available, suitable approximations to a_{o_x} and a_{o_z} can be made, since the hub IMU sensor frame is non-rotating, $a_{o_z} \approx (g = 9.806\,\frac{m}{s^2})$ and $a_{o_x} \approx a_x$ the linear acceleration of the truck transformed into wheel-coordinate at the center of the wheel.

3.5 Accelerometer model

In order to replicate real world setup of accelerometers, it is essential to add noise and drift to TruckMaker's ideal sensor measurements. An accelerometer model that replicates a accelerometer was designed based on the Mathworks' implementation [16] of the model.
The model consists of following components:

3.5.1 Axes misalignment and bias

If $TrueMeas \in \mathbb{R}^{3\times N}$ $\left(\frac{m}{s^2}\right)$ is the set of true accelerometer measurements, then the measurements after axes misalignment and constant bias are added

$$b = \left(\begin{bmatrix} 1 & \frac{\eta_2}{100} & \frac{\eta_3}{100} \\ \frac{\eta_1}{100} & 1 & \frac{\eta_3}{100} \\ \frac{\eta_1}{100} & \frac{\eta_2}{100} & 1 \end{bmatrix} TrueMeas^T \right)^T + ConstantBias$$

where η_1, η_2, and η_3 are given by the first, second, and third elements of the $AxesMisalignment$ property of the sensor.

3.5.2 White Noise Drift

$$\xi_1 = w \left(\sqrt{\frac{SampleRate}{s}} \right) NoiseDensity$$

where w is the white noise with variance 1, $SampleRate$ (Hz) is sampling frequency of the measurements, $s = 1 \text{ or } 2$ as the scale variable and $NoiseDensity$ $\left(\frac{m/s^2}{\sqrt{Hz}}\right)$ is a property of the sensor.

3.5.3 Random Walk Drift

$$\xi_2(k) = \xi_2(k-1) + w(k)\left(\frac{RandomWalk}{\sqrt{\frac{SampleRate}{s}}}\right)$$

where $RandowmWalk\left(\frac{m/s^2}{\sqrt{(Hz)}}\right)$ is a property of the sensor.

3.5.4 Environmental Drift Noise

The environmental drift noise is modeled by multiplying the temperature difference from $25\,\deg C$ with the temperature bias.

$$\Delta_e = (Temperature - 25)TemperatureBias$$

where $Temperature$ $(degC)$ is the temperature of the sensor and $TemperatureBias$ $\left(\frac{m/s^2}{degC}\right)$ is a property of the sensor.

3.5.5 Quantization Model

Combining the above noise and drift parameters as follows:

$$c = b + \Delta_e + \xi_1 + \xi_2$$

The measurement is then saturated as:

$$d = \begin{cases} MeasurementRange, & \text{if } c > MeasurementRange. \\ -MeasurementRange, & \text{if } -c < MeasurementRange. \\ c, & \text{otherwise} \end{cases}$$

where $MeasurementRange$ $(\frac{m}{s^2})$ is a property of the sensor.
the final output from the accelerometer model $accReadings$ is given as

$$accReadings = Resolution\left(round\left(\frac{d}{Resolution}\right)\right)$$

where $Resolution$ $\left(\frac{m/s^2}{LSB}\right)$ is a property of the sensor.
Figure 3.4 shows the accelerometer model used in the simulation of sensor noise and bias.

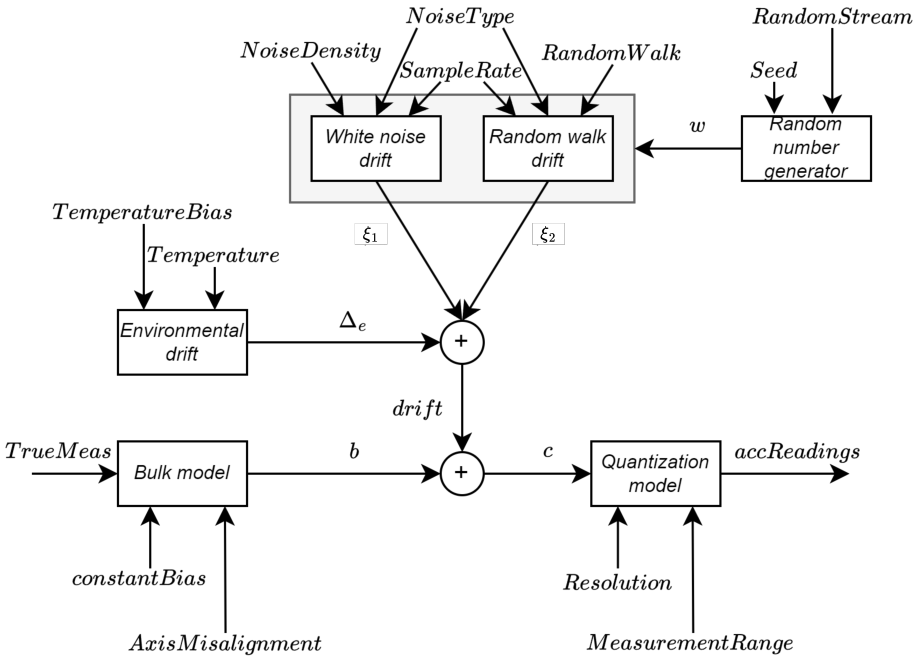


Figure 3.4: Accelerometer model

The values for the above sensor properties collected from the datasheet of an IMU manufacturer are given in table 3.1.

Property	Value	Unit
$MeasurementRange$	$100 \cdot 9.81$	$\frac{m}{s^2}$
$Resolution$	$\frac{100 \cdot 9.81}{2^{16}}$	$\frac{m/s^2}{LSB}$
$ConstantBias$	0	$\frac{m}{s^2}$
$NoiseDensity$	$15e^{-3} \cdot 9.81$	$\frac{m/s^2}{\sqrt{Hz}}$
$Temperature$	25	$degC$
$TemperatureBias$	0	$\frac{m/s^2}{degC}$
$SampleRate$	1000	Hz
$AxesMisalignment$	0	%
$NoiseType$	2	–
$RandomWalk$	$15e^{-3} \cdot 9.81$	$\frac{m/s^2}{\sqrt{Hz}}$

Table 3.1: Accelerometer parameters

3.6 Sensor position at initialization

For the observer to estimate the states accurately, it is important to know the true value of states at the beginning of the simulation i.e., initial state at time t_0, $x(t_0)$. The states of angular velocity ω and acceleration α are assumed to $\omega_0 = 0$ and $\alpha_0 = 0$, whereas the value of θ_0 can be calculated as follows

From the equation (2.22), at time t_0, $a_{o_x,0} = 0$, $\omega_0 = 0$, $\alpha_0 = 0$:

$$\begin{aligned} a_{p_\theta,0} &= -a_{o_z,0}\, sin\theta_0 \\ a_{p_r,0} &= a_{o_z,0}\, cos\theta_0 \end{aligned} \tag{3.9}$$

where a_{o_z} is the acceleration from the hub IMU in the body coordinates. Therefore, the sensor position at initialization can be estimated as

$$\theta_0 = \arctan 2(-a_{p_\theta,0},\, a_{p_r,0}) \tag{3.10}$$

The equation (3.10) is only valid for sensors with no noise. Therefore for noisy sensor measurements following steps are used:

1. Collect 0.5 seconds of rim IMU sensor measurements in the beginning sequence, assuming that the vehicle is stationary of that period of time.
2. Two low pass butterworth filters are used to filter noisy measurements of the cropped radial and tangential accelerations from step 1.
3. A set of initial positions $\epsilon_{\mathbf{0}}$ is calculated for the filtered measurements from step 2 using equation (3.10).
4. The median of set $\epsilon_{\mathbf{0}}$ is used as the sensor start position.

$$\theta_0 = median(\epsilon_{\mathbf{0}})$$

3.7 Covariance matrices

The observer has three matrices to be selected for effective performance, Q_k, R_k and P_0.

3.7.1 The measurement noise covariance matrix R

According to the *central limit theorem*, given a sequence of i.i.d. random variables with $\mathbb{E}[X_i] = \mu$ and $\mathrm{Var}[X_i]{=}\sigma^2$. As the sequence approaches infinity, the random variables converge to a normal distribution $\sim \mathcal{N}(0, \sigma^2)$[17]. The units of the diagonal elements in the measurement noise covariance is always the squared of the units of a measured variables that they represent. The off-diagonal elements have the units of product of the corresponding variables units.

To determine the covariance of the radial and tangential acceleration values, sensor measurement is taken when the truck is stationary and on a flat road. Figure 3.5 shows the distribution of the sensor measurement when truck is stationary and on flat surface. The the mean and covariance values for these measurements are in table 3.2.

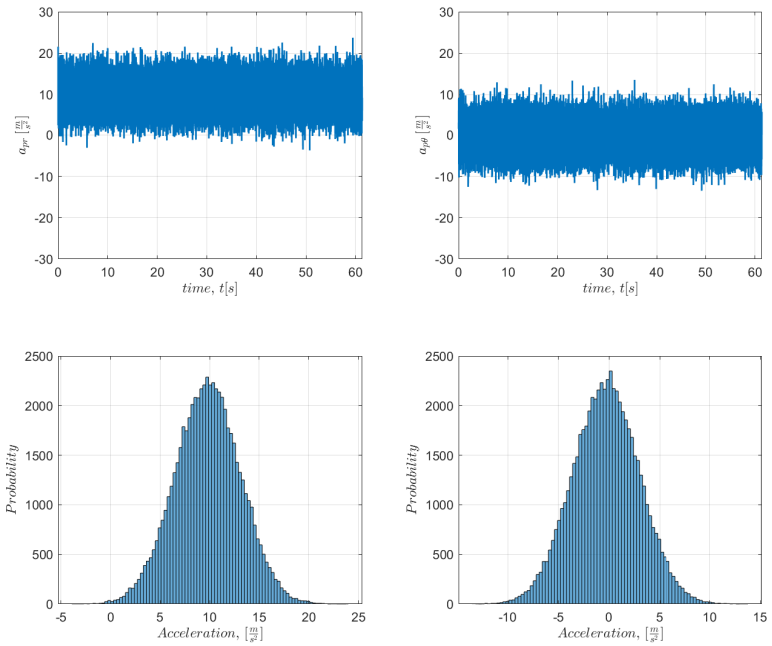


Figure 3.5: Rim IMU measurements when truck is stationary

Signal	Mean $[\frac{m}{s^2}]$	Covariance
Radial acceleration a_{p_r}	9.7909	10.8250
Tangential acceleration a_{p_θ}	-0.3013	10.8493

Table 3.2: Mean and covariance values of stationary sensor measurements

Therefore the noise covariance matrix is

$$R = \begin{bmatrix} 10.8250 & 0 \\ 0 & 10.8493 \end{bmatrix}$$

3.7.2 The prior covariance matrix P

The prior state covariance matrix $P_0 \in \mathbb{R}^{3\times3}$ defines the uncertainty of the state variables at t_0, since the sensor position θ_0 is known from section 3.6 and that $\omega_0 = \alpha_0 = 0$, the prior uncertainty P_0 is low. The units of the diagonal elements in the prior covariance is always the squared of the units of state variables that they represent. The off-diagonal elements have the units of product of the corresponding variables units.

$$P_0 = \begin{bmatrix} 1e^{-2} & 0 & 0 \\ 0 & 1e^{-2} & 0 \\ 0 & 0 & 1e^{-2} \end{bmatrix}$$

3.7.3 The motion noise covariance matrix Q

The motion noise covariance matrix Q describes how much the predicted state deviates from the previous estimate i.e., $p(\hat{x}_{k+1}|\hat{x}_k)$. As trucks and cars in general tend to move under constant acceleration and constant velocity conditions for a major portion of time, the values of the Q matrix tend to be small. The units of the diagonal elements in the motion noise covariance is always the squared of the units of state variables that they represent. The off-diagonal elements have the units of product of the corresponding variables units.
Therefore, the Q matrix chosen for the estimation in this thesis through experimentation are split based on the speed of the wheel. If the wheels rotates with with speed $\omega \geq 1.5\, rad/s$, the Q matrix is chosen as:

$$Q = \begin{bmatrix} 4e^{-4} & 0 & 0 \\ 0 & 4e^{-4} & 0 \\ 0 & 0 & 9e^{-3} \end{bmatrix}$$

and if the wheel speed is $\omega < 1.5\, rad/s$:

$$Q = \begin{bmatrix} 1e^{-6} & 0 & 0 \\ 0 & 1e^{-6} & 0 \\ 0 & 0 & 1e^{-6} \end{bmatrix}$$

3.8 Roll and pitch angle estimation

In section 2.7 and in equation (2.33), we relate the hub accelerometer measurements to the roll and pitch angles of the road. The relation in (2.33) will only be valid if the a^n acceleration vector is available. As accelerometers measure both the linear acceleration and the local gravity vector, measuring or estimating acceleration of a vehicle accurately without the influence of gravity is a challenging task. TruckMaker provides acceleration vector of the truck in ISO 8855 standard definition, another way to accurately measure the linear acceleration of the vehicle could be to use ego motion estimates of acceleration of the truck obtained using radar measurements.

3.8.1 Orientation estimation using true acceleration values

Using the TruckMaker's linear acceleration $a^n = a_{veh} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T$ values, the equation (2.33) can be formulated as equation (3.11):

3.8.1.1 Relation between quaternion and acceleration

$$\begin{bmatrix} a_{o_x} \\ a_{o_y} \\ a_{o_z} \end{bmatrix} = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix} \begin{bmatrix} a_x+0 \\ a_y+0 \\ a_z+g \end{bmatrix} \tag{3.11}$$

An EKF observer which estimates the quaternion $q_k = \begin{bmatrix} q_{0_k} & q_{1_k} & q_{2_k} & q_{3_k} \end{bmatrix}$ over time has the following motion and measurement models:

3.8.1.2 Motion model for orientation estimation

Usual orientation estimation algorithms like [18] and [19] use gyroscope measurements of the angular velocity as input signal for quaternion prediction, in absence of gyroscope measurements a simple random walk motion model is used:

$$\begin{bmatrix} q_{0_{k+1}} \\ q_{1_{k+1}} \\ q_{2_{k+1}} \\ q_{3_{k+1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} q_{0_k} \\ q_{1_k} \\ q_{2_k} \\ q_{3_k} \end{bmatrix} + v_k \tag{3.12}$$

where $v_k \sim \mathcal{N}(0, M_k)$ is white noise with covariance matrix $M_k \in \mathbb{R}^{4\times4}$.

3.8.1.3 Measurement model for orientation estimation

Using the equation (3.11), the measurement model that is used to update the predicted state vector can be formulated as:

$$\begin{bmatrix} a_{o_x,k} \\ a_{o_y,k} \\ a_{o_z,k} \end{bmatrix} = \underbrace{\begin{bmatrix} 1-2(q_{2_k}^2+q_{3_k}^2) & 2(q_{1_k}q_{2_k}-q_{0_k}q_{3_k}) & 2(q_{1_k}q_{3_k}+q_{0_k}q_{2_k}) \\ 2(q_{1_k}q_{2_k}+q_{0_k}q_{3_k}) & 1-2(q_{1_k}^2+q_{3_k}^2) & 2(q_{2_k}q_{3_k}-q_{0_k}q_{1_k}) \\ 2(q_{1_k}q_{3_k}-q_{0_k}q_{2_k}) & 2(q_{2_k}q_{3_k}+q_{0_k}q_{1_k}) & 1-2(q_{1_k}^2+q_{2_k}^2) \end{bmatrix} \begin{bmatrix} a_{x,k}+0 \\ a_{y,k}+0 \\ a_{z,k}+g \end{bmatrix}}_{h(q_k, a_{n_k})} + w_k \tag{3.13}$$

where $w_k \sim \mathcal{N}(0, R_k)$ is white noise with covariance matrix $R_k \in \mathbb{R}^{3\times3}$.
As the function $h(q, a_n)$ is nonlinear in nature, its jacobian matrix $h'(q_k, a_n)$ is given as follows:

$$h'(q, a_n) = \begin{bmatrix} \frac{\partial h}{\partial q_0} & \frac{\partial h}{\partial q_1} & \frac{\partial h}{\partial q_2} & \frac{\partial h}{\partial q_3} \end{bmatrix} \cdot \begin{bmatrix} a_x+0 \\ a_y+0 \\ a_z+g \end{bmatrix} \tag{3.14}$$

where the partial derivatives in equation (3.14) are given as follows:

$$\frac{\partial h}{\partial q_0} = \begin{bmatrix} 0 & 2q_3 & -2q_2 \\ -2q_3 & 0 & 2q_1 \\ 2q_2 & -2q_1 & 0 \end{bmatrix} \quad \frac{\partial h}{\partial q_1} = \begin{bmatrix} 0 & 2q_2 & 2q_3 \\ 2q_2 & -4q_1 & 2q_0 \\ 2q_3 & -2q_0 & -4q_1 \end{bmatrix}$$

$$\frac{\partial h}{\partial q_2} = \begin{bmatrix} -4q_2 & 2q_1 & -2q_0 \\ 2q_1 & 0 & 2q_3 \\ 2q_0 & 2q_3 & -4q_2 \end{bmatrix} \quad \frac{\partial h}{\partial q_3} = \begin{bmatrix} -4q_3 & 2q_0 & 2q_1 \\ -2q_0 & -4q_3 & 2q_2 \\ 2q_1 & 2q_2 & 0 \end{bmatrix}$$

Using the estimated quaternion vectors, the orientation of the truck wheel can be estimated using equations (2.35) and (2.36):

$$\begin{aligned} R_b^n &= \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix} \\ &= \begin{bmatrix} cos\Psi cos\Theta & cos\Psi sin\Theta\, sin\Phi - cos\Phi sin\Psi & sin\Psi sin\Phi + cos\Psi cos\Phi sin\Theta \\ cos\Theta sin\Psi & cos\Psi cos\Phi + sin\Psi sin\Theta sin\Phi & cos\Phi sin\Psi sin\Theta - cos\Psi sin\Phi \\ -sin\Theta & cos\Theta sin\Phi & cos\Theta cos\Phi \end{bmatrix} \end{aligned} \tag{3.15}$$

$$\begin{aligned} roll\,angle,\ \Phi &= atan2(R_{32}, R_{33}) \\ pitch\,angle,\ \Theta &= atan2(-R_{31}, \sqrt{R_{32}^2 + R_{33}^2}) \end{aligned} \tag{3.16}$$

3.9 Closed loop simulation

A wheel slip controller is developed in TruckMaker for Simulink. The controller was parameterized and integrated into TruckMaker program as a plugin model. This controller takes vehicle longitudinal speed, longitudinal acceleration and wheel speed as inputs and applies a sliding mode slip control to the brake system. The sliding mode algorithm, developed for speed-dependent wheel slip control is presented in [15], where the braking torque M_b is given as:

$$M_b = r_b F_x + \frac{J v_x}{r_r}\left(\frac{\omega\,\dot{v}_x\,r_r}{v_x^2} + \dot{\sigma}_{d_x} - k\frac{s}{\|s\| + \delta} - \phi\,s\right) \tag{3.17}$$

where:

- F_x is the longitudinal tire force
- J is wheel moment of inertia
- r_r is effective rolling radius
- r_b is the loaded radius at which braking force acts
- k is the sliding mode controller gain
- σ_{d_x} is the slip limit
- ω is the wheel speed
- $\dot{v}_x$ is the derivative of the longitudinal velocity at the wheel coordinate system
- δ is used to define a boundary layer
- ϕ is the proportional term
- $s = \sigma_x - \sigma_{d_x}$ is the sliding surface, with σ_x is the longitudinal wheel slip

$$\sigma_x = \frac{\omega\,r_r}{v_x} - 1 \tag{3.18}$$

The standard values considered in equation (3.17) are:

$$\begin{aligned} \delta &= 1e^{-3} \\ k &= 8 \\ \phi &= 10 \\ r_r &= 0.505\,m \\ J &= 10\,kgm^2 \end{aligned}$$

The wheel speed estimator designed provides the slip control model with wheel speed values ($\hat{x} = \omega$). The slip controller provides the truck model in TruckMaker with braking torque values (u_{torque}) and in response the truck model provides the wheel speed estimator with IMU measurements (y and u_1) and other signals such as F_x, r_b, σ_{d_x} and $\dot{v}_x$ as input (u_2) to the controller. A block diagram of the closed loop simulation is shown in figure 3.6. Simulation of a braking scenario where the slip

control algorithm is active is used to evaluate the response, stability, and robustness of the wheel speed estimator at different sensor sampling intervals.

