Institute of Mechanics and Mechatronics
Research Unit of Control and Process
Automation

Master Thesis

Experimental modelling and parameter estimation of a 1:10 scale vehicle

carried out for the purpose of obtaining the degree of Diplom-Ingenieur
(Dipl.-Ing. oder DI)

submitted at TU Wien

Faculty of Mechanical and Industrial Engineering

by

Maksym GORDIIENKO
Mat.No.: 01428819

under the supervision of

Privatdoz. Dipl.-Ing. Dr.techn. Alexander, Schirrer
Institute of Mechanics and Mechatronics

May 30, 2024

Ich habe zur Kenntnis genommen, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

DIPLOMARBEIT

nur mit Bewilligung der Prüfungskommission berechtigt bin.

Ich erkläre an Eides statt, dass die vorliegende Arbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen von mir selbstständig erstellt wurde. Alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, sind in dieser Arbeit genannt und aufgelistet. Die aus den Quellen wörtlich entnommenen Stellen, sind als solche kenntlich gemacht.

Das Thema dieser Arbeit wurde von mir bisher weder im In- noch Ausland einer Beurteilerin/einem Beurteiler zur Begutachtung in irgendeiner Form als Prüfungsarbeit vorgelegt. Diese Arbeit stimmt mit der von den Begutachterinnen/Begutachtern beurteilten Arbeit überein.

Ich nehme zur Kenntnis, dass die vorgelegte Arbeit mit geeigneten und dem derzeitigen Stand der Technik entsprechenden Mitteln (Plagiat-Erkennungssoftware) elektronisch-technisch überprüft wird. Dies stellt einerseits sicher, dass bei der Erstellung der vorgelegten Arbeit die hohen Qualitätsvorgaben im Rahmen der geltenden Regeln zur Sicherung guter wissenschaftlicher Praxis „Code of Conduct“ an der TU Wien eingehalten wurden. Zum anderen werden durch einen Abgleich mit anderen studentischen Abschlussarbeiten Verletzungen meines persönlichen Urheberrechts vermieden.

Stadt und Datum *Unterschrift*

Abstract

This thesis provides a comprehensive overview of the identification process for nonlinear tyre and drivetrain models, serving as a roadmap for understanding the selection of experiments, data preprocessing, parameters identification and validation. Precise mathematical models of vehicle dynamics play a significant role in the design, maintenance, and testing of key components, as well as in monitoring and driving assistance systems. Several manoeuvres were analyzed to determine the best approach to capture the longitudinal motion of the Research Platform for Autonomous Driving (RPAD). The linear model for the description of the longitudinal dynamics outperformed other candidates across all identification manoeuvres. Two approaches were analyzed to estimate the lateral velocity used to calculate the tyre slip angles. The simulation of the model states was proven to be the best solution to approximate lateral velocity in the absence of a measurement source. The identified vehicle model demonstrated high accuracy in predicting longitudinal and lateral accelerations, yaw angle and yaw rate during the validation stage on the reference track with velocity up to 3 m/s and front tyre slip angle values reaching up to 0.33 rad indicating a high slip region. Extended Kalman filter and particle filter based on LiDAR measurements were designed to estimate the model's accuracy in predicting RPAD's position. The steps described in various chapters of this thesis were encapsulated in the automated identification toolbox, consisting of a ROS 2 testing node and a Python identification package.

Kurzfassung

Diese Arbeit konzentriert sich darauf, einen umfassenden Überblick über den Identifikationsprozess für nichtlineare Reifen- und Antriebsstrangmodelle zu bieten und dient als Leitfaden für das Verständnis der Auswahl von Experimenten, der Datenverarbeitung, der Identifikation von Parametern und der Validierung. Die präzisen mathematischen Modelle der Fahrzeugdynamik spielen eine wichtige Rolle bei der Gestaltung, Wartung und Prüfung von Schlüsselkomponenten sowie bei Überwachungs- und Fahrassistenzsystemen. Mehrere Manöver wurden analysiert, um den besten Ansatz zur Erfassung der Längsbewegung des Research Platform for Autonomous Driving (RPAD) zu bestimmen. Das lineare Modell zur Beschreibung der Längsdynamik übertraf andere Kandidaten bei allen Identifikationsmanövern. Zwei Ansätze wurden analysiert, um die Quergeschwindigkeit zur Berechnung der Schräglaufwinkel zu schätzen. Die Simulation der Modellzustände erwies sich als die beste Lösung, um die Quergeschwindigkeit in Abwesenheit einer Messquelle zu approximieren. Das identifizierte Fahrzeugmodell zeigte eine hohe Genauigkeit bei der Vorhersage von Längs- und Querbeschleunigungen, Gierwinkel und Giergeschwindigkeit während der Validierungsphase auf der Referenzstrecke bei Geschwindigkeiten von bis zu $3\,\mathrm{m/s}$ und einem maximalen Schräglaufwinkel der Vorderreifen von bis zu $0.33\,\mathrm{rad}$, was auf einen hohen Schlupfbereich hinweist. Ein erweitertes Kalman Filter und Partikelfilter basierend auf LiDAR-Messungen wurden entwickelt, um die Genauigkeit des Modells bei der Vorhersage der RPAD-Position zu schätzen. Die in verschiedenen Kapiteln dieser Arbeit beschriebenen Schritte wurden in einem automatisierten Identifikationstool zusammengefasst, bestehend aus einem ROS 2 Testnode und einem Python Identifikationspackage.