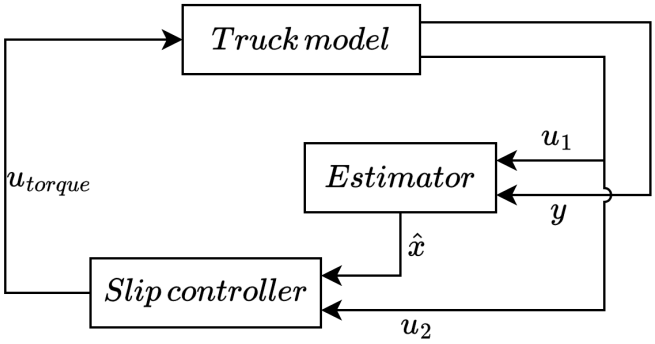


Figure 3.6: Closed loop simulation model

3.10 Test Scenarios

The truck model with IMU on the rim and hub of a wheel is simulated under different driving conditions in the simulation environments described in section 3.2, the conditions are follows:

Maneuver	Speed of truck	Additional parameter
Truck traveling in straight line	$90\,km/h$ from $0\,km/h$ standstill	-
Making a lane change	$90\,km/h$	-
Truck going up and down an elevated surface	$90\,km/h$	Gradient:21% uphill (15^o), 26% downhill (25^o)
"Figure-8" with road bank	$25\,km/h$	12^o bank angle
Slalom	$40\,km/h$	Steering angle: $\pm 50^o$
Very slow forward motion (creep)	$0.3\,m/s$	-
Potholes and bumpy surfaces	$90\,km/h$	-
Slip Control active	$80\,km/h$	Friction: 0.3, Slip limit:$-1e^{-1}$

Table 3.3: Test matrix

4

Results

This chapter will present a detailed discussion on the results related to the verification of rigid body kinematics, as well as the estimation of angular displacement, angular velocity, angular acceleration, and the determination of roll and slope angles.

4.1 Verification using Simulated Data:

The equations derived using rigid body kinematics as mentioned in equation (2.21) are verified using data generated from Modelon Impact in few driving scenarios. The verification involved computing the right hand side (RHS) of the rigid body equation and comparing it with the left hand side (LHS) of the equation (2.21). The left hand side are measurements from the rim IMU. Due to flexibility of TruckMaker in designing complex scenarios and model in loop development, verification of the same rigid body equations was carried out in TruckMaker. By verifying the equation (2.21), its performance could be evaluated and to see how well it performs in the estimation stage.

4.1.1 Verification of tangential acceleration

The tangential acceleration in the sensor frame from equation (2.21) is given as

$$a_{p_\theta} = a_{o_x}\, cos\theta - a_{o_z} sin\theta + \frac{r_1\, \omega_x^2\, sin2\theta}{2} + r_1 \omega_x \omega_z\, cos2\theta - \frac{r_1\, \omega_z^2\, sin2\theta}{2} + r_1 \alpha_y \quad (4.1)$$

The values for the parameters θ, ω_x, ω_z, a_{o_x}, a_{o_z}, α_y and a_{p_θ} are recorded IMU data at the rim and the wheel hub and other sensor measurements in Modelon Impact and TruckMaker. Then the right side of equation (4.1) is computed and compared to the left hand side of the equation.

The figure 4.1 shows the tangential acceleration of the rim IMU verified with truck traveling in a straight line using Modelon Impact

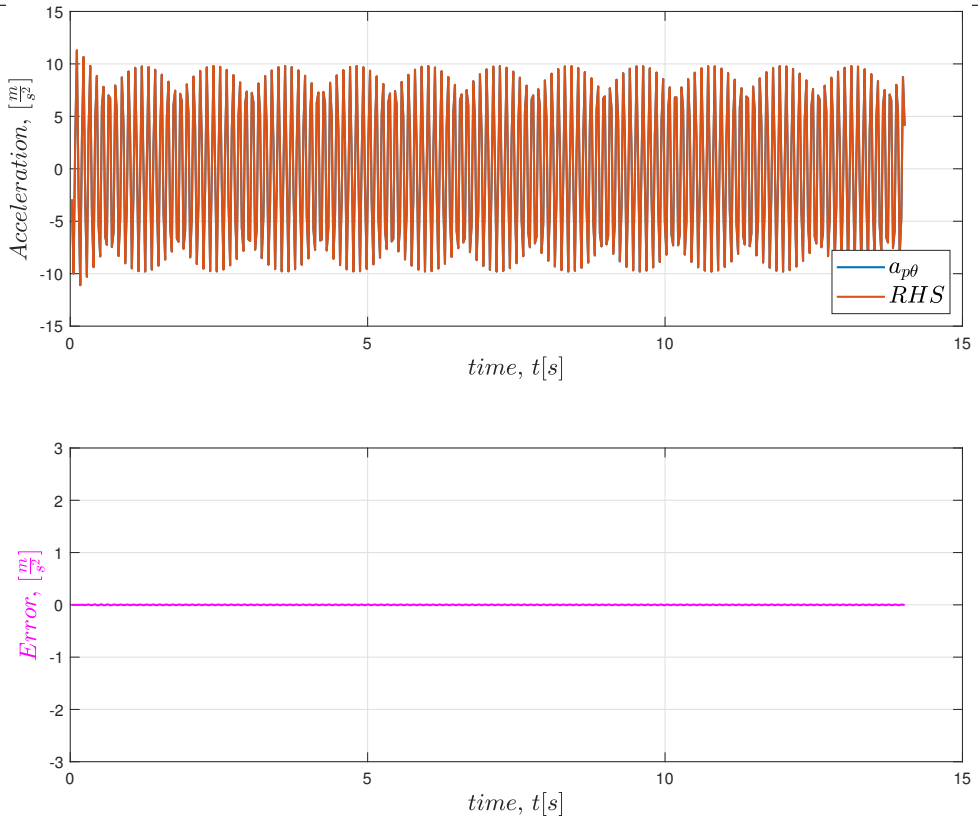


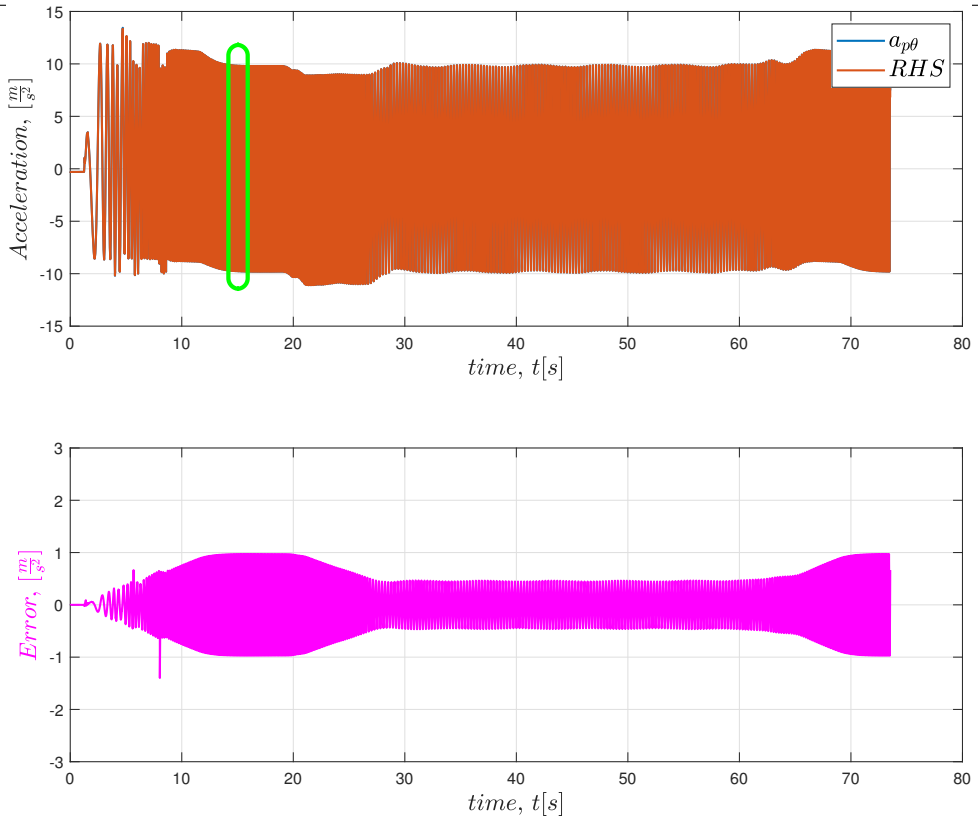
Figure 4.1: Verification of rim IMU tangential acceleration using Modelon Impact in straight maneuver

The error between the two sides of the equation is quite small and is mostly due to numerical precision of sensor measurements. The table 4.1 shows the error that was observed during verification of equation (4.1) in different scenarios in Modelon Impact. Plots for these scenarios are in Appendix A.

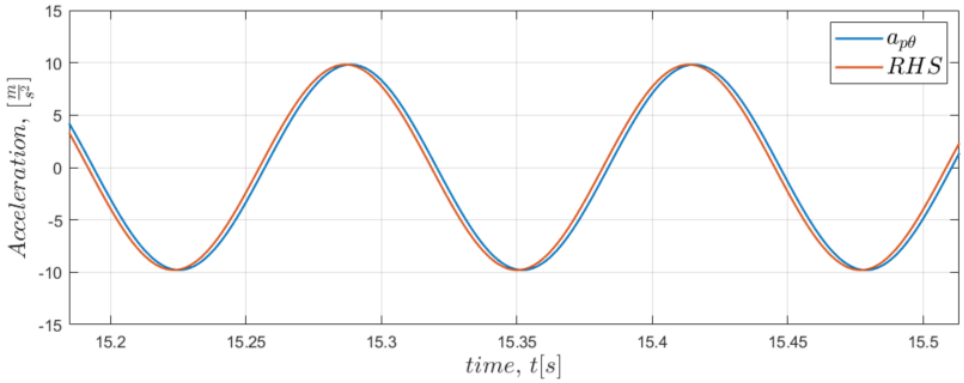
Maneuver	mean(|Error|) $[\frac{m}{s^2}]$	RMSE $[\frac{m}{s^2}]$	max(|Error|) $[\frac{m}{s^2}]$
Straight	0.005	0.005	0.009
Double lane change	0.060	0.124	0.514
Steer ramp	0.447	0.683	2.129

Table 4.1: Verification error in tangential acceleration using Modelon Impact

Similarly, verification of equation (4.1) was performed for TruckMaker data under different scenarios.
Figure 4.2a shows the verification results for tangential acceleration measurements for a truck driving in slalom maneuver. The error for this and other maneuvers are shown in table 4.2.
The error is mainly due to unmodelled dynamics in the equation (4.1) like α_z, the rate of road wheel angle changes. Plots for other scenarios are in Appendix A. The table 4.2 shows the error that was observed during verification of equation (4.1) in different scenarios in TruckMaker.



(a) Verification of equation using TruckMaker



(b) Magnification of rim IMU tangential acceleration

Figure 4.2: Verification of rim IMU tangential acceleration using TruckMaker in slalom maneuver

Maneuver	mean(|Error|) $[\frac{m}{s^2}]$	RMSE $[\frac{m}{s^2}]$	max(|Error|) $[\frac{m}{s^2}]$
Straight	0.449	0.594	0.972
Creep	0.007	0.008	0.062
Slope up-down	0.590	0.678	1.840
Slalom	0.372	0.445	0.972
Figure 8 with bank	0.229	0.273	0.933

Table 4.2: Verification error in tangential acceleration using TruckMaker

4.1.2 Verification of radial acceleration

The radial acceleration in the sensor frame from equation (2.21) is given as

$$a_{p_r} = a_{o_x} sin\theta + a_{o_z} cos\theta - \frac{r_1 \omega_x^2 cos2\theta}{2} - \frac{r_1 \omega_x^2}{2} + r_1 \omega_x \omega_z sin2\theta - r_1 \omega_y^2 + \frac{r_1 \omega_z^2 cos2\theta}{2} - \frac{r_1 \omega_z^2}{2} \tag{4.2}$$

The values for the parameters θ, ω_x, ω_y, ω_z, a_{o_x}, a_{o_z} and a_{p_r} are recorded from the IMUs and other sensor measurements in Modelon Impact and TruckMaker. Then the right side of equation (4.2) is computed and compared to the left hand side of the measurement.
The figure 4.3 shows the radial acceleration of the rim IMU verified with truck traveling in a straight line using Modelon Impact.

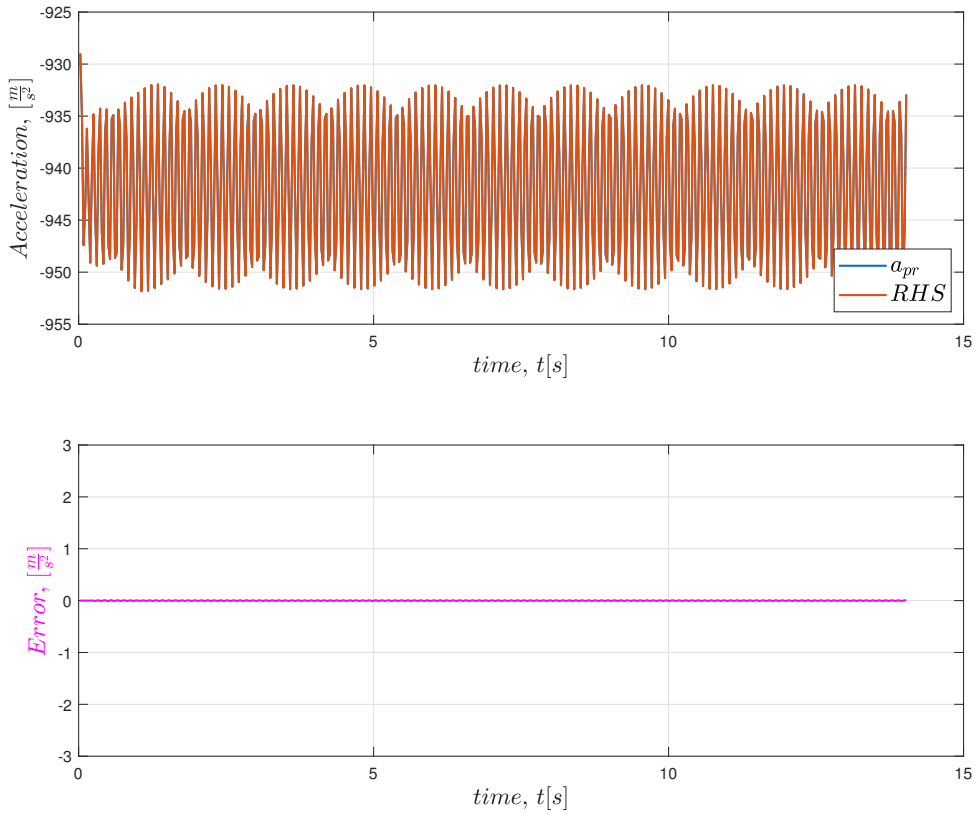
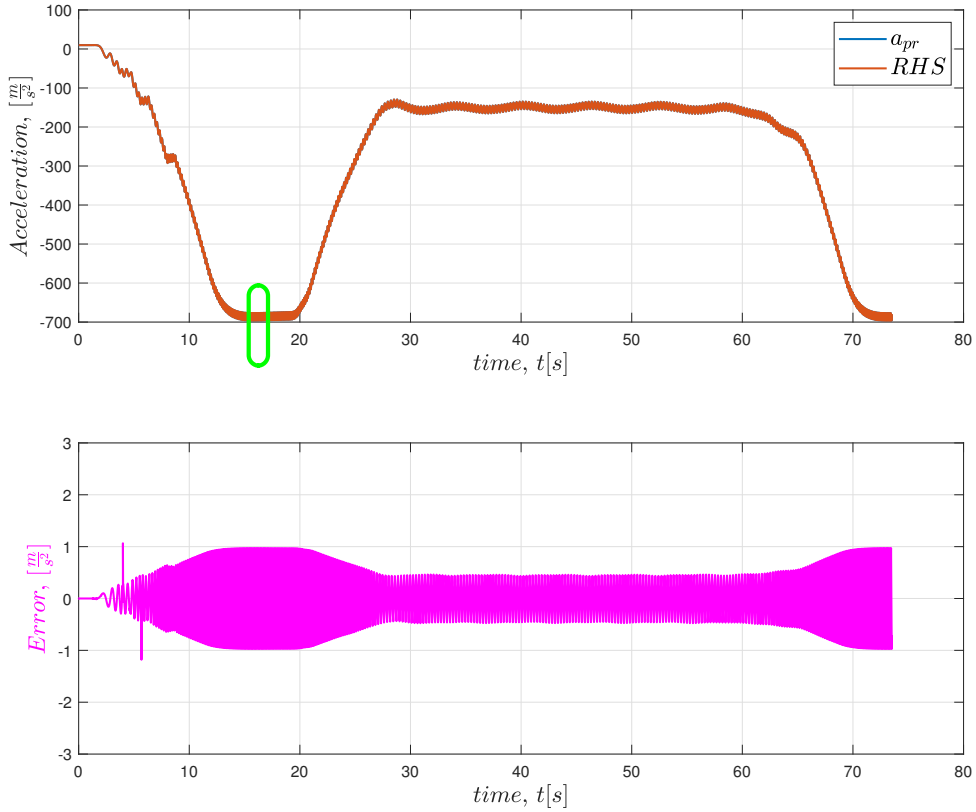


Figure 4.3: Verification of rim IMU radial acceleration using Modelon Impact in straight maneuver

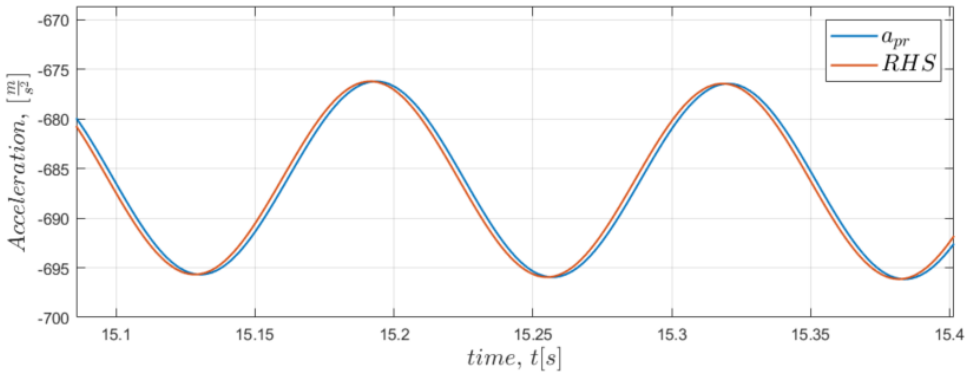
Similar to the tangential acceleration, the error between the two sides of the equation is quite small and is mostly due to numerical precision of sensor measurements. The table 4.3 shows the error that was observed during verification of equation (4.2) in different scenarios in Modelon Impact. Plots for these scenarios are also in Appendix A.
Similarly, verification of equation (4.2) was performed for TruckMaker data under different scenarios. Figure 4.4a shows the verification results for radial acceleration measurements for a truck driving in slalom maneuver. The error for this and other

maneuvers are shown in table 4.4. Plots for other scenarios are also presented in Appendix A.

The error is mainly due to unmodelled dynamics in the equation (4.2) like α_z, the rate of steering wheel angle changes.



(a) Verification of equation using TruckMaker



(b) Magnification of rim IMU radial acceleration

Figure 4.4: Verification of rim IMU radial acceleration using TruckMaker in slalom maneuver

Maneuver	mean(|Error|) $[\frac{m}{s^2}]$	RMSE $[\frac{m}{s^2}]$	max(|Error|) $[\frac{m}{s^2}]$
Straight	0.0050	0.0055	0.0090
Double lane change	0.0602	0.1235	0.5149
Steer ramp	0.4475	0.6827	2.1117

Table 4.3: Verification error in radial acceleration using Modelon Impact

Maneuver	mean(|Error|) $[\frac{m}{s^2}]$	RMSE $[\frac{m}{s^2}]$	max(|Error|) $[\frac{m}{s^2}]$
Straight	0.499	0.594	1.065
Creep	0.007	0.008	0.106
Slope up-down	0.590	0.678	1.840
Slalom	0.372	0.445	1.065
Figure 8 with bank	0.229	0.273	1.448
Active slip control(open loop)	0.266	0.656	3.569

Table 4.4: Verification error in radial acceleration using TruckMaker

4.2 State estimation using simulated data

4.2.1 Estimation of angular speed

After verifying and validating the equation (2.21), the simplified equation (2.22) is used for the estimation algorithm described in 3.4. The developed EKF state observer, is utilized to estimate the states of angular displacement ($\theta\,[rad]$), angular velocity ($\omega\,[rad/s]$), and angular acceleration ($\alpha\,[rad/s^2]$). Simulation data is generated under various driving scenarios using TruckMaker, to the ideal sensor data generated in these scenarios, noise is added according to the accelerometer model described in section 3.4. The prior and tuning parameters used for the EKF in all scenarios are in section 3.6 and 3.7 respectively. The results from the estimation are presented here.

Estimating the angular speed with high accuracy and fidelity compared to existing solutions under various driving scenarios is of particular interest in this thesis.

Figure (4.5) shows the estimation of angular speed estimation using the rim and hub IMUs of the front-left wheel of the truck performing a slalom maneuver. The estimation is compared to the true angular speed of the same wheel and estimation error is also shown.

In this maneuver, the truck first accelerates to $90\,km/h$ from standstill in a straight line, followed by a sequence of turns around cones in alternate directions with constant vehicle velocity of $40\,km/h$ before accelerating back to $90\,km/h$. This simulation provides diverse driving patterns that includes significant longitudinal and lateral accelerations, steering wheel angle changes, variations in longitudinal speed and brake torque. The longitudinal speed, steering angle, and brake pedal status of the vehicle are also plotted and are shown in figure 4.6 for the slalom maneuver.

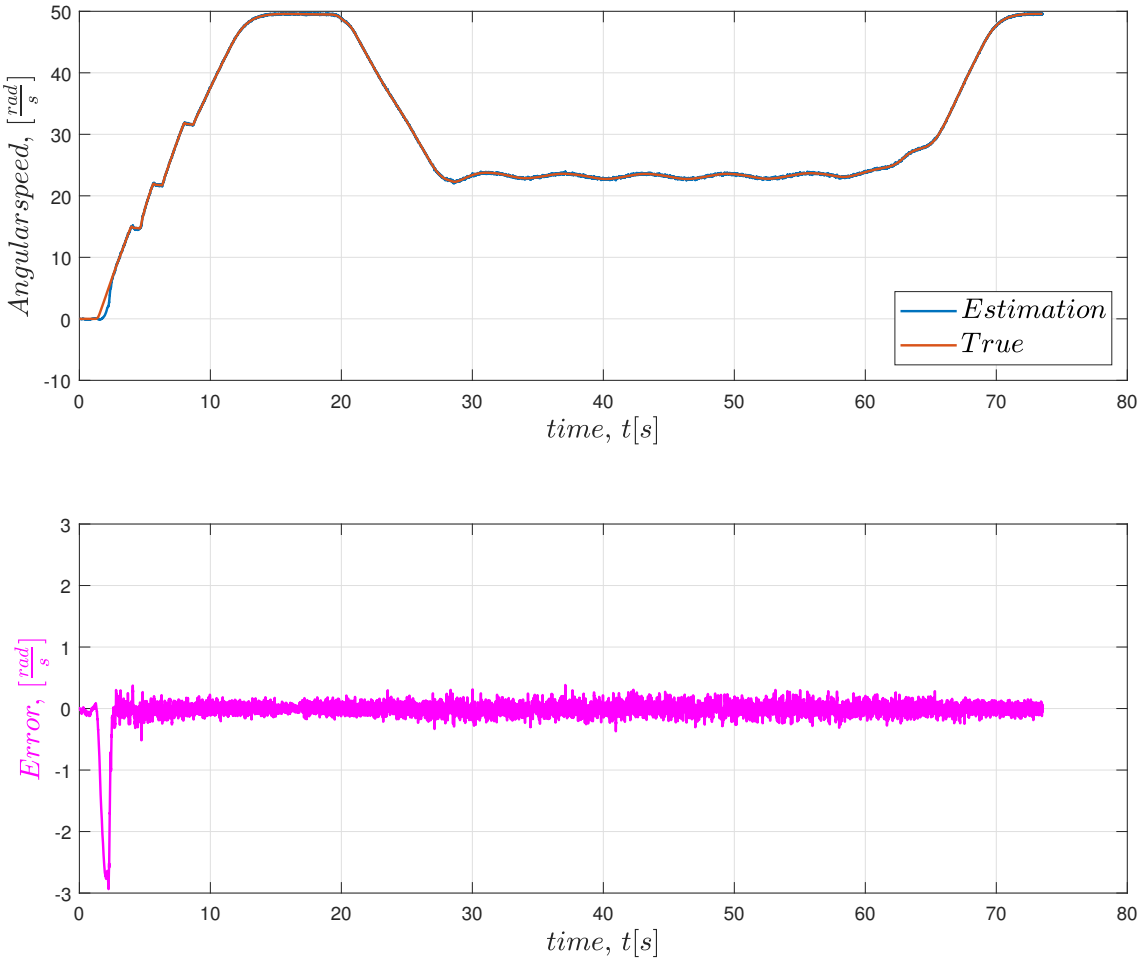


Figure 4.5: Angular speed estimation versus true in Slalom maneuver

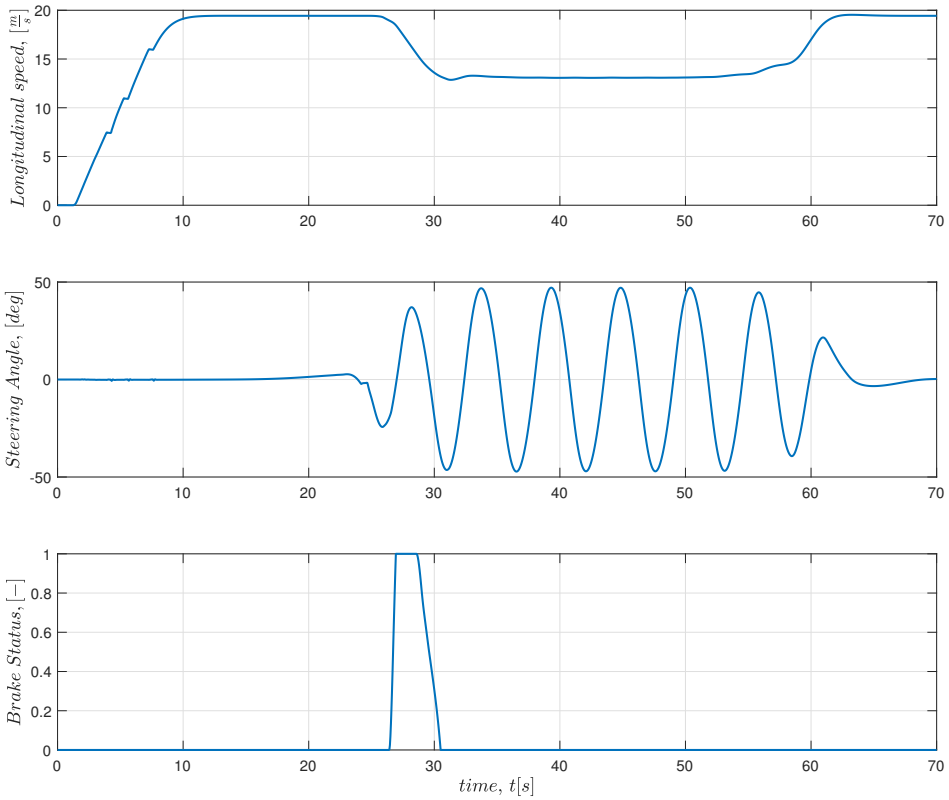


Figure 4.6: Longitudinal speed, steering angle and brake status in Slalom maneuver

The figure 4.5 shows a very good estimation of ω, with the estimator performing well with $|error| < 0.5\,rad/s$ for most parts of the maneuver. The larger error at the beginning of the measurement is due to large transients and a compromise between the EKF being very adaptive to transients or being an aggressive smoother.
The table 4.5 shows the estimation error for this and other maneuvers.

Maneuver	mean(|Error|) $[\frac{rad}{s}]$	RMSE $[\frac{rad}{s}]$	max(|Error|) $[\frac{rad}{s}]$
Slalom	0.075	0.262	2.946
Creep	0.034	0.061	0.618
Slope up-down	0.051	0.217	2.848
Straight	0.954	0.357	2.850
Figure 8 with bank	0.093	0.247	2.846
Active slip control(open loop)	0.132	0.283	3.127

Table 4.5: Error in angular speed estimation using TruckMaker

When there is no torque applied on the wheel, the angular speed ω can be used to measure the linear velocity v_x of the wheel using the relation $v_x = r_r\omega$. A $1\,km/h = 0.278\,m/s$ change in linear velocity of a wheel with $r_r = 0.505\,m$, translates to

$0.55\,rad/s$ change in angular speed. Therefore, $\omega_{lim} = \omega_{truth} \pm 0.550\,rad/s$ can be considered as the upper and lower limits of estimation.
Figure 4.7 shows the estimate plotted against $1\,km/h$ based error limits.

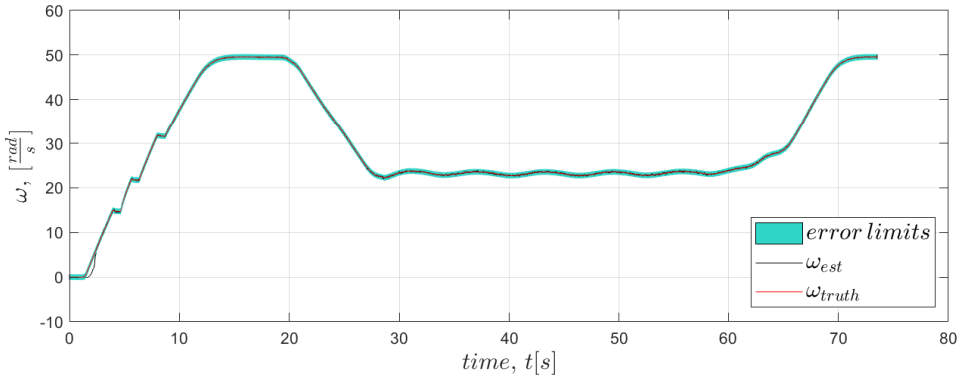


Figure 4.7: Angular speed estimation with error limits in Slalom maneuver

Results of angular speed estimation for other scenarios are in Appendix B. Accurate estimation of ω has therefore been achieved under different conditions which is a major goal of this thesis.

4.2.2 Estimation of angular displacement

Estimating the angular displacement of the wheel (θ) is also a major scope of this thesis, this would involve knowing the initial position of the rim IMU(s) on the wheel θ_0 using the section 3.6 and then using it as the first element of the prior x_0 vector to calculate the angular displacement θ as the wheel rotates.

4.2.2.1 Estimation of initial position

The left wheel of the truck has 6 IMUs mounted on its rim as in figure 4.8, the true value of θ_0 and their estimates obtained using noisy accelerometer measurements of the rim IMU when the truck is standstill using the estimation method described in section 3.6 are shown in table 4.6.

Sensor	True position $[rad](deg)$	Estimated position $[rad](deg)$
Sensor 1	0.030 (1.740)	0.005 (0.182)
Sensor 2	1.077 (61.740)	1.0833 (60.552)
Sensor 3	2.823 (161.740)	2.815 (162.740)
Sensor 4	-3.111 (181.740)	-3.019 (183.760)
Sensor 5	-2.064 (241.740)	-2.063 (241.764)
Sensor 6	-0.668 (321.740)	-0.656 (321.798)

Table 4.6: Estimation results of initial sensor position

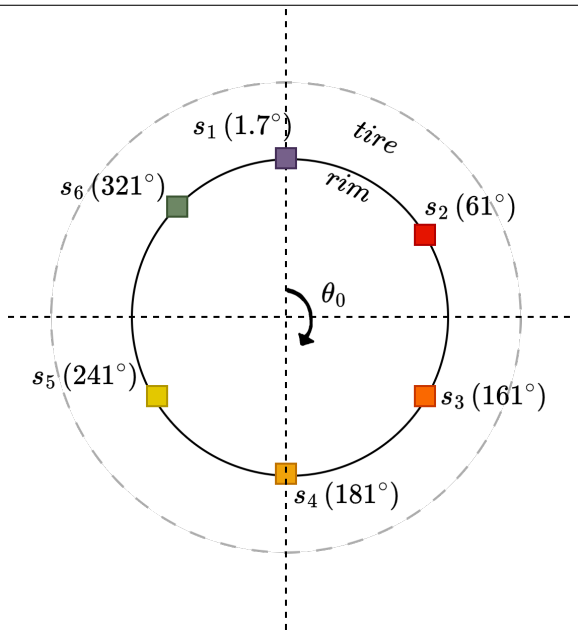


Figure 4.8: Wheel with 6 rim IMUs

4.2.2.2 Estimation of relative angular displacement

The angular displacement is calculated as $\theta_{k+1} = \theta_k + T\omega_k + \frac{T^2}{2}\alpha_k$, where ω_k is the estimated angular speed from section 4.2.1 and α_k is the estimated angular acceleration from section 4.2.3. For the same slalom maneuver as previously used, the angular displacement estimation is shown in figure 4.9, also shown is the estimation error. The estimated θ is wrapped between 0 and 2π.

Maneuver	mean(|Error|) $[rad]$	RMSE $[rad]$	max(|Error|) $[rad]$
Slalom	0.108	0.135	0.663
Creep	0.034	0.041	0.192
Slope up-down	0.138	0.164	0.606
Straight	0.127	0.149	0.603
Figure 8 with bank	0.086	0.112	0.643
Active slip control(open loop)	0.126	0.294	3.126

Table 4.7: Error in angular displacement estimation using TruckMaker

Accurate estimation of θ is a combination of estimating θ_0 and the relative displacement, this been achieved under different conditions. Results of angular displacement estimation for other scenarios are in Appendix B.

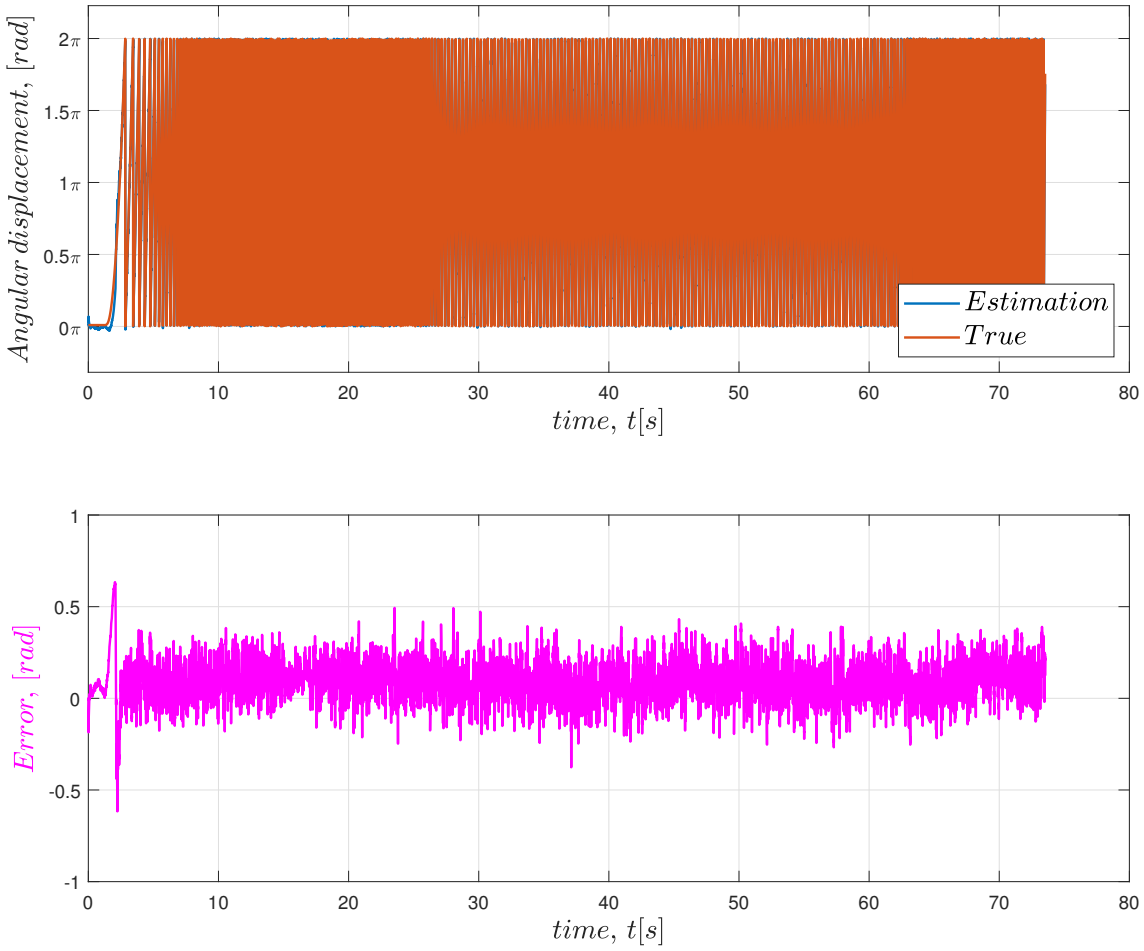


Figure 4.9: Angular displacement estimation versus true in Slalom maneuver

4.2.3 Estimation of angular acceleration

The last state to be estimated is the angular acceleration α of the wheel. The angular acceleration estimation of the wheel in the same slalom maneuver is shown in figure 4.10. The error in estimation of α also shown. The relatively higher error in estimation of α is due to the limitations of the EKF, and the balance between EKF being adaptive to transients or being an aggressive smoother.