Contents

1. Introduction

1.1. Motivation

In the world of modern vehicles, precise mathematical models of vehicle dynamics play an important role. These models find application in state monitoring and vehicle dynamics control. Autonomous driving stands at the forefront of technological advancement, reshaping mobility concepts and enhancing safety, efficiency, and convenience.

Experimental modelling and system identification form the basis of developing these accurate and reliable mathematical representations for complex systems. These modelling concepts establish a crucial link between abstract models and real-world applications by combining theoretical insights with practical data.

The identification process developed in this thesis is tested on the 1:10 scale RC car model. The same scaled vehicle model is used for the F1TENTH race [1]. The development of control algorithms for autonomous vehicles under critical driving conditions holds promise not only for the F1TENTH race but also for real-world autonomous vehicles navigating urban streets. Implementing these control algorithms requires an accurate mathematical representation of the vehicle dynamics. The parameters of the employed models should be identified systematically and reproducibly, ensuring minimal user involvement in preparation before the race.

The vehicle models developed and calibrated in this work may be used for the small-scale validation and demonstration of Model Predictive Control (MPC) algorithms developed for cooperative platooning [2] and autonomous driving [1,3–5].

1.2. Problem Statement

This thesis focuses on the methodology and identification of a model for the Research Platform for Autonomous Driving (RPAD) and its parameters. A suitable vehicle model and appropriate modelling techniques are selected based on the literature. The parameters of this model are determined through system identification, which considers available real-time measurements and hardware and software limitations. Specific test manoeuvres are defined and executed on the institute's model car, the RPAD, to gather the necessary data for parameter identification. The design and implementation of control functionalities will facilitate automated test processing. The identified vehicle dynamics model and its parameters are validated on the real hardware (RPAD) to evaluate the accuracy of the estimates.

1.3. Organisation of the thesis

A proper understanding of the investigated system is required before discussing model selections or the identification process. The starting point for further analysis is an overview of RPAD in Chapter 2. This chapter lays the foundation for the subsequent modelling and experiment selection by discussing the available sensors and hardware limitations. Since communication in the RPAD is conducted via Robot Operation System (ROS) 2, the architecture and main elements of the operating system are introduced to better understand the structure of a test node designed for automation purposes. With hardware and software limitations defined, Chapter 3 discusses the modelling approaches. The modelling is divided into two parts: the longitudinal dynamics, described primarily as a function of velocity and the duty cycle of the electric motor, and the lateral dynamics, characterised by nonlinear tyre models. Considering the limitations and the selected models, Chapter 4 introduces the design and selection of experiments. The required steps to preprocess the collected data establish a solid foundation for reliable and reproducible identification results. Different approaches to address unmeasurable system states are discussed and applied during identification. Chapter 5 serves as a critical phase in assessing the assumptions made during the experiments and the accuracy of the selected models and their identified parameters. The final chapter summarises the key findings, insights, and results and suggests potential areas for further research. The preprocessing and identification steps described in Chapter 4 were summarised in the identification toolbox consisting of the ROS 2 node and the identification package, both explained in Appendix A.

2. Hardware and Software Overview

This chapter describes the hardware and software used. First, RPAD is introduced as a key platform for experiments and identification. The focus is laid mainly on available controllers and sensors. Then, ROS 2 and its main communication components are described for better comprehension of the designed automated process presented in Appendix A.

2.1. Research Platform for Autonomous Driving

The RPAD is the institute's enhanced implementation of the original F1TENTH Autonomous Vehicle System. The RPAD main components can be categorized into three distinct levels: lower chassis, upper chassis, and autonomy elements [6]. The upper chassis acts as a connecting element, bridging all electronics on the autonomy elements level with the lower chassis.

Figure 1: Research Platform for Autonomous Driving RPAD.

The lower chassis serves as the foundation and is based on the Traxxas Slash 4x4 Ultimate 1:10 scaled race car. Equipped with a brushless Direct Current (DC) electromotor that powers four driving wheels, its low placement of the battery, electronics, and other components ensures a low centre of gravity [7].

The autonomy elements level relies on the NVIDIA Jetson Xavier NX, designed explicitly for autonomous application, making it ideal for handling visual odometry, sensor fusion, localization, and obstacle avoidance [8].

Another crucial component is the Vedder Electronic Speed Controller (VESC) 6 MKVI Electronic Speed Controller [9], responsible for controlling motor speed, current, voltage, duty cycle, etc. Additionally, it features an Inertial Measurement Unit (IMU) chip with a 9-axis gyroscope and accelerometer. Configurable parameters for gears and wheel diameters aid in estimating speed and distance travelled.

For obstacle detection, localization, and environmental awareness, the Hokuyo 10LX 2D LiDAR sensor is employed. It delivers precise measurements in a 270° field of view up to 10 meters.

In addition to these components, RPAD uses a WiFi antenna for communication, an HDMI emulator for display, and is powered by Lithium Polymer (LiPo) battery. However, to ensure the accuracy of IMU measurements, relying solely on the VESC's IMU is insufficient. Therefore, an LP-Research Motion Sensor LPMS series IMU is installed [10].