Maneuver	mean(|Error|) $[\frac{rad}{s^2}]$	RMSE $[\frac{rad}{s^2}]$	max(|Error|) $[\frac{rad}{s^2}]$
Slalom	0.599	1.030	7.345
Creep	0.030	0.136	3.360
Slope up-down	0.610	0.955	6.888
Straight	0.740	1.338	7.304
Figure 8 with bank	0.653	1.012	6.560
Active slip control(open loop)	6.597	33.048	624.876

Table 4.8: Error in angular acceleration estimation using TruckMaker

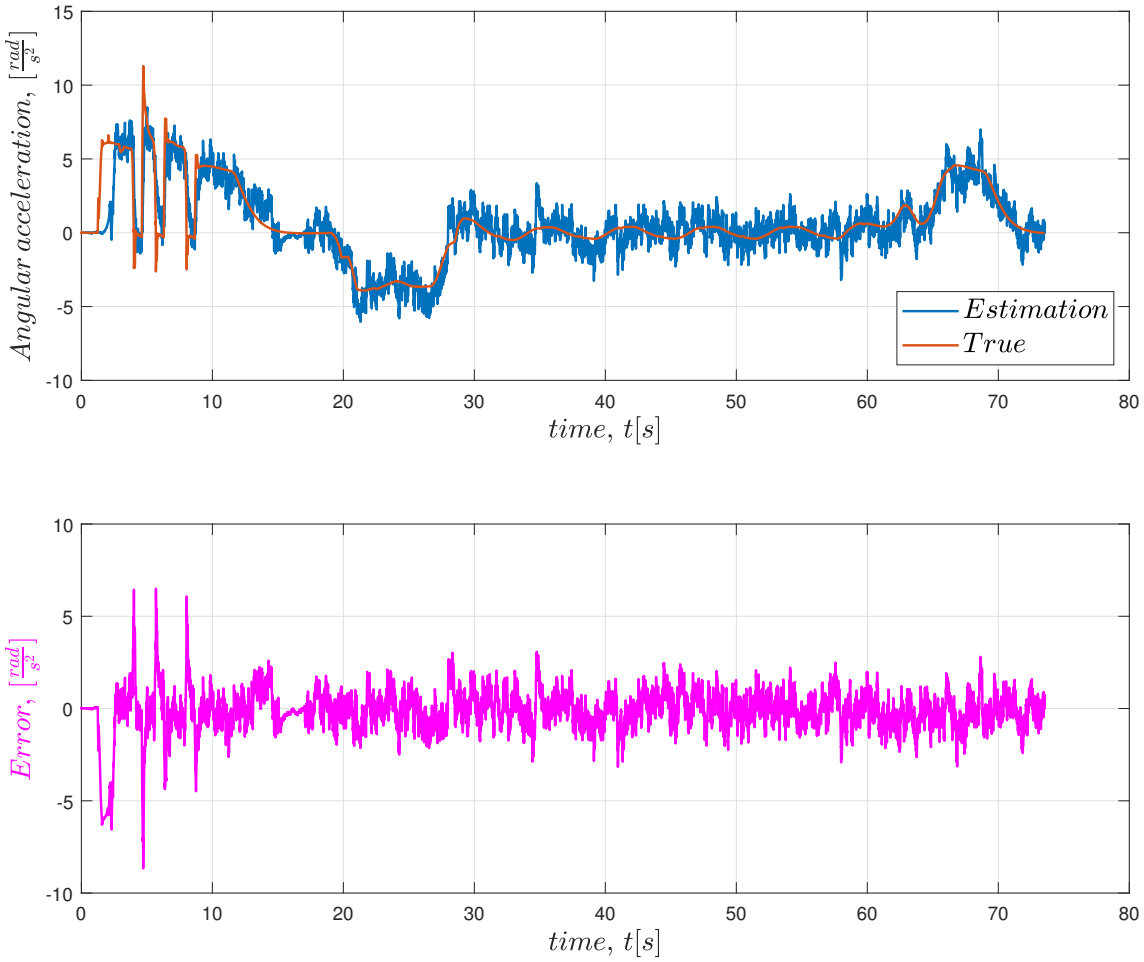


Figure 4.10: Angular acceleration estimation versus true in Slalom maneuver

Results of angular acceleration estimation for other scenarios are in Appendix B. The results from the estimator show that estimation to be accurate to a large extent with minimal error between the true and estimated states. The estimation accuracy can be improved based on the end user's need. If accuracy of ω is of higher priority that the other estimates, than the EKF can be tuned to meet that requirement and vice versa to other states. The tuning that was used in this thesis is a balanced outcome of accuracy and speed of the filter.

4.3 Open loop simulation

The slip controller described in section 3.9 was developed in TruckMaker for Simulink and embedded into TruckMaker. The controller would supply the truck model with the braking torque and the inputs to the controller would be outputs from the truck model. The initial approach involved embedding the estimator and controller, where the estimator would supply angular wheel speed information to the controller and other parameters from the truck model to create a estimator-controller-model simulation. However, this attempt was not successful due to lack of time in getting and the controller and estimator to work correctly in parallel.

For the open loop simulation, only the slip controller was embedded into TruckMaker to create a controller-model simulation. The data generated from truck was then processed through the estimator to calculate the angular displacement, angular speed, and angular acceleration of the wheel. In future, the controller and estimator would work in parallel.

Figure 4.11 shows the estimation of angular speed using the rim and hub IMUs of the front-left wheel of the truck in a scenario where the controller was active. In this maneuver, the truck first accelerates to $80\,km/h$ from standstill in a straight line for first 15 seconds, and the brake is applied before the vehicle comes to stand still. The low friction (0.3) of the asphalt surface promotes the activation of the controller.

The estimated angular wheel speed is compared to the true value and estimation error is also shown in figures 4.11 and 4.12. Whereas figures 4.13 & 4.14 shows the true vs estimated angular displacement and angular acceleration in active slip control maneuver respectively.

The large error in estimation observed in figures 4.11, 4.13 and 4.14, is due to large changes in angular acceleration of the wheel α under braking. The controller action causes large changes in acceleration of the wheel. Since ω and θ are first and second integrals of α respectively, a large change in acceleration is reflected on the angular velocity and displacement states.

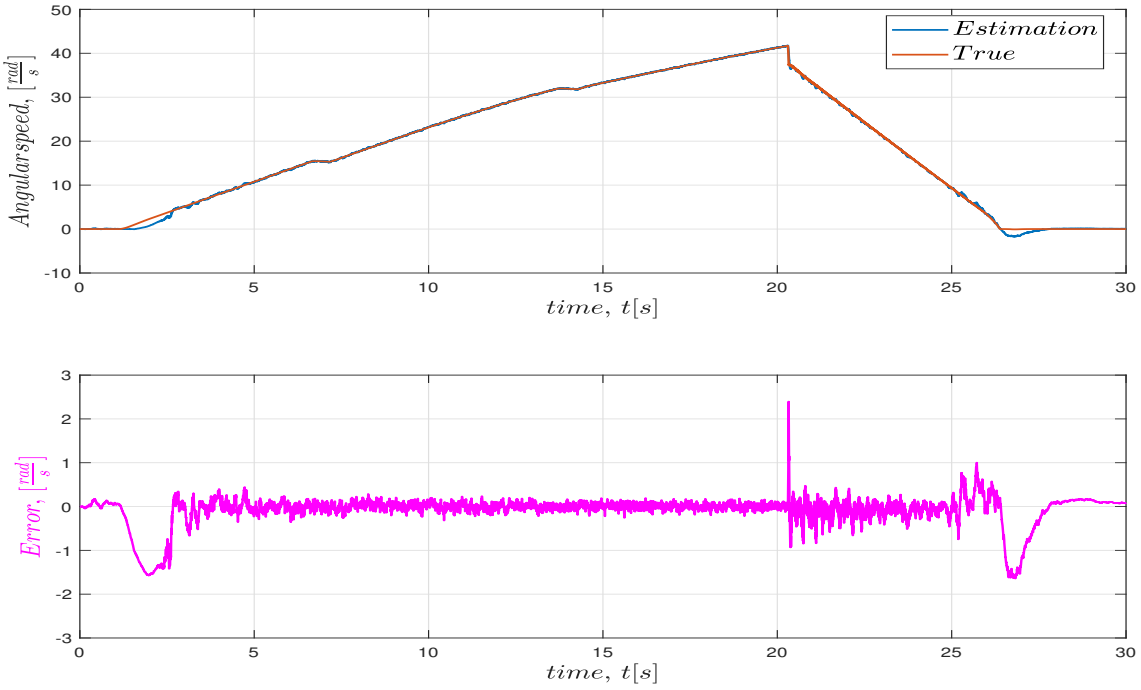


Figure 4.11: Angular speed estimation versus true in active slip control (open loop) maneuver

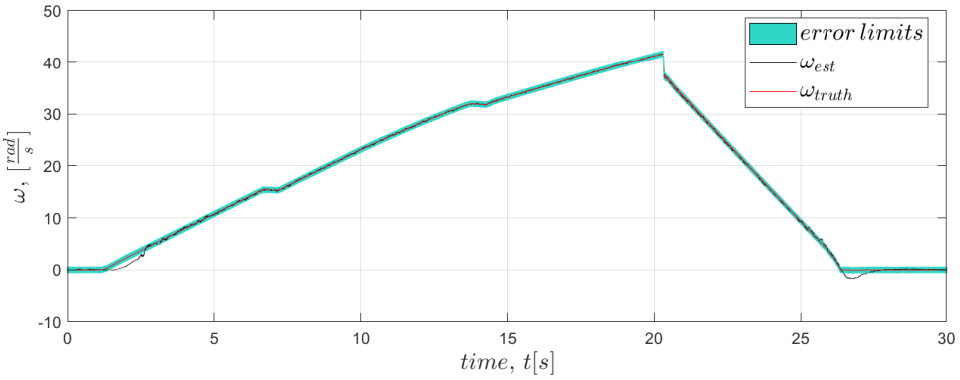


Figure 4.12: Angular speed estimation with error limits in active slip control (open loop) maneuver

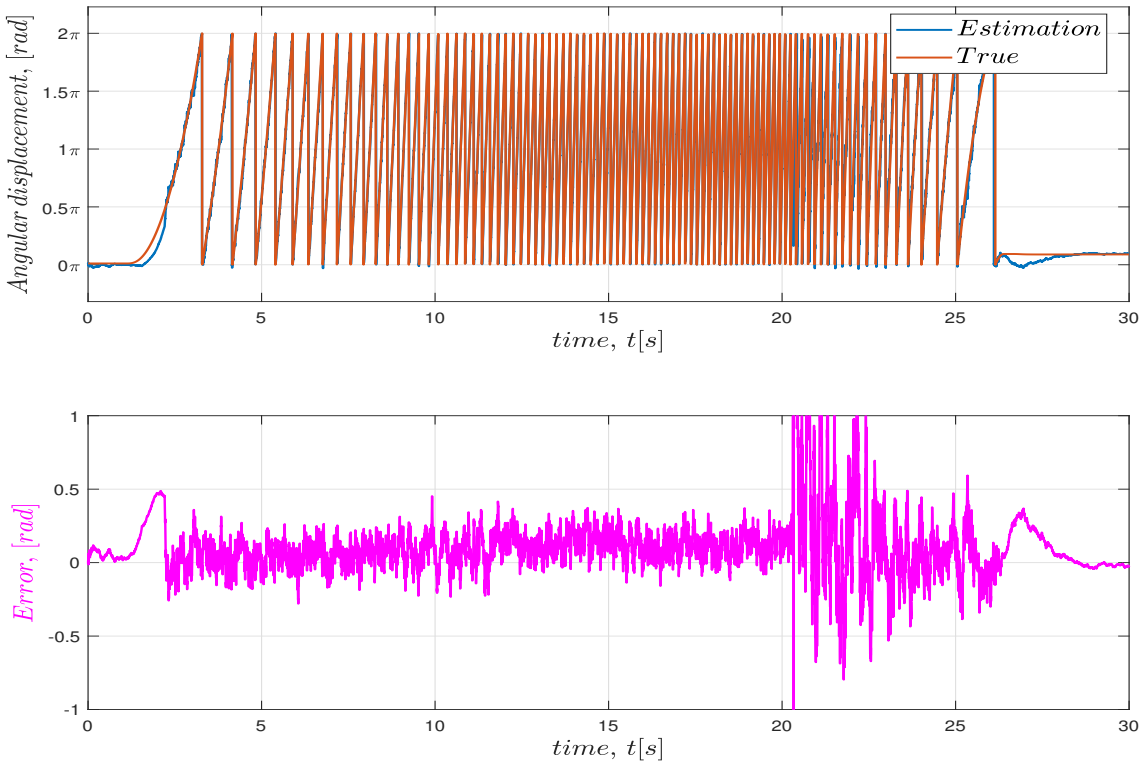


Figure 4.13: Angular displacement estimation versus true in active slip control (open loop) maneuver

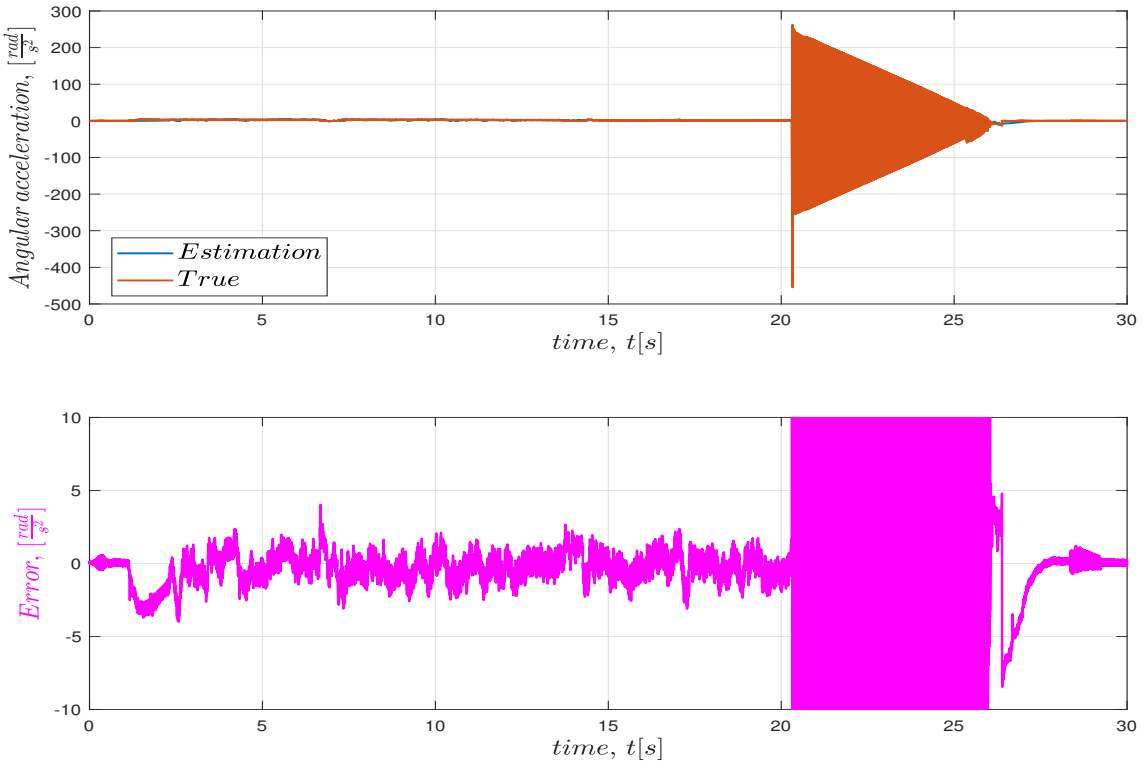


Figure 4.14: Angular acceleration estimation versus true in active slip control (open loop) maneuver

4.4 Roll and slope estimation using true sensors

In section 3.8.1, estimating the roll and slope angle using accelerometers, quaternions and the linear acceleration sensors (without effect of gravity) is formulated. The estimator uses noisy acceleration measurements from the hub IMU and also noisy acceleration values of the truck linear acceleration.

The roll and slope angles are usually measured in *degrees* and the estimation of the angles are checked for following driving scenarios

Straight-up-down-straight: The truck travels in a straight line till 12 s, uphill till 37 s, straight line till 60 s, then downhill till 75 s and then again straight till 150s. The uphill portion of the trajectory has a negative sign in the slope plot according to the ENU convention in the figure 4.15 and vice versa for the downhill portion.

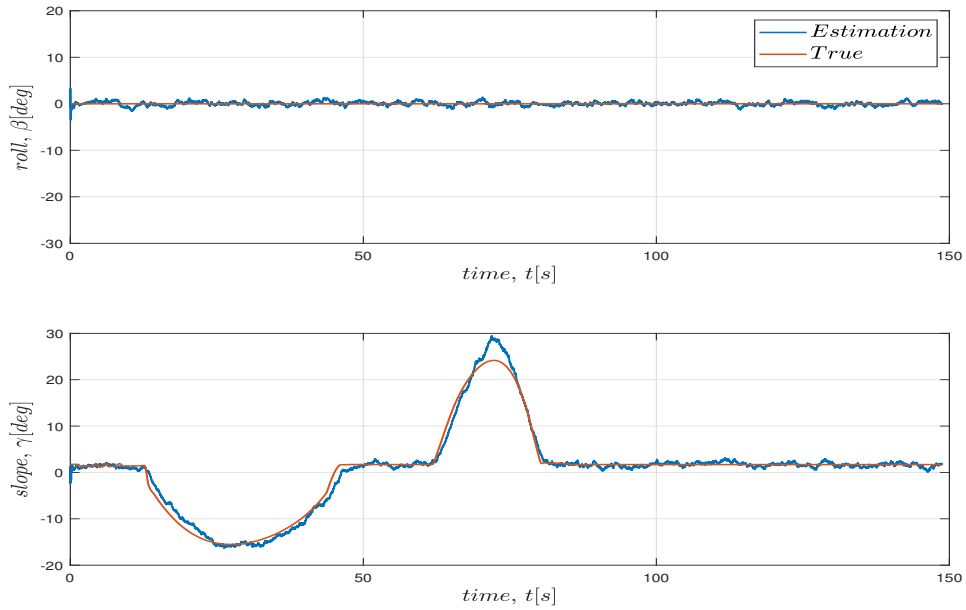


Figure 4.15: Roll and slope estimation versus true in straight-up-down-straight.

"Figure-8" with bank: The truck travels straight for 13 s and then along a "Figure-8" pattern twice and then back on a straight trajectory as shown in figure 4.16. The "Figure-8" pattern has varying banked sections along its path.

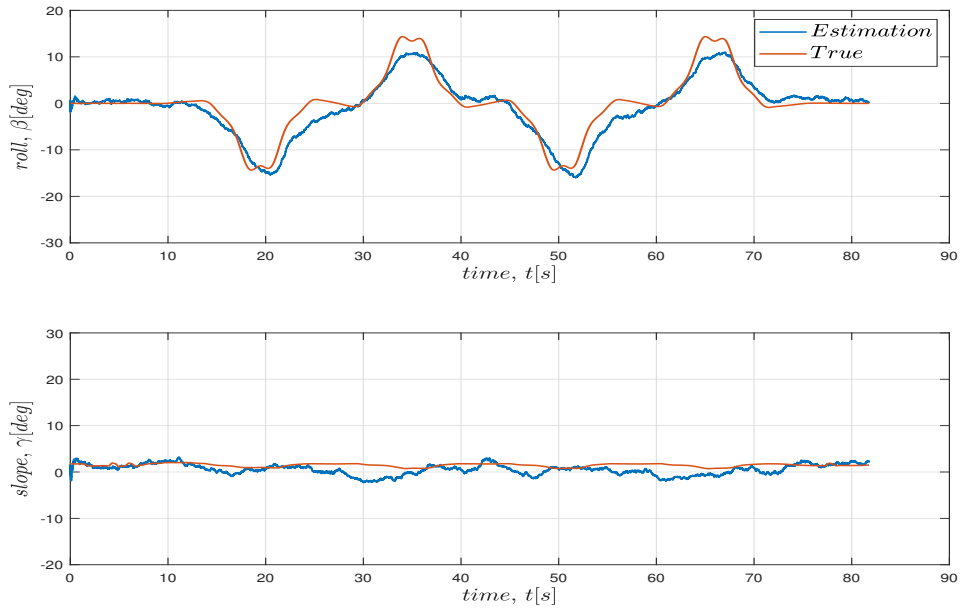


Figure 4.16: Roll and slope estimation versus true in "Figure-8" with bank.

Straight with bank: The truck travels in a straight line but along a banked road as shown in figure 4.17.

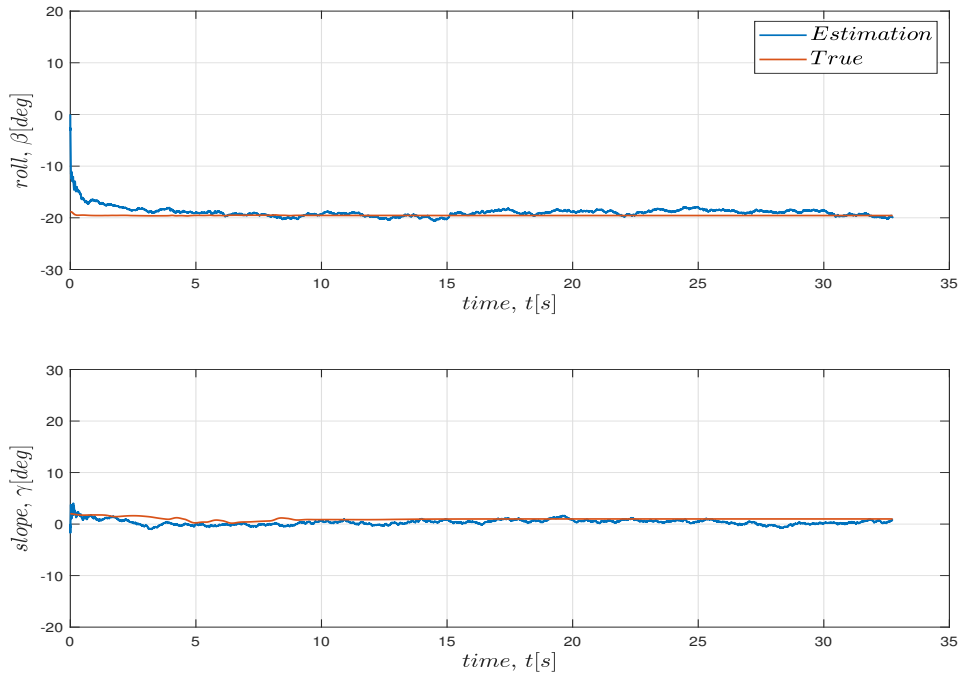


Figure 4.17: Roll and slope estimation versus true in straight banked road.

Tables 4.9 and 4.10 show the accuracy of the roll and pitch estimates in different scenarios. Since the bank and slope of of the road surface is a slow changing parameter, the estimation can be run slowly and the first 2 seconds of ignored for error calculation.

Maneuver	mean(|Error|) [deg]	RMSE [deg]	max(|Error|) [deg]
Straight-up-down-straight	0.320	0.401	1.384
Figure8-bank	1.666	2.187	5.914
Constant bank	0.770	0.938	2.293

Table 4.9: Error in roll estimation using TruckMaker

Maneuver	mean(|Error|) [deg]	RMSE [deg]	max(|Error|) [deg]
Straight-up-down-straight	0.727	1.045	3.530
Figure8-bank	1.216	1.549	4.272
Constant bank	0.648	0.755	1.777

Table 4.10: Error in pitch estimation using TruckMaker

The results show that in presence of sensors that measure only the linear acceleration of the truck (without effect of gravity), and a wheel hub accelerometer that measures

the acceleration and projection of the gravity vector as the truck undergoes roll and slope changes, the roll and slope angles of the wheel can be estimated to a high accuracy.

5

Discussion

5.1 Sensor modeling and simulation

The modeling and simulation of sensors in a virtual environment offers a reliable platform for assessing sensor performance under different road conditions. To do this, Modelon Impact and IPG TruckMaker were used in this study, proving their ability to accurately simulate real-world events. Accurate wheel rotational speed data as well as vehicle roll and pitch data were the main goals since they are essential for control system development and vehicular dynamics analysis.

5.2 Verification of simulated data

The rigid body kinematic equations developed in this study were thoroughly verified using simulated data, first from Modelon Impact and then for complex scenarios from IPG TruckMaker. As detailed in the results section, the error was found to be low, underscoring the accuracy of these equations. Furthermore, to ensure consistency and reliability, the equations were cross-verified between both simulation environments. This was accomplished by generating and comparing data from similar driving scenarios in each simulation platform.

5.3 Estimation accuracy and precision

According to the simulation results, wheel rotational speed may be estimated with great accuracy, typically with an error margin of less than $0.55\,rad/s$ in most conditions. The model's ability to accurately estimate the wheel's rotational speed is indicated by its low error rate, which offers a further research and testing of related technologies. The estimation of both angular displacement and angular acceleration is also highly precise. The estimation could further be improved by using multiple model Kalman filters [20] or other alternative estimation algorithms that couldn't be investigated in this thesis.
The accelerometer model implemented in this thesis only added noise to the ideal sensors, whereas real world sensors will have axis misalignment, biases and temperature related biases that were not used in this thesis. The estimation filter in this case would need to be augmented with states that keep track of the biases and other noise parameters or determining the bias separately and then add the contribution of bias.

5.4 Roll and Pitch information

The accurate estimation of vehicle roll and pitch was achieved using true linear acceleration data. This information is critical for understanding the vehicle's orientation and dynamics during maneuvers. Similar to estimation of kinematics of the wheel, estimation of the roll and pitch information is a balance between speed and accuracy of estimates. For the sake of visualization and limitations of simulation, scenarios used in this thesis contained roads with quickly changing and "unrealistic" bank and slope values. In real world road conditions, the bank and slope values of roads evolve slowly, this discrepancy needs to be addressed using filters that evolve slowly compared to the implementation in this thesis.

5.5 Setup of sensors

At least one IMU sensor on the wheel rim per wheel is needed for estimation of kinematic parameters. However, in order to estimate roll and pitch, a non-rotating IMU installed on the wheel hub or any non rotating frame is needed, in addition to radar or other estimates of vehicle's longitudinal and lateral accelerations.

The rim IMU should at least have a 2-axes accelerometer mounted to measure the radial and tangential acceleration of the wheel. The hub IMU should have a 3-axes accelerometer measuring accelerations in x, y and z directions.

5.6 Limitations and Challenges

Despite the promising results, few limitations and challenges were identified:

- **Power Delivery and Communication:** Powering the sensors is a challenge because they are positioned on the rims of each tire to estimate wheel speed. The assessment is also contingent upon the other systems' communication dependability. There has to be more research done to determine how much power the sensor uses for processing, transmission, and data collection.
- **Operating Temperature of the IMU:** There is a defined working temperature range for the Inertial Measurement Unit (IMU) that is employed in the simulation. Variations from this range may affect the sensor's functionality and result in data inaccuracies. Maintaining the simulation's accuracy depends on the IMU operating within its ideal temperature range.
- **Angular Velocities (ω_x and ω_z):** The assumption that no significant angular velocities are present around the x and z axes in relation to the y axis i.e. $\omega_x = \omega_z \sim 0$ and $\alpha_x = \alpha_z \sim 0$ during most truck maneuvers. Although Gaussian approximations often perform well in many filtering problems, they may not be appropriate when the distributions exhibit characteristics such as multi-modality, skewness, or other deviations from Gaussian behavior. Other filtering alternatives like particle filters may be suitable.

5.7 Future Work

There are additional difficulties when it comes to post-processing real data from a real truck. To guarantee its dependability, the precision of the simulated sensor data must be verified against actual data in future. The closed loop testing can be implemented using the estimated wheel speed to develop a slip controller. Further studies have to concentrate on resolving the highlighted constraints, specifically enhancing the reliability of power distribution and communication and expanding the IMU's operational temperature range. Furthermore, in relation to ω_y, the angular velocities (ω_x and ω_z) are taken as zero. However, in situations where these velocities should be taken into consideration for future research.

5.8 Ethical and Sustainability aspects

Precise measurement and estimation of wheel speed is essential to enable brake system applications, such as anti-lock braking, traction control, stability control and almost all control functions dealing with tire vehicle dynamics. Fast and accurate wheel speed estimation is enabler for optimizing the performance of such control algorithms in real time applications. Good performance of vehicle dynamics control functions can help reduce accidents, reduce injuries, loss of life and property.
The trend of using embedded systems at wheel is in tie with smart tire and autonomous driving functionality. Nevertheless, it can impact the vehicle dynamics control function in all types of vehicles. The advantage lies in improving the efficiency of control algorithms, implementing more advanced controllers and consequently elevating safety levels.
Last but not least, the proposed technology is highly beneficial for autonomous driving which plays a crucial role in promoting sustainability.

Bibliography

[1] VECTORNAV, "Mems acceleromters," 2024, last accessed 8 May 2024. [Online]. Available: 'https://www.vectornav.com/resources/inertial-navigation-primer/theory-of-operation/theory-mems'

[2] O. Mezentsev and J. Collin, "Design and performance of wheel-mounted mems imu for vehicular navigation," in *2019 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*. IEEE, 2019, pp. 1–4.

[3] J. Morgan and J. M. Conrad, "An ekf, accelerometer, gravity based wheel odometry method," in *2021 IEEE 18th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*. IEEE, 2021, pp. 94–99.

[4] X. Niu, Y. Wu, and J. Kuang, "Wheel-ins: A wheel-mounted mems imu-based dead reckoning system," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 9814–9825, 2021.

[5] M. Moussa, A. Moussa, M. Elhabiby, and N. El-Sheimy, "Wheel-based aiding of low-cost imu for land vehicle navigation in gnss challenging environment," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. IEEE, 2020, pp. 1–6.

[6] A. A. Youssef, N. Al-Subaie, N. El-Sheimy, and M. Elhabiby, "Accelerometer-based wheel odometer for kinematics determination," *Sensors*, vol. 21, no. 4, p. 1327, 2021.

[7] L. Villani and G. Oriolo, "Robotics: Modelling, planning and control."

[8] J. L. Meriam, L. G. Kraige, and J. N. Bolton, *Engineering mechanics: dynamics*. John Wiley & Sons, 2020.

[9] BS-ISO, "8855: 2011. road vehicles—vehicle dynamics and road-holding ability—vocabulary," Tech rep., ISO, Tech. Rep., 2011.

[10] T. Glad and L. Ljung, *Control theory multivariable and nonlinear methods*. Taylor and Francis, 2000.

[11] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *Foundations and Trends® in Signal Processing*, vol. 11, no. 1–2, p. 1–153, 2017. [Online]. Available: http://dx.doi.org/10.1561/2000000094

[12] W. R. Hamilton, "On quaternions; or on a new system of imaginaries in algebra," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 25, no. 163, pp. 10–13, 1844. [Online]. Available: https://doi.org/10.1080/14786444408644923

[13] T. I. Fossen, "Handbook of marine craft hydrodynamics and motion control," *John Willy & Sons Ltd*, 2011.

[14] Modelica, "Modelon impact." [Online]. Available: 'https://help.modelon.com/latest/articles/ao_what_is_modelon_impact/'

[15] A. Marzbanrad, F. Bruzelius, B. Jacobson *et al.*, "Enhanced sliding mode wheel slip controller for heavy goods vehicles."

[16] MathWorks, "Imu simulation model," 2024, last accessed 21 May 2024. [Online]. Available: 'https://se.mathworks.com/help/nav/ref/imusensor-system-object.html#responsive_offcanvas'

[17] P. Billingsley, *Convergence of probability measures*. John Wiley & Sons, 2013.

[18] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of imu and marg orientation using a gradient descent algorithm," 2011.

[19] R. V. Vitali, R. S. McGinnis, and N. C. Perkins, "Robust error-state kalman filter for estimating imu orientation," 2021.

[20] A. Akca and M. Önder Efe, "Multiple model kalman and particle filters and applications: A survey," 2019.

A

Appendix 1

A.1 Verification results

A.1.1 Verification of tangential acceleration

A.1.1.1 Using Modelon Impact

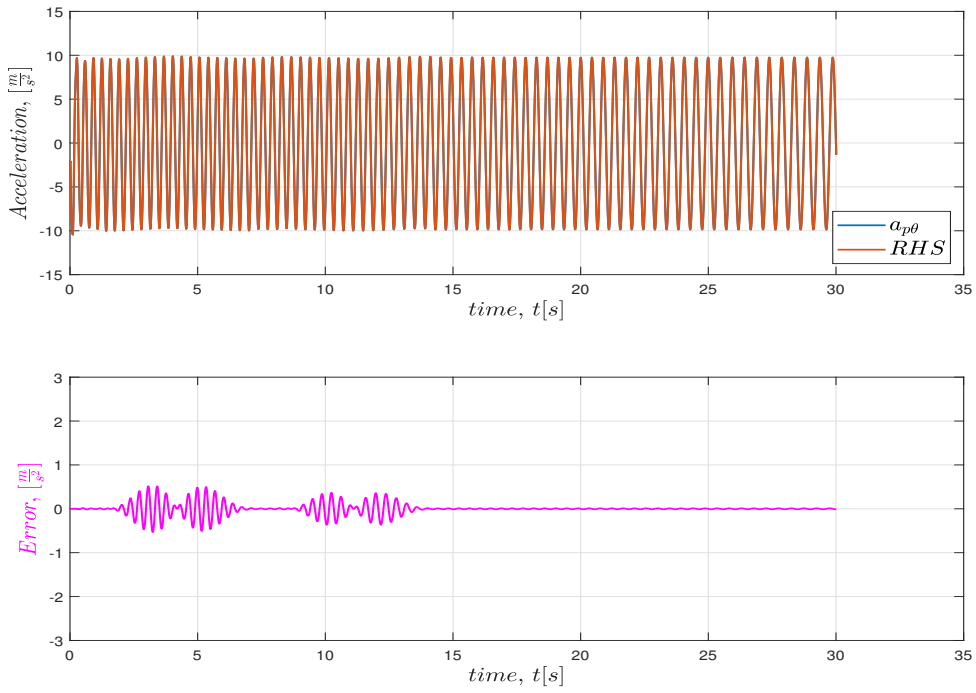


Figure A.1: Verification of rim IMU tangential acceleration using Modelon Impact in double lane change maneuver at $15\frac{m}{s}$

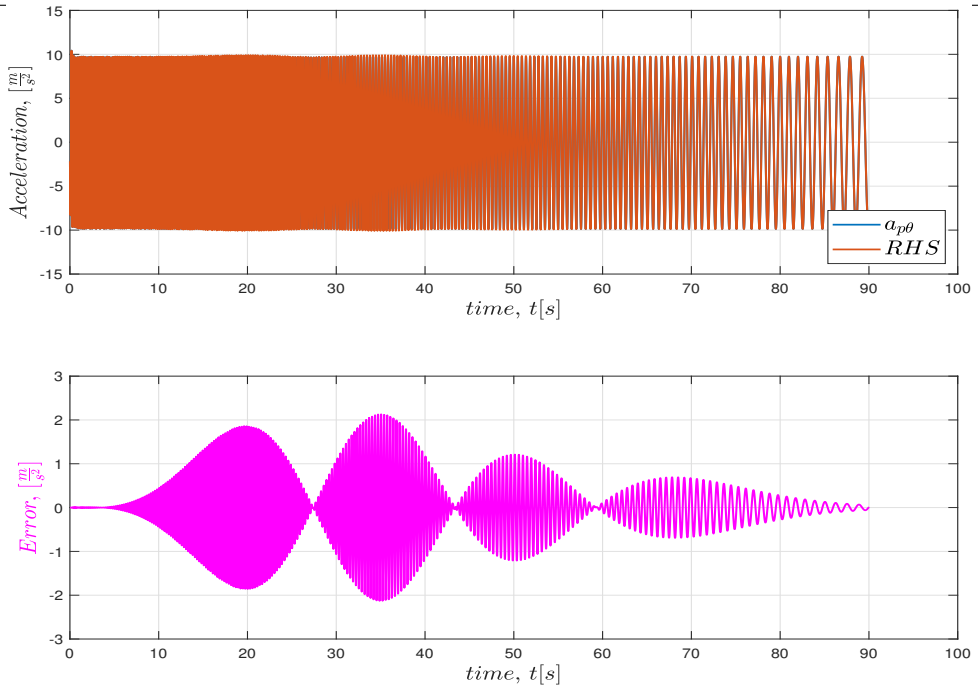


Figure A.2: Verification of rim IMU tangential acceleration using Modelon Impact in steer ramp maneuver

A.1.1.2 Using TruckMaker

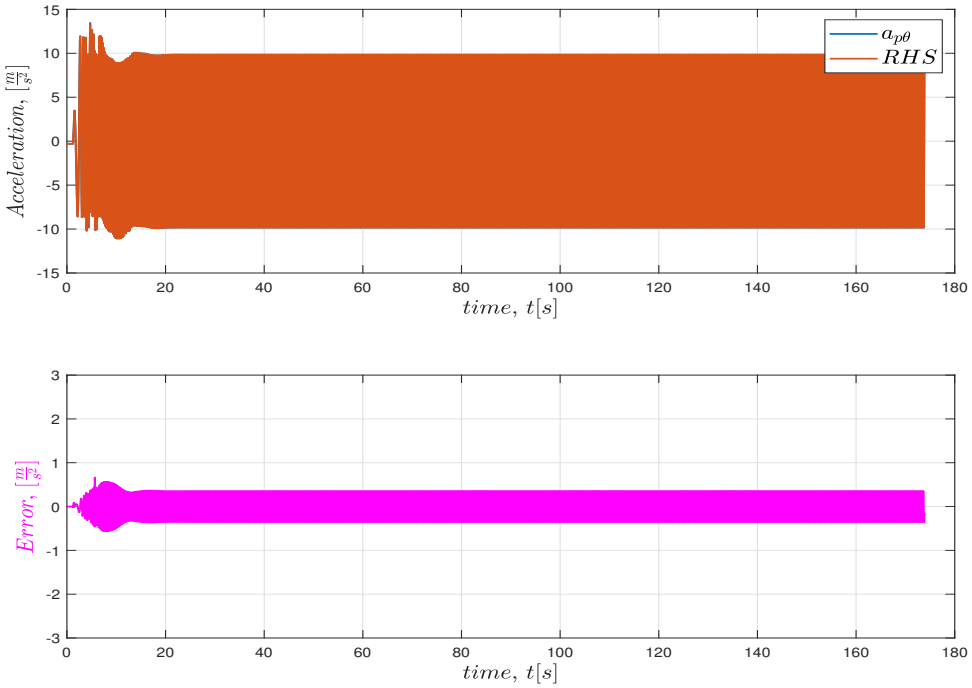


Figure A.3: Verification of rim IMU tangential acceleration using TruckMaker in creep maneuver at $0.3\frac{m}{s}$