The described configuration leads to the following measurable data considering the available sensors:

- VESC: x,y-coordinates, yaw, pitch and roll angles, velocity in x-direction, yaw, pitch and roll rates, accelerations in x-y-z-directions, duty cycle of the motor, motor speed, servo position
- LP-Research IMU: yaw, pitch and roll angles and rates, accelerations in x-y-z-directions
- Hokuyo LiDAR: beam ranges between RPAD and surrounding objects and their corresponding angles

The longitudinal motion of RPAD can be manipulated by adjusting the duty cycle, motor speed, or current commands. Lateral motion depends on the servo position. To design, maintain, simulate, and control the RPAD, ROS 2 is employed. The x,y-coordinates, orientation angles and velocity in the x-direction of the VESC are only estimates based on motor speed and steering angle. The reliability of the longitudinal velocity data is checked for the specific test manoeuvre of the lateral identification.

2.2. ROS 2 Overview

An overview of ROS 2 is essential for monitoring and controlling the system states of RPAD. ROS 2 is a software platform for developing robotics applications, also known as a robotics Software Development Kit (SDK). Importantly, ROS 2 is an open-source framework distributed under the Apache 2.0 License, providing users with extensive rights to modify, apply, and redistribute the software without any obligation to contribute back [11]. ROS 2 is used in various robotics applications for simulation, control, autonomous navigation, visualization and more. Using its community-driven capabilities, it was designed to address security and reliability challenges in nontraditional environments. Compared with ROS 1, it supports Windows and macOS platforms besides Linux. Furthermore, ROS 2 introduces the ability to process multiple nodes concurrently. The communication patterns of ROS 2 are topics, actions, and services provided by nodes.

2.2.1. ROS 2 Node

A ROS 2 node is an essential unit for communicating data in the form of requests, responses (both services), actions, etc., with other nodes. It is designed to fulfil a specific modular purpose, such as publishing servo inputs or the current robot's position and orientation. ROS2 nodes can be implemented in either C++ or Python programming languages.

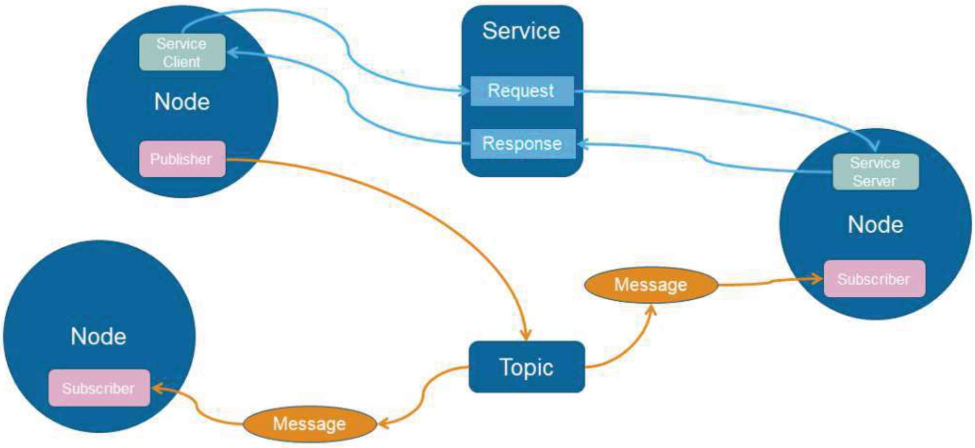

Figure 2: Visualization of ROS 2 node communication. Adapted from ROS 2 documentation [11].

To list the active nodes within the system, one can execute the following line in the bash or command line:

```
ros2 node list
```

If the desired node is not listed in the output, the following command should be executed to launch the inactive node:

```
ros2 run package_name executable_name
```

Here, `package_name` is the name of the package (node) to be run, and `executable_name` is the name of the executable function inside the package. Launching the node in a separate terminal is recommended, as the terminal becomes blocked and unusable until the node is terminated (using the Ctrl-C combination).

For additional information about a node before usage or debugging purposes, the following command returns information about subscribers, publishers, services, actions, etc., for a custom node named `lidar_node`:

```
ros2 node info lidar_node
```

```
/lidar_node
  Subscribers:
    /ego_racecar/odom: nav_msgs/msg/Odometry
    /scan: sensor_msgs/msg/LaserScan
  Publishers:
    /drive: ackermann_msgs/msg/AckermannDriveStamped
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rosout: rcl_interfaces/msg/Log
  Service Servers:
    /lidar_node/describe_parameters: rcl_interfaces/srv/Descr
    ibeParameters
    /lidar_node/get_parameter_types: rcl_interfaces/srv/GetPa
    rameterTypes
    /lidar_node/get_parameters: rcl_interfaces/srv/GetParamet
    ers
    /lidar_node/list_parameters: rcl_interfaces/srv/ListParam
    eters
    /lidar_node/set_parameters: rcl_interfaces/srv/SetParamet
    ers
  Service Clients:

  Action Servers:

  Action Clients:
```

To create a node, one must navigate to the `src` folder and execute the following command:

```
ros2 pkg create --build-type ament_python tester_node
```

Created package `tester_node` consists of following components:

- `tester_node` .... Package folder
  - `resource`
    - `tester_node` .... Marker file for the package
  - `test` .... Folder for unit testing
  - `tester_node` .... Used by ROS2 to find the package

```
│   ├── __init__.py
│   └── node_function.py ........................ Executable of the package
├── package.xml ...................... Meta information about the package
├── setup.cfg ............... Required in case the package has executable(s)
└── setup.py ................... Information about installation instructions
```

To build the new package, one should first move to the workspace folder and execute:

```
colcon build --packages-select tester-node
```

2.2.2. ROS 2 Topic

The ROS 2 topic is an asynchronous message-passing framework [11]. It is one of the main ways to transfer data between the nodes. Users can observe or send messages on a topic by creating a Subscriber or Publisher in a node, enabling one-to-one, one-to-many, many-to-one, and many-to-many communications. Each node can subscribe or publish to an arbitrary number of topics.

Similar to nodes, it is useful to gather information about available topics and their types, which determine how nodes identify that they are communicating the same data. Running the following command results in a list of topics and their message types (use the appendix option `-t`) for the RPAD:

```
ros2 topic list -t
```

```
/ackermann_cmd [ackermann_msgs/msg/AckermannDriveStamped]
/commands/motor/brake [std_msgs/msg/Float64]
/commands/motor/current [std_msgs/msg/Float64]
/commands/motor/duty_cycle [std_msgs/msg/Float64]
/commands/motor/speed [std_msgs/msg/Float64]
/commands/servo/position [std_msgs/msg/Float64]
/odom [nav_msgs/msg/Odometry]
/openzen/data [sensors_msgs/msg/Imu]
/parameter_events [rcl_interfaces/msg/ParameterEvent]
/rosout [rcl_interfaces/msg/Log]
/scan [sensor_msgs/msg/LaserScan]
/sensors/core [vesc_msgs/msg/VescStateStamped]
/sensors/imu [vesc_msgs/msg/VescImuStamped]
/sensors/imu/raw [sensors_msgs/msg/Imu]
/sensors/servo_position_command [std_msgs/msg/Float64]
/tf [tf2_msgs/msg/TFMessage]
/tf_static [tf2_msgs/msg/TFMessage]
```

To observe data published to a specific topic, such as LiDAR, use:

```
ros2 topic echo /scan
```

For publishing data to a topic using the command line, specify the message's data in the syntax of a .yaml file. It is recommended first to examine the message type using the following command:

```
ros2 show interface nav_msgs/msg/Odometry
```

```
# This represents an estimate of a position and velocity in
# free space.
# The pose in this message should be specified in the
# coordinate frame given by header.frame_id
# The twist in this message should be specified in the
# coordinate frame given by the child_frame_id

# Includes the frame id of the pose parent.
std_msgs/Header header

# Frame id the pose points to. The twist is in this
# coordinate frame.
string child_frame_id

# Estimated pose that is typically relative to a fixed world
# frame
geometry_msgs/PoseWithCovariance pose

# Estimated linear and angular velocity relative to
# child_frame_id
geometry_msgs/TwistWithCovariance twist
```

The following lines illustrate how to publish a message of type `nav_msgs/msg/Odometry`:

```
ros2 topic pub --once /odom nav_msgs/msg/Odometry '{pose:
{pose:{position:{x:0.0,y:0.0,z:0.0}}}}'
```

Here, `--once` indicates that the message will be published only once. To specify the rate of publishing the message, replace `--once` with `--rate Number`, where `Number` is the desired frequency in Hertz (Hz).

2.2.3. ROS 2 Publisher

The ROS 2 publisher is a part of the node responsible for creating and sending the data or message to a specific topic, which acts as a channel for communication.

For example, a publisher can set the motor speed or the servo position. Since communication in ROS 2 is asynchronous, the publisher sends whenever it has new data. A publisher does not know the existence of any subscriber, so it can send data to multiple subscribers and, thus, multiple nodes.

Since the publisher is a part of the node, its initialization is located in the node package, precisely, in the node executable. The custom node `tester_node` described in Section 2.2.1 will be used for demonstration purposes. The executable function of this node calls `node_function`. The central part of this file is a description of class `TesterNode`, which will be called in a loop after the node has been launched. To create a publisher for the motor controller, it is necessary to create a class attribute as an object of the Publisher class of the Node superclass:

```
self.motor_publisher = self.create_publisher( Float64,
'/commands/motor/speed', 10 )
```

The first input argument, `Float64`, is a message type to publish, which has to be first imported:

```
from std_msgs.msg import Float64
```

The second argument, `'/commands/motor/speed'`, is a topic to publish to. The third argument, 10, is optional and describes a history depth to apply to the publisher setting the limits of queued messages.

A publisher callback function should be created to publish the data more than once. A node timer function will trigger this callback. Alternatively, the publish method should be defined inside of a timer function. To create a timer function, it is necessary to initialize its class object specifying rate (in seconds) and callback function:

```
self.timer = self.create_timer(0.025,self.publisher_callback)
```

The publishing of the message happens inside of the `publisher_callback`:

```
def publisher_callback( self ):
    """Publisher function - set velocity command. """
    msg = Float64()
    msg.data = 8000.0
    self.motor_publisher.publish( msg )
```

2.2.4. ROS 2 Subscriber

The ROS2 subscriber is a component within a node responsible for listening to a specific topic and receiving messages published by other nodes. The subscriber's callback function is triggered each time a message is published.

The custom node `tester_node` described in Section 2.2.1 will be used for demonstration purposes. Similar to a publisher, a subscriber must be initialized in the node constructor:

```
self.servo_subscription = self.create_subscription(
    Float64,
    '/sensors/servo_position_command',
    self.servo_callback,
    10)
```

Here, the first argument, `Float64`, specifies the message type to subscribe to. The second argument, `'/sensors/servo_position_command'`, indicates the topic to subscribe to. The third argument, `self.servo_callback`, is a user-defined function triggered when a message is published. The last argument, 10, is an optional parameter indicating the history depth.

3. Vehicle Dynamics

A solid mathematical foundation for vehicle dynamics is essential for designing controllers, system tests and overall performance evaluation. This chapter provides an overview of various modelling approaches, shedding light on the underlying motivations behind selected models for the complete vehicle system and its distinct components. The discussion begins with examining single-track kinematic and dynamic bicycle models as the groundwork for understanding the vehicle dynamics of the RPAD. The total longitudinal force described by motor and resistance forces is introduced to describe the longitudinal dynamics of the vehicle. At the same time, the tyre model plays a significant role in capturing lateral dynamics during high slip manoeuvres. Finally, the chapter outlines the system states and their corresponding state equations, primarily in the form of nonlinear differential equations. Additionally, supporting model variants for longitudinal force (physical and data-driven) and lateral force in terms of tyre semi-empirical curves are presented.

3.1. Introduction

The application of mathematical models became a crucial part of different stages of the design and development of real hardware components. These models serve as the basis for controller design and can be utilized in critical conditions to avoid risks during simulations. The mathematical description should be accurate enough to capture vehicle dynamics and simple enough for implementation in controllers and other components, all while considering the available computation resources.

The mathematical description can be derived based on physical laws, system knowledge, or the choice of an appropriate model structure for measured input and output signals. In the first case, the parameters and terms of the model equations have physical interpretations [12]. However, due to a lack of system knowledge and details, certain model parts may only be approximated or completely ignored. On the other hand, experimental modelling does not require in-depth system knowledge, as the model structure is chosen based on input and output signals. The identified parameters are less interpretable without a clear relation between the system and the derived model. It is possible to combine both approaches in the so-called Grey-Box model, where the model structure is chosen based on theoretical knowledge of the system, while some parts of the model and parameters are identified via experimental modelling.

In general, a vehicle has six Degrees of Freedom (DoF), which can be distinguished into three translational (describing the motion in the direction of the main axis of the coordinate system) and three rotational (describing rotation around the same axis) DoF [12]. However, considering available input and output variables that can be measured to identify the model and choosing the desired accuracy or complexity, the number of DoF may be significantly reduced. For example, to describe the vehicle's longitudinal and lateral dynamics, only three DoFs are sufficient.

The choice of DoF correspondingly narrows down the selection of dynamic models. The most popular models are the single-track, dual-track, and multi-body models [12]. Combining distinct parts of each model into a hybrid one is possible to mitigate their respective disadvantages. Once the model type is chosen, subcomponents and tyre models should be determined. For a traditional vehicle, typical subcomponents are the chassis, steering, powertrain, braking system, etc. Consequently, the corresponding model inputs are, for example, acceleration or braking commands, desired steering angle, etc. In the case of the RPAD, the chosen input commands are the electric motor demand in terms of the duty cycle d to control longitudinal dynamics and the steering angle δ to control the lateral behaviour of the vehicle.

Based on the limitations introduced by the RPAD and the driving conditions for which it is utilized, it is reasonable to focus on kinematic and dynamic bicycle models. A simpler model, such as the point mass model [13], would be inadequate for describing sports-like behaviour. In contrast, more complex ones, such as the dual-track or multi-body models, cannot be provided with sufficient measurable system states.

Before going into the details of each model, the used coordinate system is introduced, see Figure 3. The Center of Gravity (CoG) of the vehicle coincides with the origin O_v of the moving vehicle's coordinate system. The position of the vehicle is described by a pair of X and Y values in the global coordinate system denoted by the sub-index O, while equations for acceleration and yaw rate will be described in the local vehicle's coordinate system denoted by sub-index v. The angle between the global coordinate system and the vehicle's coordinate system is yaw or heading angle ψ.

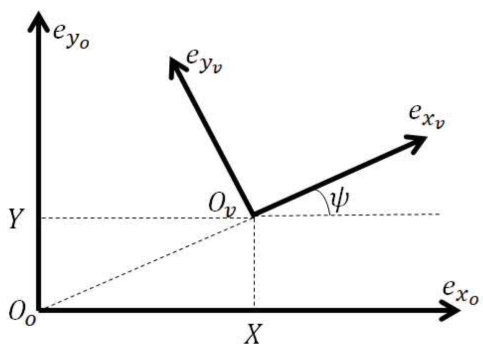

Figure 3: Coordinate systems used to describe RPAD dynamics.

3.2. Kinematic Bicycle Model

In the bicycle or single-track model, two wheels of the same axle are simplified to one central wheel [14]. The vehicle is assumed to have planar motion, allowing its motion to be described by three DoF: the position coordinates X, Y in the global coordinate system and the heading (yaw) angle ψ for orientation while pitching and rolling are neglected. Cornering is characterized by the vehicle's longitudinal axis rotation by a certain angle relative to the velocity vector v of the CoG [12]. This angle β between the velocity vector and the longitudinal axis is known as the vehicle's slip angle. The total course angle of the vehicle is then the sum of the heading and slip angles:

$$\gamma = \psi + \beta \tag{1}$$

The complete model system is summarized with the following equations [14]:

$$\dot{X} = v \cos(\psi + \beta) \tag{2}$$

$$\dot{Y} = v \sin(\psi + \beta) \tag{3}$$

$$\dot{\psi} = \frac{v \cos\beta}{l_{wb}} \tan\delta \tag{4}$$

$$\beta = \arctan\left(\frac{(l_f + l_r)\tan\delta}{l_{wb}}\right) \tag{5}$$

where l_f, l_r and l_{wb} are the distances between the front axle and the CoG, the rear axle and the CoG, the front and the rear axles (wheelbase), respectively, see Figure 4. The input to the system is the desired velocity v and the steering angle δ. However, it is possible to include a longitudinal model to derive the velocity v from a system state equation, allowing the use of the desired longitudinal acceleration a_{long} [13] or the desired duty cycle d [15] as the system input.

At higher speeds, the assumption that the velocity of the wheels aligns with the wheel's longitudinal axis is not valid, and a more detailed description is needed.

3.3. Dynamic Bicycle Model

As the velocity vector of the wheel does not align with its longitudinal axis for higher speeds, the same geometrical assumptions used to describe the vehicle's motion in Section 3.2 are no longer applicable [14]. Applying Newton's second law to the lateral direction of the vehicle yields the following equations:

$$m(\dot{v}_y + \dot{\psi}\, v_x) = F_{y_f} \cos\delta + F_{y_r} \tag{6}$$

$$\ddot{\psi}\, I_z = l_f F_{y_f} \cos\delta - l_r F_{y_r} \tag{7}$$

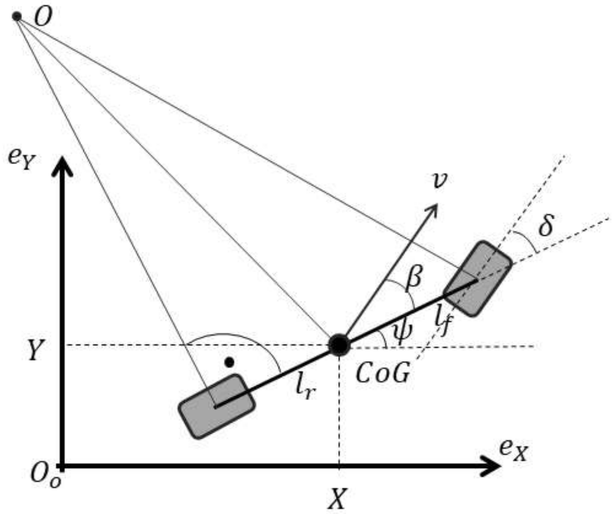

Figure 4: Kinematic bicycle model.

Here, F_{y_f} and F_{y_r} represent lateral tyre forces aligned with the e_{y_w}-axis in the corresponding local tyre coordinate system (Figure 5), which can be expressed as a function of the tyre slip angle α. The tyre slip angle is an angle between the velocity vector of the wheel v_w and the longitudinal axis of the wheel, which is aligned with e_{x_v}-axis of the vehicle (Figure 5).

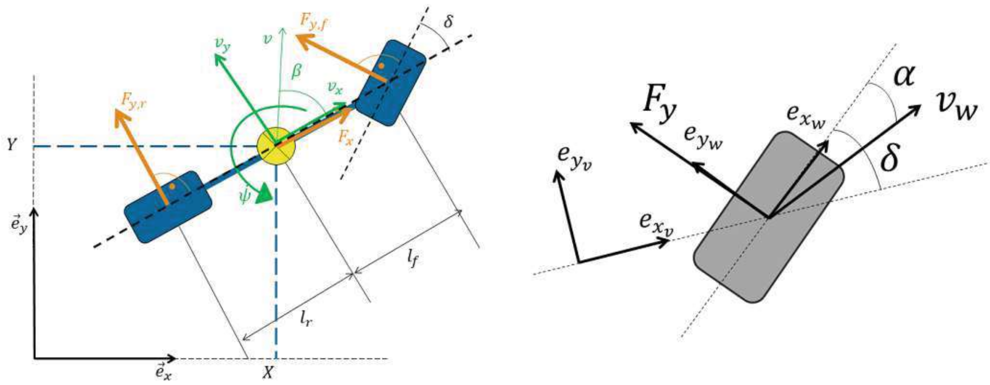

Figure 5: Dynamic bicycle model (left) and tyre lateral force as a function of the tyre slip angle (right).

Considering the geometry of the single-track model, the front and rear tyre slip angles are derived as follows:

$$\alpha_f = -\arctan\left(\frac{\dot{\psi} l_f + v_y}{v_x}\right) + \delta \tag{8}$$

$$\alpha_r = \arctan\left(\frac{\dot{\psi} l_r - v_y}{v_x}\right) \tag{9}$$

The functions that describe the dependency between lateral force and tyre slip angle are explained in Section 3.5 below. It's important to note that this model is not applicable at low-speed values due to the potential zero-division in the fraction numerator. To address this, the kinematic bicycle model can be employed for low-velocity values (as low as 0.1 m/s) [13].

To sum up, the kinematic model provides equations of motion purely in terms of geometric relationships governing the system. It is a useful model for very low-speed applications such as automated parking. In contrast, the dynamic model is useful for lane-keeping applications [14].

3.4. Longitudinal Dynamics

No wheel speed sensors are installed on the RPAD, and the vehicle speed is estimated by the VESC using a linear model based on the DC motor speed. Without a reliable velocity measurement as a key element of (13) and (14), it is not possible to exploit the slip model for describing the longitudinal dynamics. Considering the choice of the bicycle model as a basis for the description of the dynamics, both wheels of one axle are simplified to one. This neglects the difference in the wheel speed during cornering between the left and right tyres. Thus, the proper integration of the wheel speed sensor measurements into the model poses challenges. As a result, an alternative way to describe the longitudinal dynamics is needed. The description is based on the traction force while considering the loss functions, such as rolling and air drag resistances. The longitudinal velocity is expressed as a system state by

$$\dot{v}_x = F_x \frac{1}{m} + \dot{\psi}\, v_y\,, \tag{10}$$

with

$$F_x = (C_{m_1} + C_{m_2} v_x)\, d - C_r\, \mathrm{sign}(v_x) - C_d v_x^2 \tag{11}$$

being the sum of all forces acting in the longitudinal direction of the vehicle (neglecting part of the lateral force of the front tyre), which is derived using the DC motor model and the rolling and air drag resistance described in [15]. The product term $\dot{\psi}\, v_y$ in (10) considers the effects of lateral dynamics. The motor parameter

C_{m_1} is positive and describes how traction force increases with an increase of the duty cycle, while the motor parameter C_{m_2} is negative and has a damping effect opposing the increase in force with respect to increasing velocity. The parameter C_r describes the constant rolling resistance force, while being multiplied by sign function to account for the direction of the vehicle's movement and whether it is at a standstill. Finally, parameter C_d sums up all constant terms of the air drag and, like the rolling resistance coefficient, is also positive, as the minus sign is already included before the term.

Alternatively, the rolling resistance may be expressed with respect to v_x as a fourth-order polynomial [12], whereby F_z resembles the vertical load force

$$F_r = \left(C_{r_0} + C_{r_1} v_x + C_{r_2} v_x^4\right) F_z \,. \tag{12}$$

3.5. Tyre Models

Forces between the tyre and the road surface are transmitted through tyre patches [12]. The magnitude of the transmitted force depends on the corresponding tyre slip. Various approaches exist for modelling the relationship between force and tyre slip angle, with notable methods including the Burckhardt model [16], Pacejka model [17], and others. These models describe the force-slip dependency through semi-empirical curves, and their parameters are determined by applying regression techniques to attain the optimal curve fit.

3.6. Burckhardt Model

According to [16], longitudinal and lateral slip can be described for braking and acceleration, respectively, as follows:

$$\left.\begin{aligned} s_x &= \tfrac{v - r_{\text{dyn}}\,\omega\cos\alpha}{v} \\ s_y &= \tfrac{r_{\text{dyn}}\,\omega\sin\alpha}{v} \end{aligned}\right\}\text{Braking} \tag{13}$$

$$\left.\begin{aligned} s_x &= \tfrac{r_{\text{dyn}}\,\omega\cos\alpha - v}{r_{\text{dyn}}\,\omega} \\ s_y &= \sin\alpha \end{aligned}\right\}\text{Acceleration} \tag{14}$$

In these equations, v represents the velocity of the wheel centre, ω is the wheel angular speed, r_{dyn} is the dynamic radius of the wheel, and α is the slip angle of the tyre, defined as:

$$\alpha = \arctan\left(\frac{v_{y,w}}{v_{x,w}}\right) \tag{15}$$

The subscript w in (15) denotes the wheel coordinate system.

The resulting combination of longitudinal and lateral slips can be expressed as the geometrical sum using (16):

$$s_{\text{res}} = \sqrt{s_x^2 + s_y^2} \tag{16}$$

The tyre force can be determined from the resulting tyre slip using a friction-slip curve, where the friction coefficient μ is expressed as a function of slip, i.e., $\mu = f(s_{\text{res}})$. This function is dependent on surface conditions and tyre properties. According to [16], the following equation can be used to model the friction-slip curve:

$$\mu = C_1(1 - e^{-C_2\, s_{\text{res}}}) - C_3 s_{\text{res}} \tag{17}$$

Finally, the resulting tyre force can be expressed using the determined friction coefficient μ and the vertical load force F_z:

$$F_{\text{res}} = \mu\, F_z \tag{18}$$

The total force can be split into longitudinal and lateral components using the respective slip values:

$$F_x = \frac{s_x}{s_{\text{res}}} \mu\, F_z \tag{19}$$

and

$$F_y = \frac{s_y}{s_{\text{res}}} \mu\, F_z \tag{20}$$

3.6.1. Pacejka Model

A widely used semi-empirical tyre model to calculate steady-state tyre force characteristics is based on the so-called Magic Formula [17] (referred to as the Pacejka Model hereafter). The general form of the model can be described as [17]:

$$y = D \sin[C \arctan(Bx - E(Bx - \arctan(Bx)))] \tag{21}$$

with

$$Y(X) = y(x) + S_v \tag{22}$$
$$x = X + S_h \tag{23}$$

Here, the output variable Y can describe longitudinal or lateral tyre force, and the input variable X corresponds to longitudinal slip or lateral slip angle. The model introduces several parameters:

- Parameter B is the stiffness factor and determines the slope at the origin where $x = y = 0$.

- The shape factor C controls the ranges of the sine function.
- D determines the peak value.
- The curvature factor E defines the curvature at the peak.

Typically, the function goes through the origin, but horizontal shift S_h and vertical shift S_v are introduced to allow an offset [17]. For example, the offset may occur due to high wheel camber values. The curve has an anti-symmetric shape with respect to the origin $x = y = 0$ or, in the absence of shift parameters, to the origin $X = Y = 0$.

Equations to estimate initial values of the parameters are discussed in Subsection 4.6.2 of Chapter 4. The initial parameter values serve as starting values for further regression and identification of actual parameter values based on made measurements.

3.6.2. Linear Model

According to [18], the slip angle can be assumed to be small when small deviations are applied to straight-ahead motion. As a result, cornering characteristics can be linearized around the current system state. This means that the lateral force expression for both front and rear tyres (21) can be simplified to a linear model:

$$F_y = C_\alpha \alpha \tag{24}$$

where C_α represents the cornering stiffness.

3.7. Model Selection

For further analysis and identification of vehicle dynamics, the bicycle model serves as the basis for the model description. Seven system states denoted as $\boldsymbol{x} = [X,\, Y,\, \psi,\, v_x,\, v_y,\, \dot{\psi},\, \beta]^{\mathrm{T}}$ and two inputs denoted as $\boldsymbol{u} = [d,\, \delta]^{\mathrm{T}}$ for the electric motor duty cycle and the steering angle respectively, are chosen to represent longitudinal and lateral vehicle dynamics comprehensively. These states and inputs form the following system of equations:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \tag{25}$$

$$\dot{X} = v \cos(\psi + \beta) \tag{26}$$

$$\dot{Y} = v \sin(\psi + \beta) \tag{27}$$

$$\dot{\psi} = \dot{\psi} \tag{28}$$

$$\dot{v}_x = F_x \frac{1}{m} + \dot{\psi}\, v_y \tag{29}$$

$$\dot{v}_y = (F_{y,f} \cos\delta + F_{y,r}) \frac{1}{m} - \dot{\psi}\, v_x \tag{30}$$

$$\ddot{\psi} = (l_f\, F_{y,f} \cos\delta - l_r\, F_{y,r}) \frac{1}{I_z} \tag{31}$$

$$\dot{\beta} = (F_{y,r} \cos\beta + F_{y,f} \cos\beta) \frac{1}{m\, v} - \dot{\psi} \tag{32}$$

Here, X and Y represent the vehicle's position of the CoG in the global coordinate system, ψ is the heading (yaw) angle, v_x and v_y are the longitudinal and lateral velocities, respectively, in the local vehicle's coordinate system, $\dot{\psi}$ is the yaw rate and β is the slip angle of the vehicle. In addition to this, F_x represents the total longitudinal force, and $F_{y,f/r}$ represents the lateral forces at the front or rear tyres.

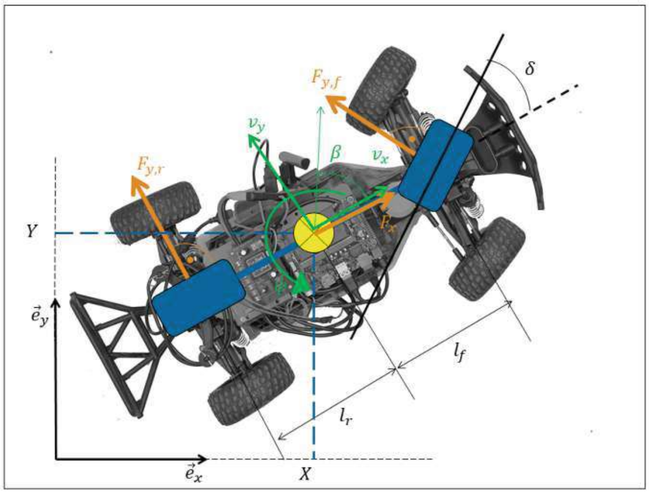

Figure 6: RPAD vehicle dynamics model.

The total longitudinal force is described with (33), as mentioned in Section 3.4. This physically motivated model captures all longitudinal forces acting on the vehicle. However, an analysis of the correlation matrix involving longitudinal acceleration a_x, longitudinal velocity v_x, duty cycle d and their products and squared terms led to the comparison of this model with a simplified version presented by Equation (34) and an extended version presented by (35). The extended model is referred to as the polynomial model, aiming to capture more features and potentially improve accuracy under certain conditions. The simplified version is referred to as the linear model, and its purpose is to avoid nonlinear terms and over-fitting. The signum function in the linear model serves as a hint for the user to consider the sign before the corresponding coefficient in case of