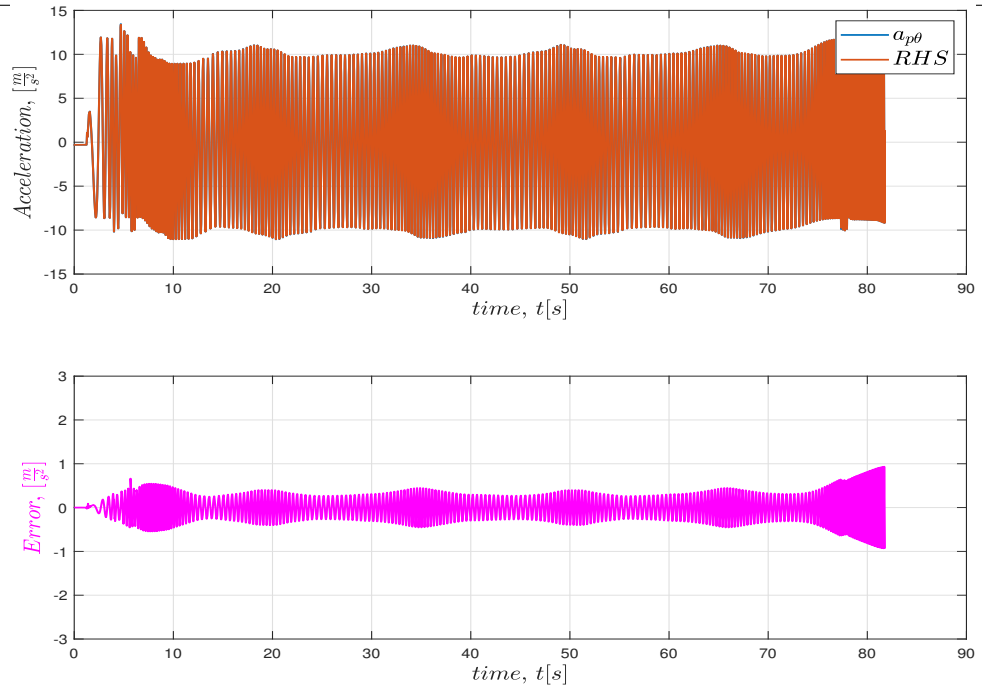


Figure A.4: Verification of rim IMU tangential acceleration using TruckMaker in Figure8 with banked road maneuver

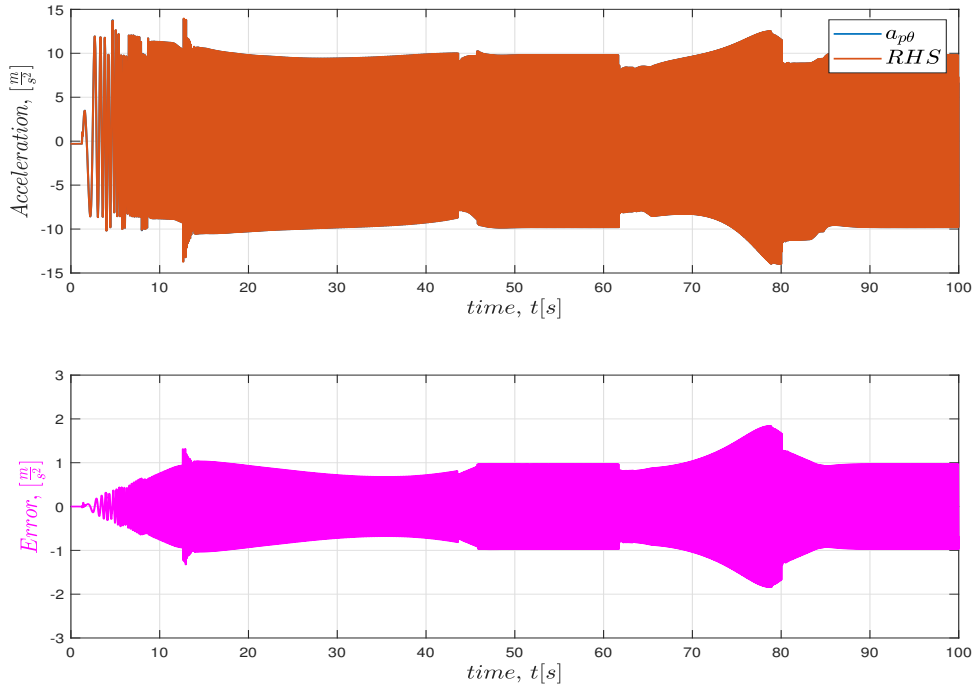


Figure A.5: Verification of rim IMU tangential acceleration using TruckMaker in straight-up-down-straight maneuver

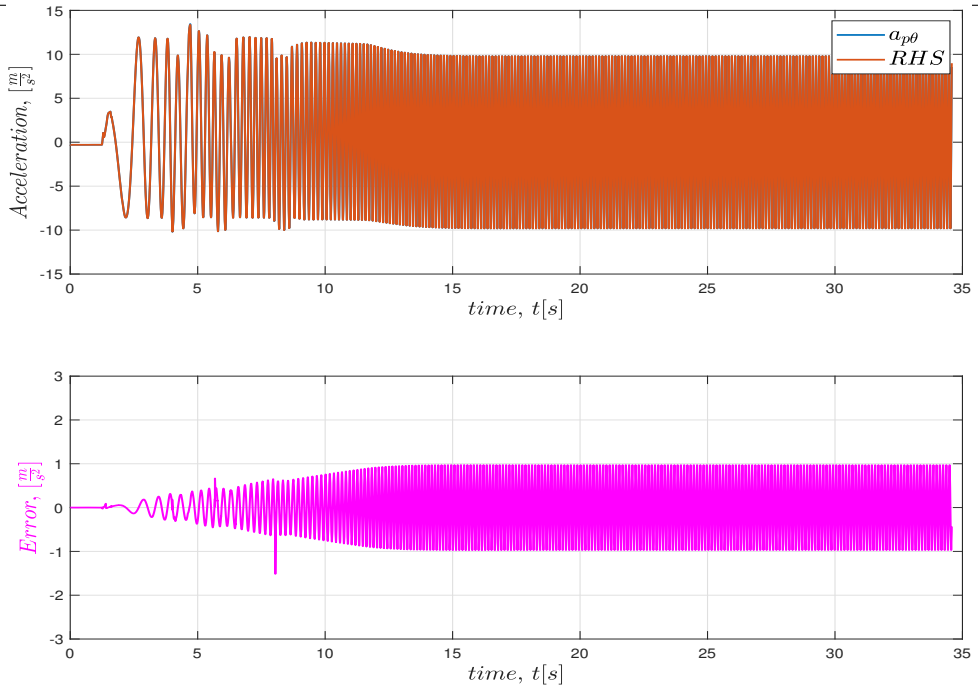


Figure A.6: Verification of rim IMU tangential acceleration using TruckMaker in straight maneuver

A.1.2 Verification of radial acceleration

A.1.2.1 Using Modelon Impact

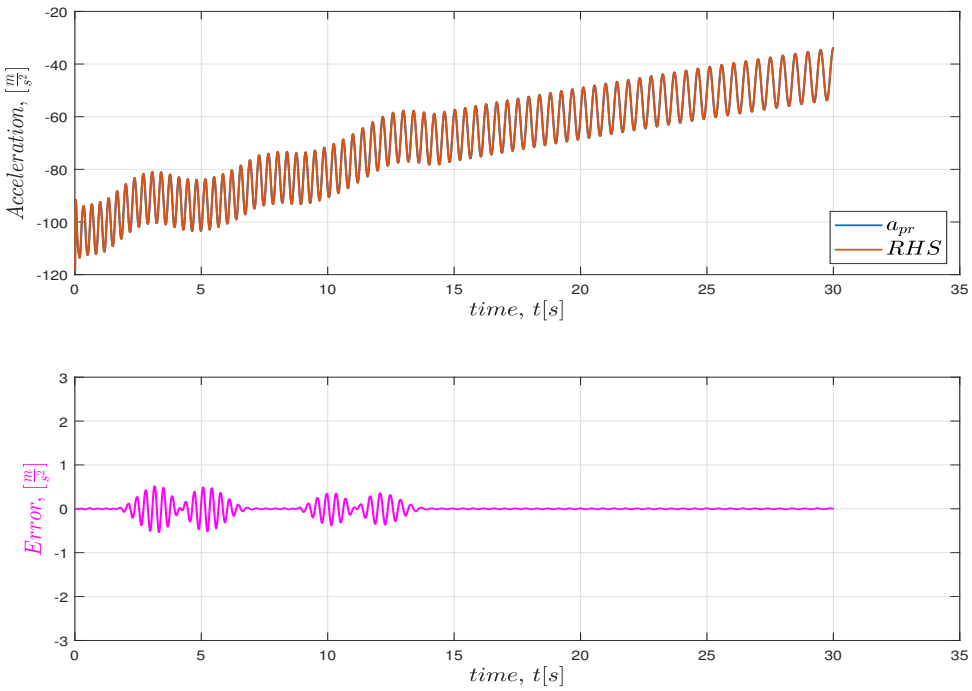


Figure A.7: Verification of rim IMU radial acceleration using Modelon Impact in double lane change maneuver at $15\frac{m}{s}$

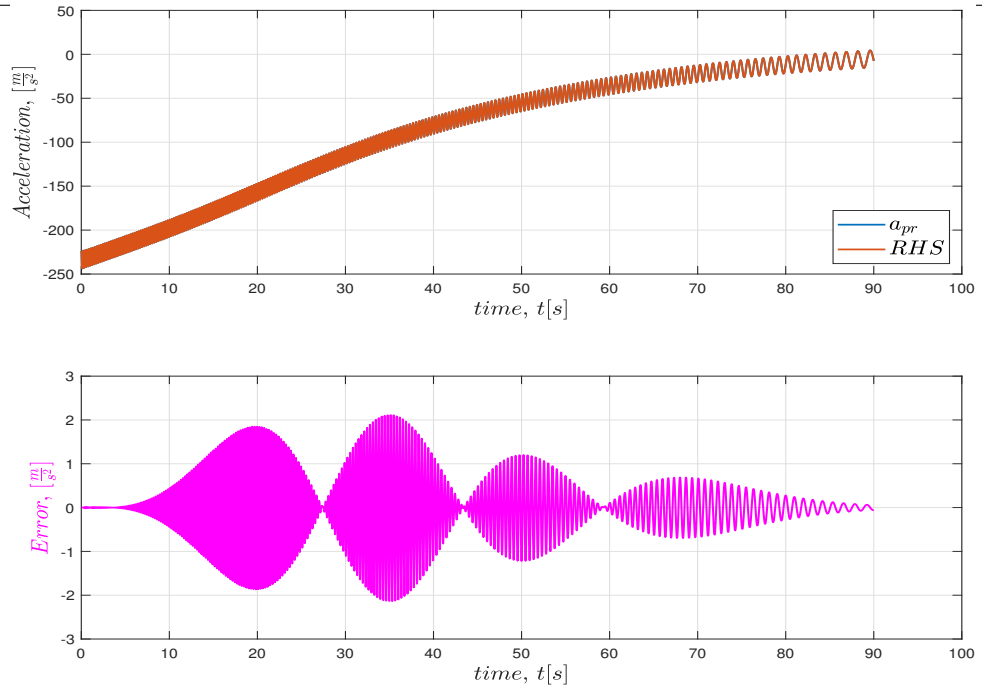


Figure A.8: Verification of rim IMU radial acceleration using Modelon Impact in steer ramp maneuver

A.1.2.2 Using TruckMaker

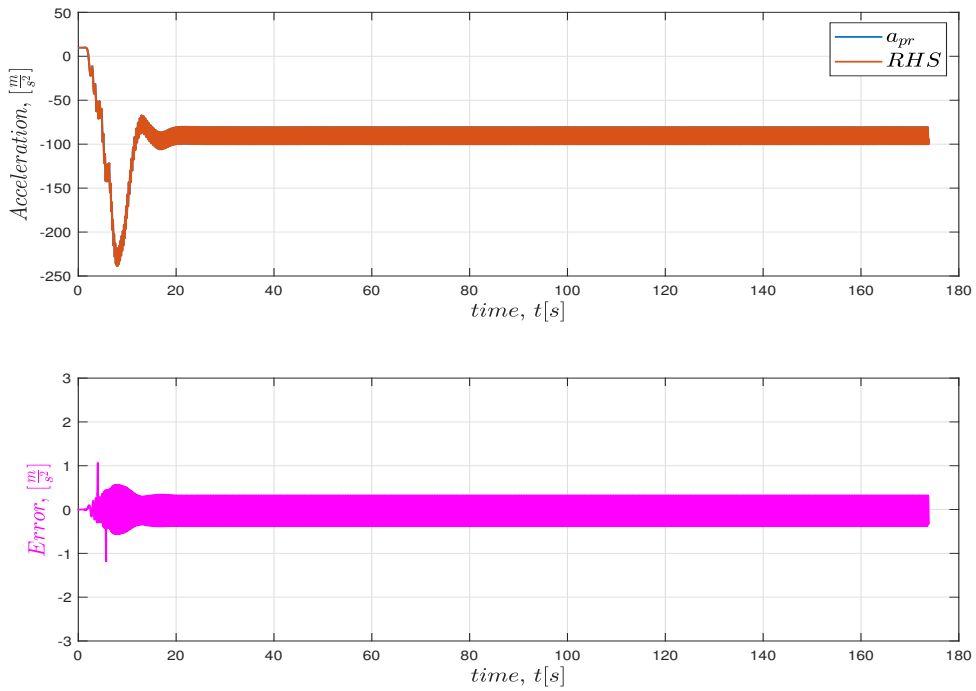


Figure A.9: Verification of rim IMU radial acceleration using TruckMaker in creep maneuver at $0.3\frac{m}{s}$

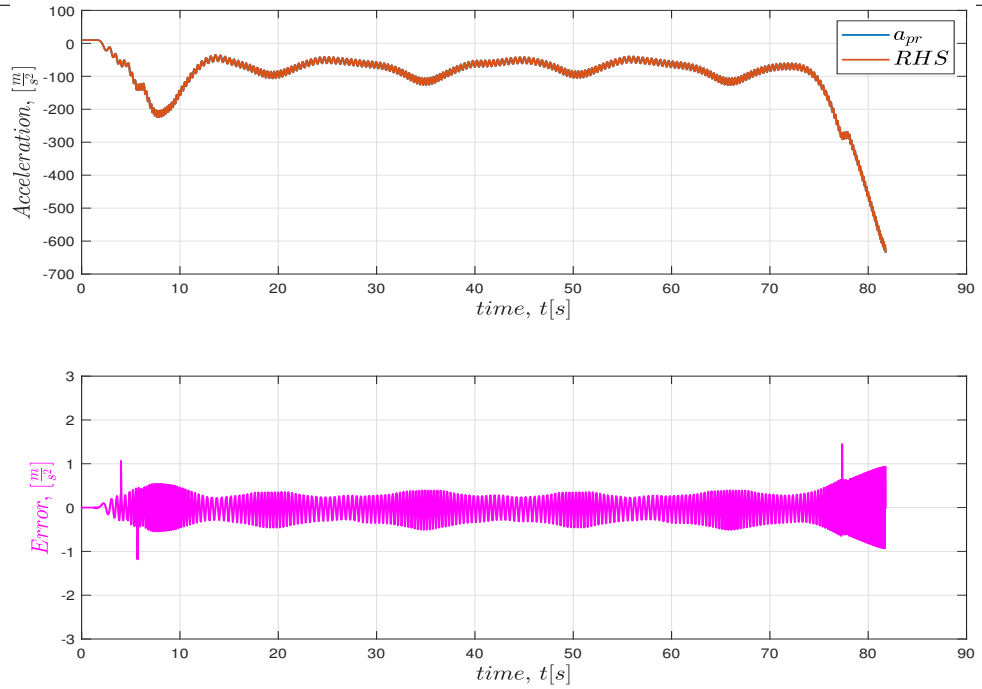


Figure A.10: Verification of rim IMU radial acceleration using TruckMaker in Figure8 with banked road maneuver

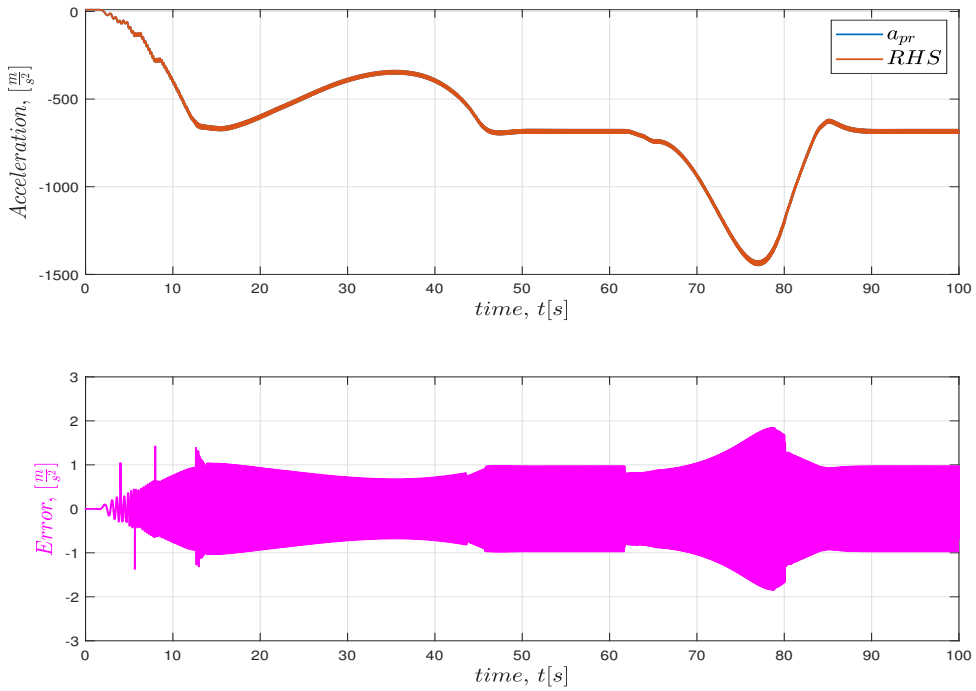


Figure A.11: Verification of rim IMU radial acceleration using TruckMaker in straight-up-down-straight maneuver

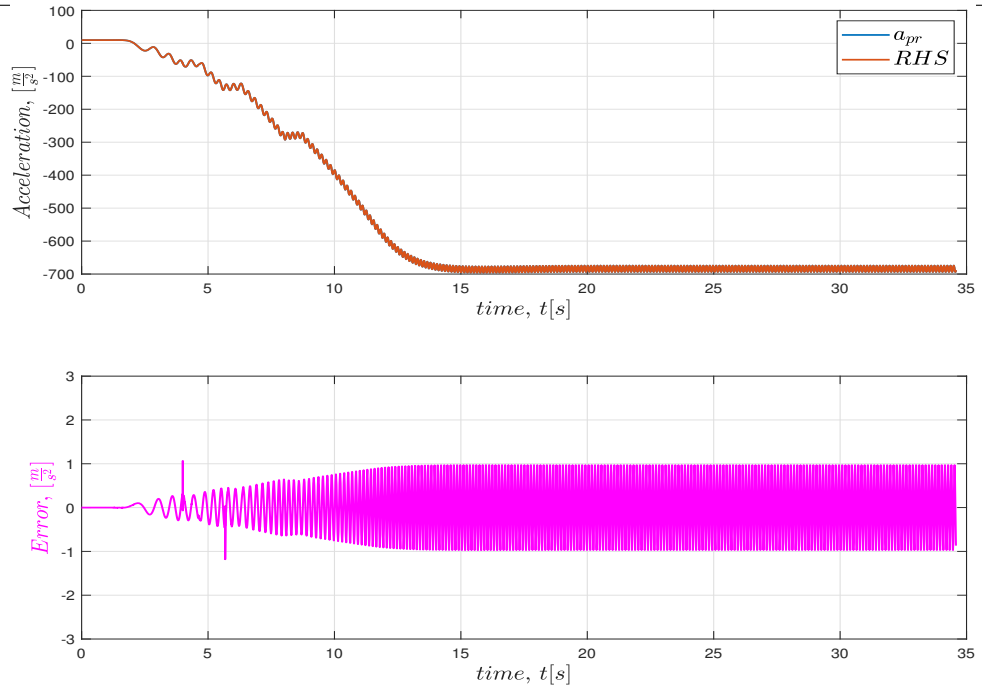


Figure A.12: Verification of rim IMU radial acceleration using TruckMaker in straight maneuver

B

Appendix 2

B.1 State estimation results

B.1.1 Estimation of angular speed

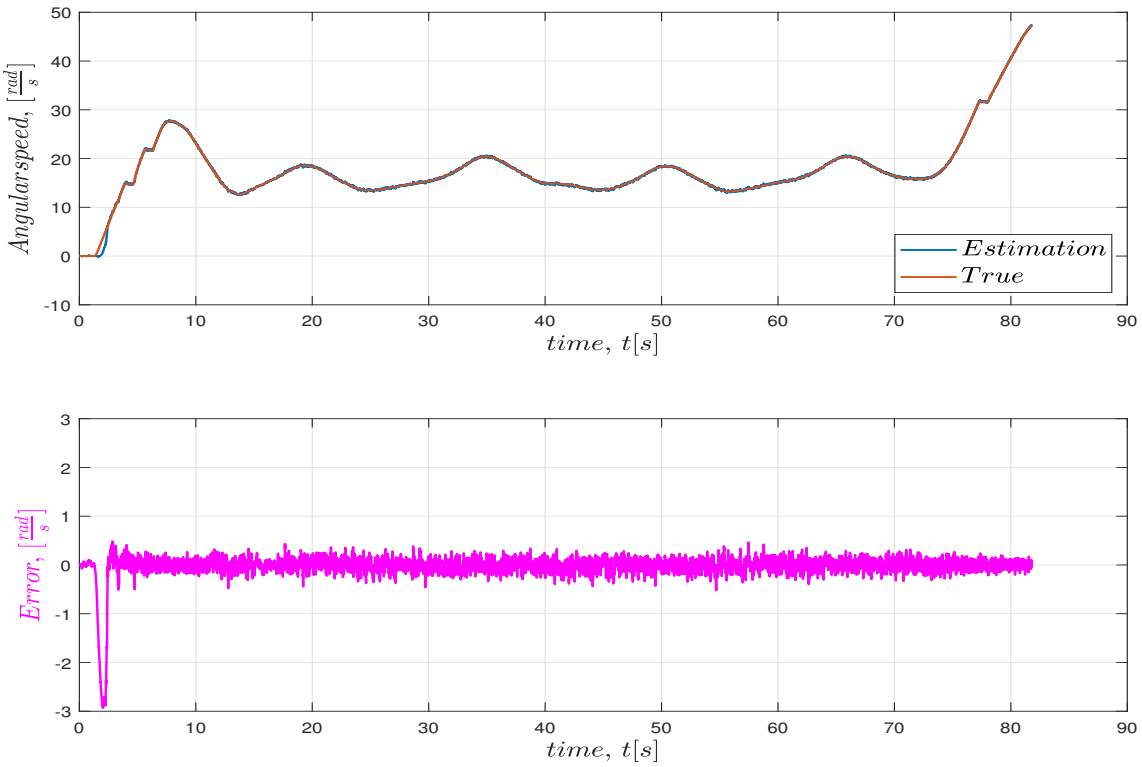


Figure B.1: Angular speed estimation versus true in Figure8 banked road maneuver

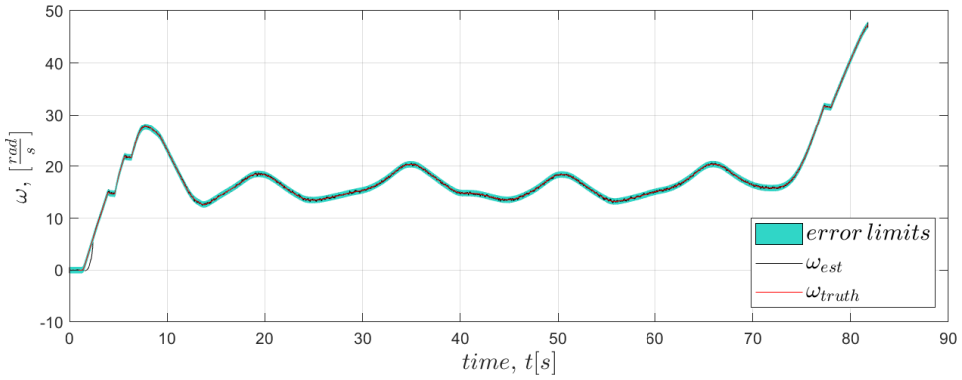


Figure B.2: Angular speed estimation with error limits in Figure8 banked road maneuver

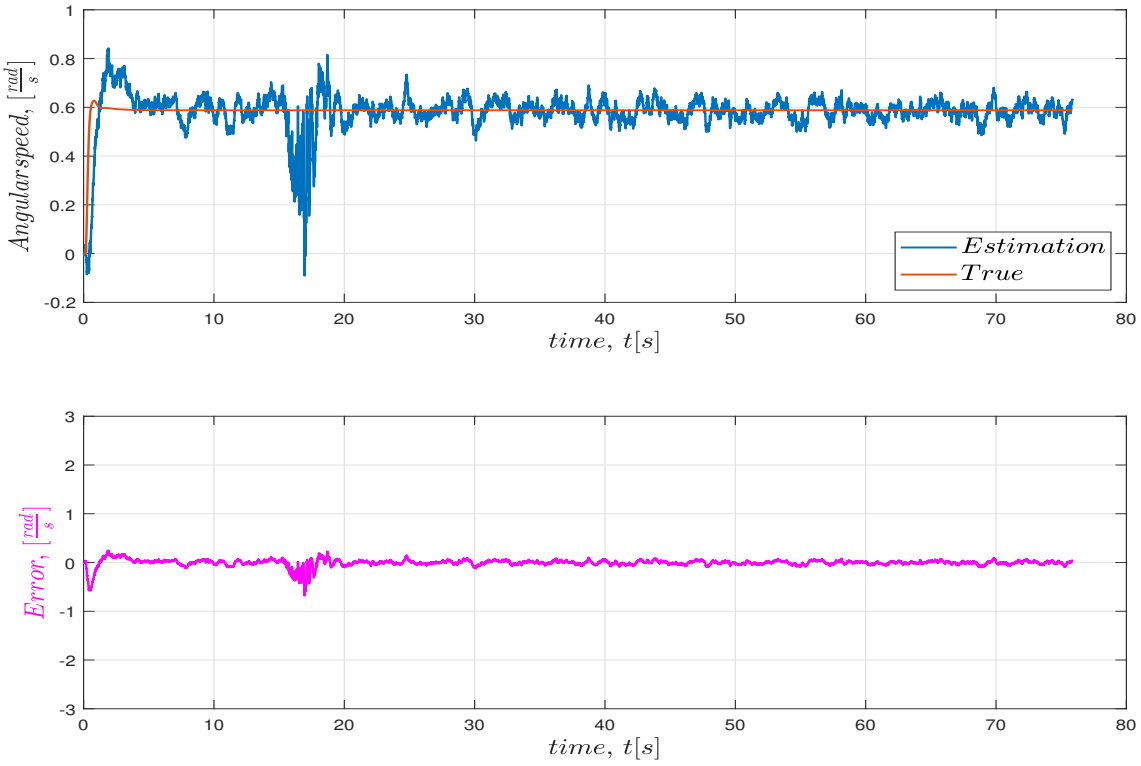


Figure B.3: Angular speed estimation versus true in creep maneuver

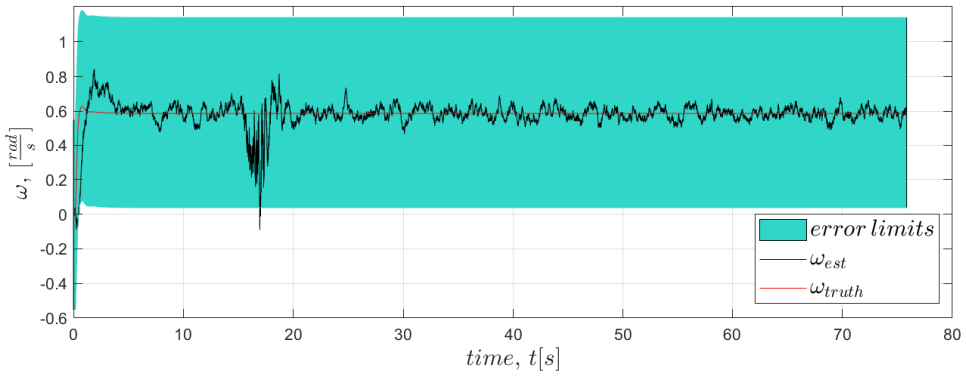


Figure B.4: Angular speed estimation with error limits in creep maneuver

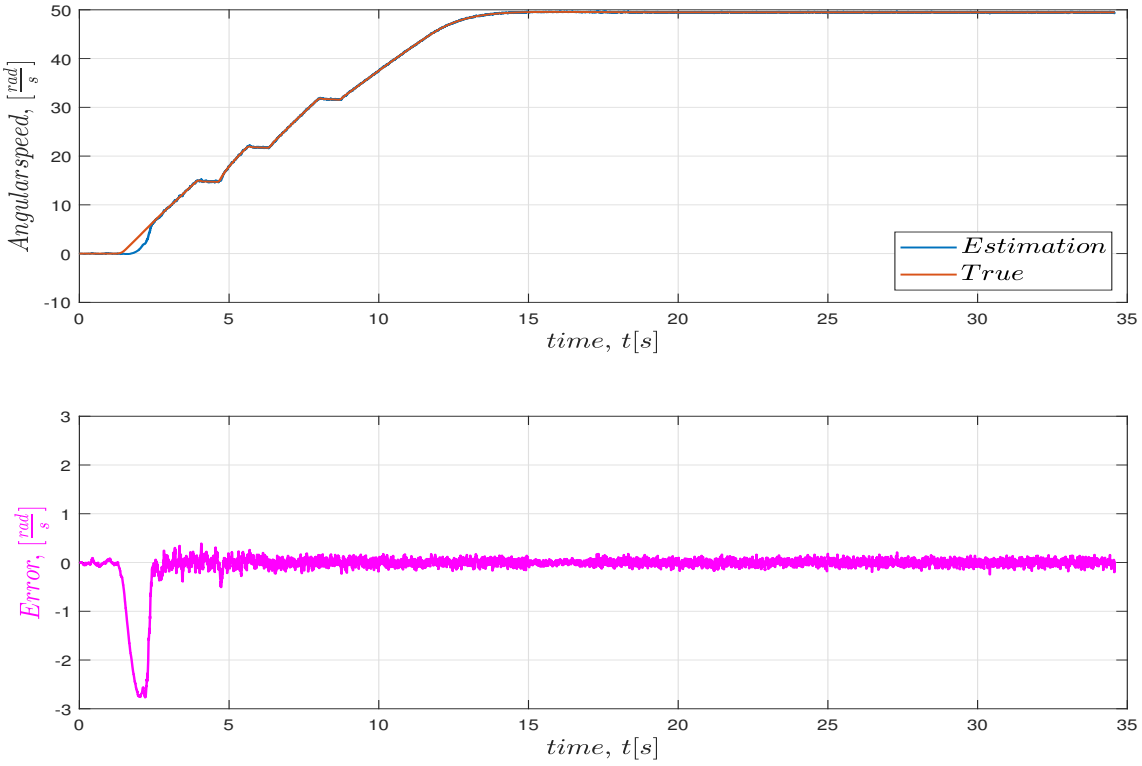


Figure B.5: Angular speed estimation versus true in straight maneuver

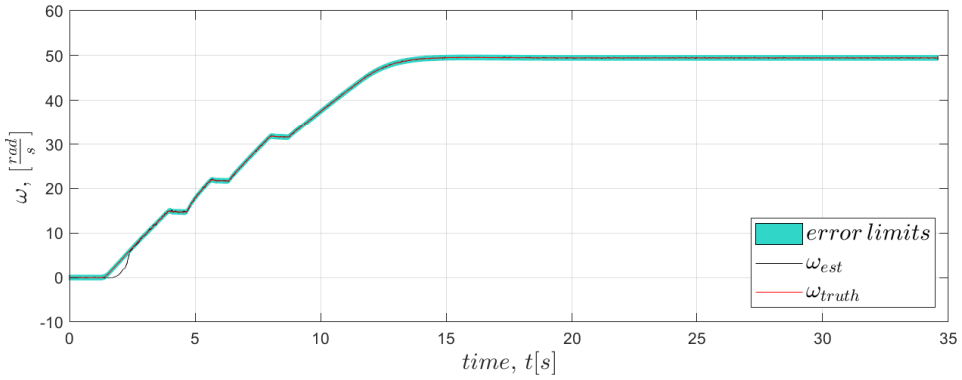


Figure B.6: Angular speed estimation with error limits in straight maneuver

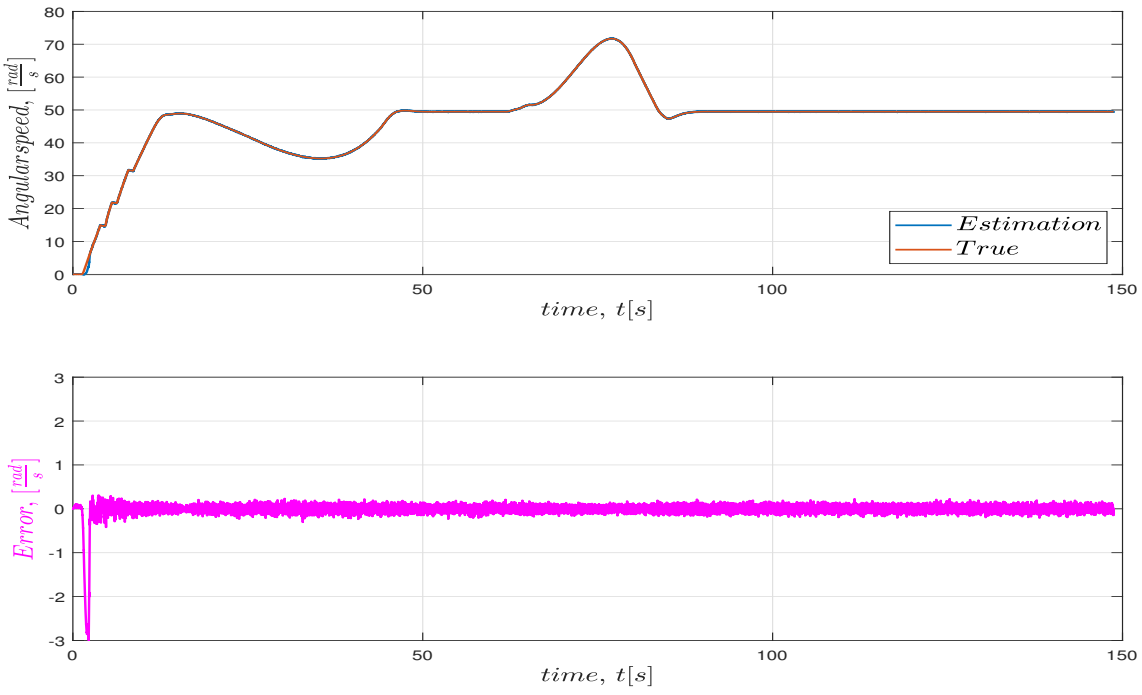


Figure B.7: Angular speed estimation versus true in straight-up-down-straight maneuver

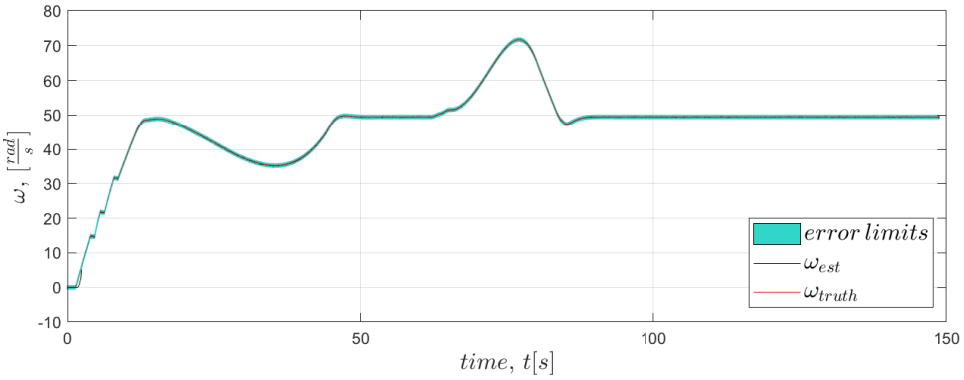


Figure B.8: Angular speed estimation with error limits in straight-up-down-straight maneuver

B.1.2 Estimation of angular displacement

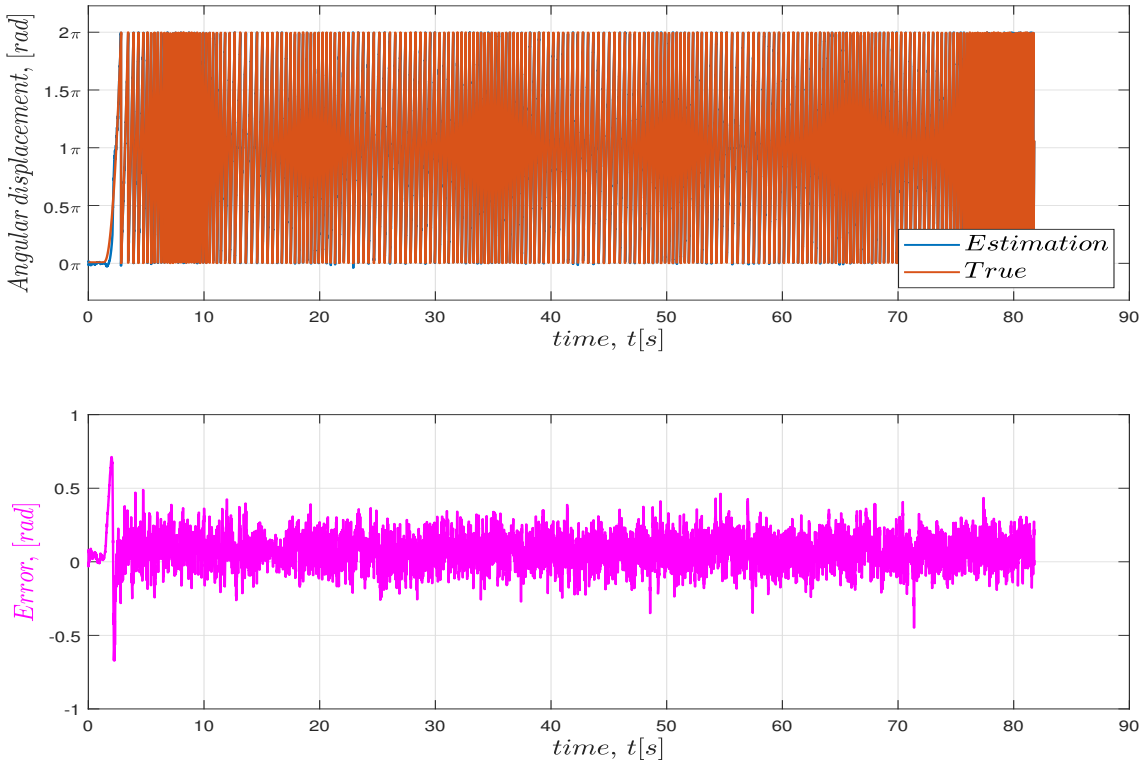


Figure B.9: Angular displacement estimation versus true in Figure8 banked road maneuver

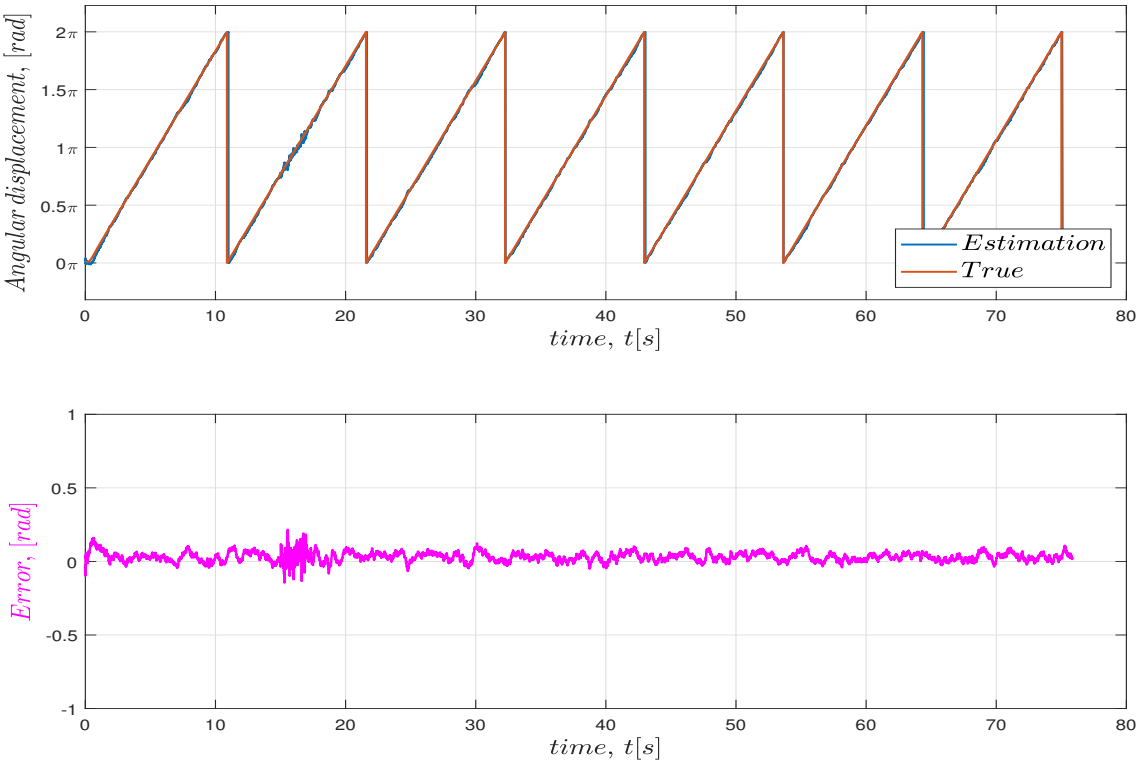


Figure B.10: Angular displacement estimation versus true in creep maneuver

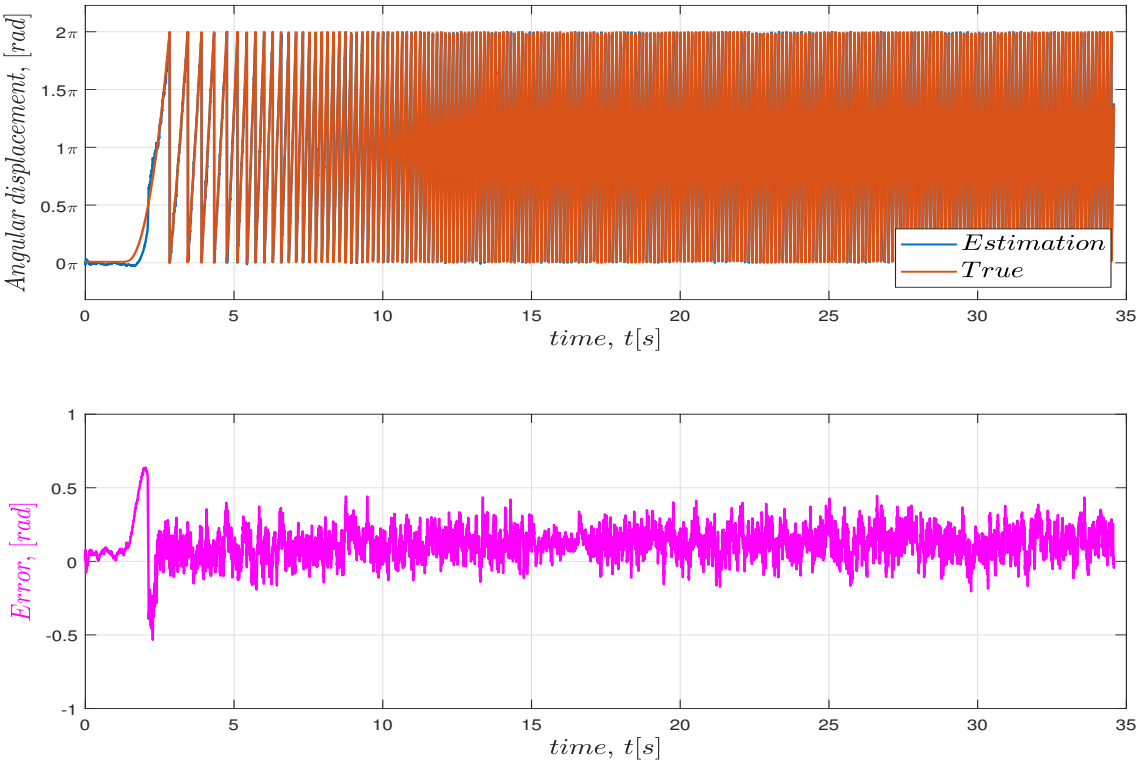


Figure B.11: Angular displacement estimation versus true in straight maneuver

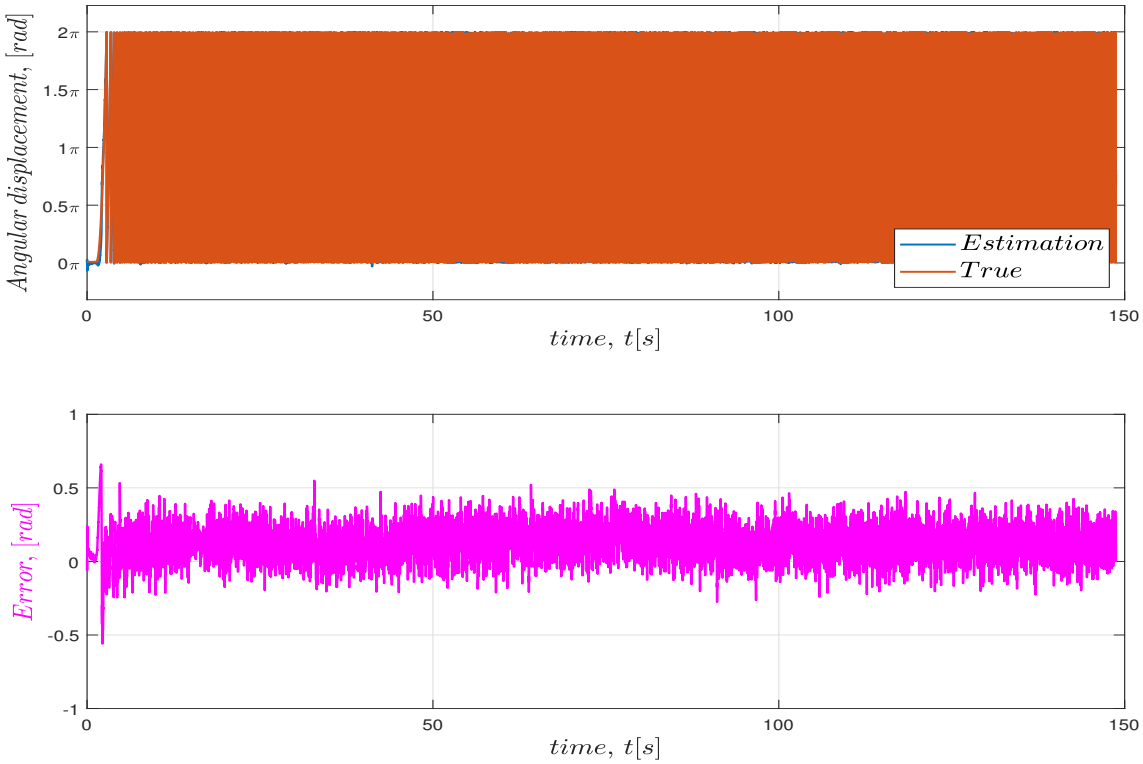


Figure B.12: Angular displacement estimation versus true in straight-up-down-straight maneuver

B.1.3 Estimation of angular acceleration

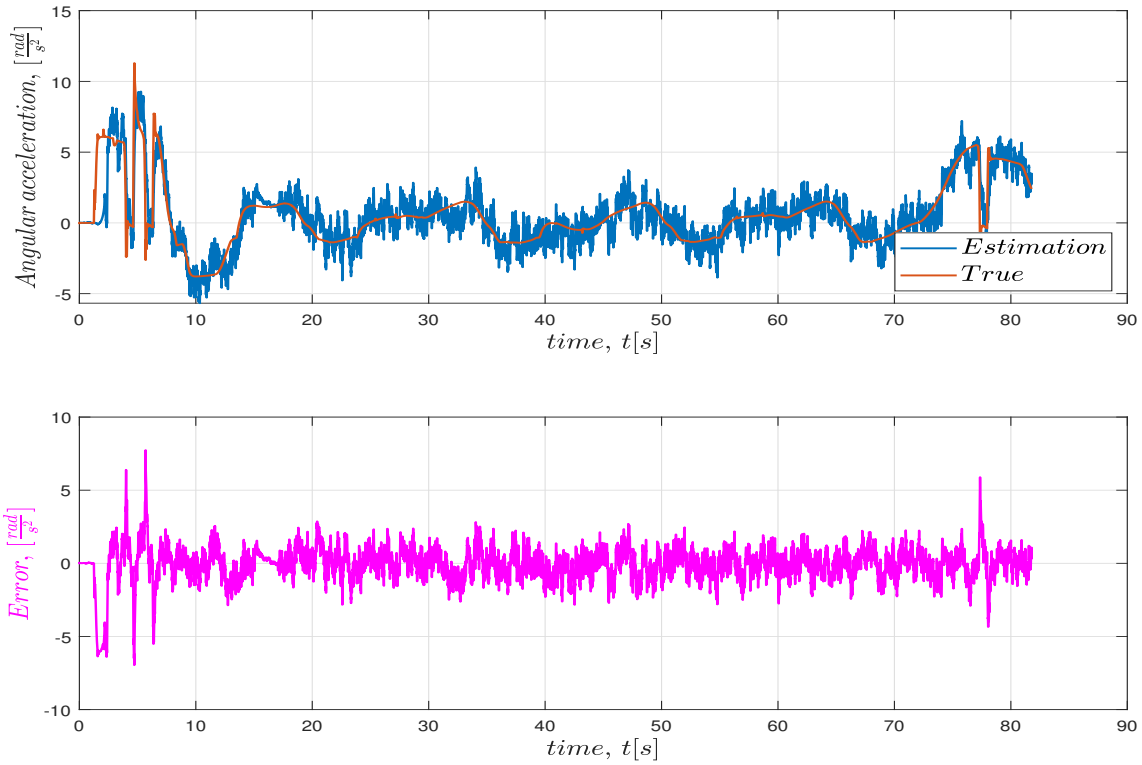


Figure B.13: Angular acceleration estimation versus true in Figure8 banked road maneuver

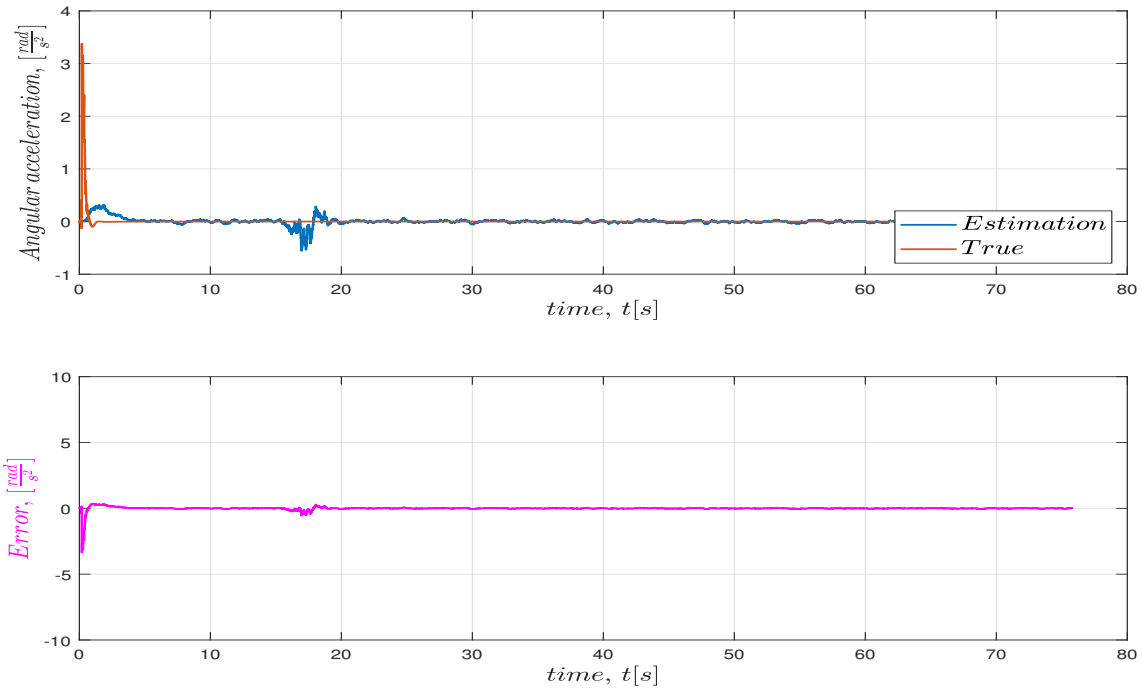


Figure B.14: Angular acceleration estimation versus true in creep maneuver

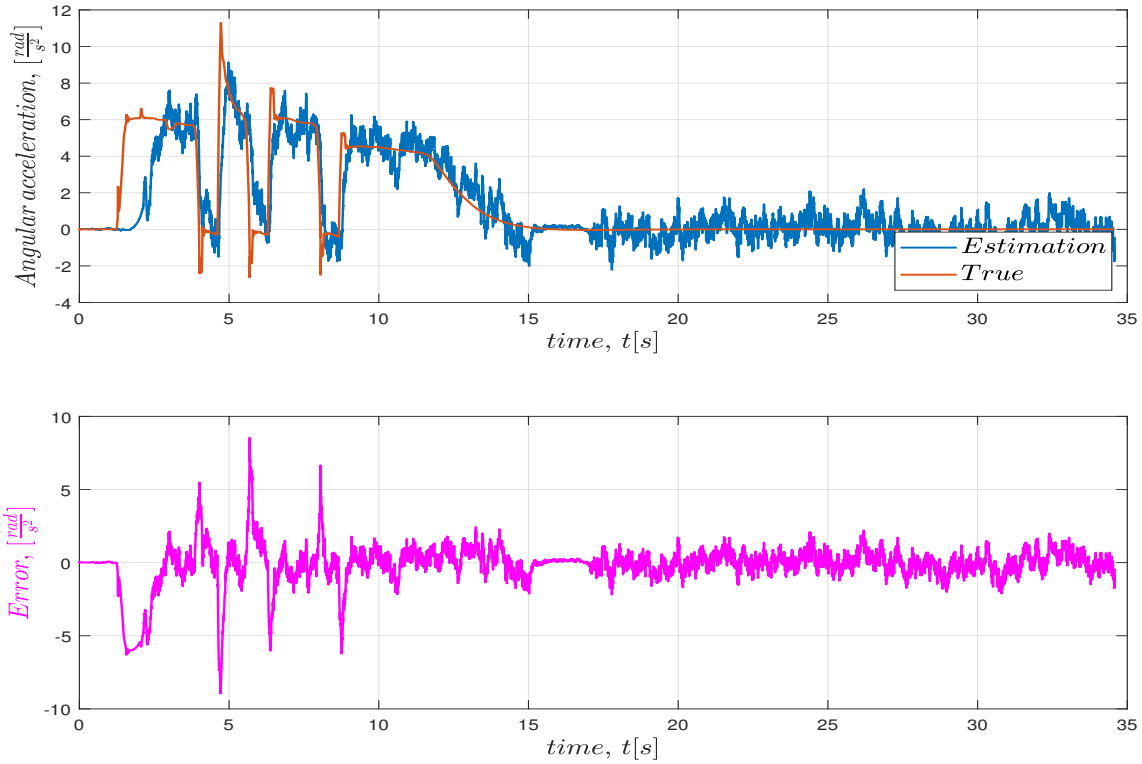


Figure B.15: Angular acceleration estimation versus true in straight maneuver

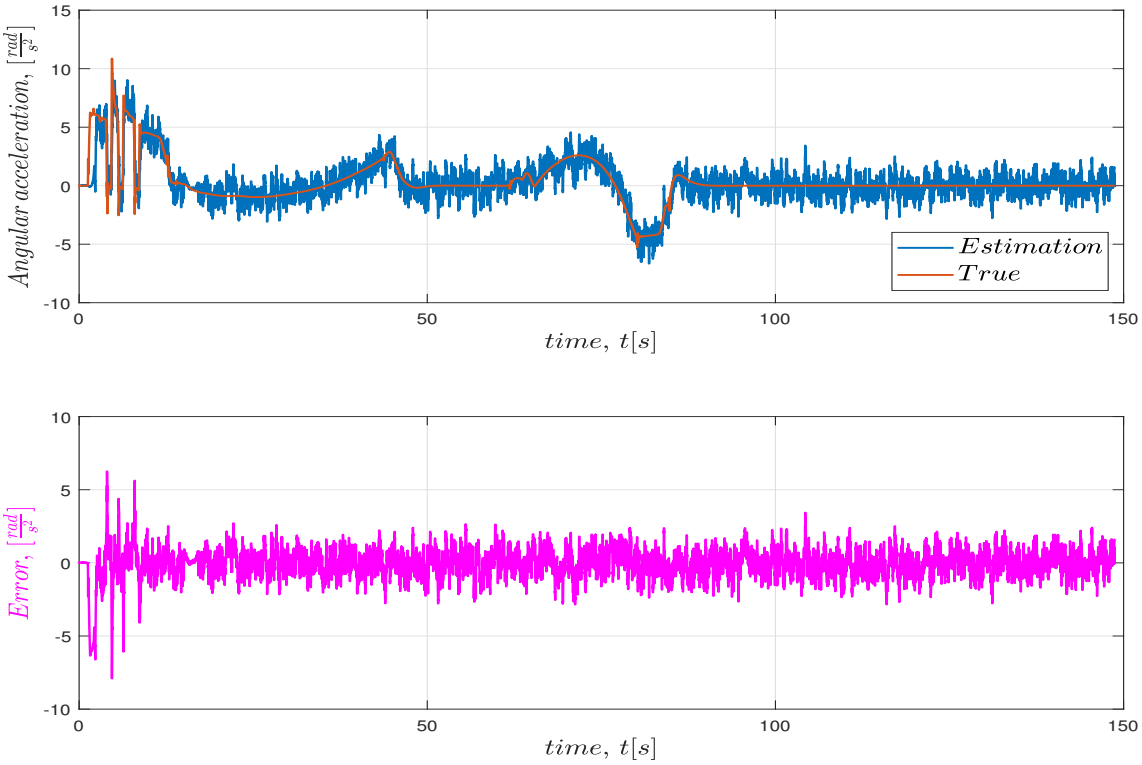


Figure B.16: Angular acceleration estimation versus true in straight-up-down-straight maneuver

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY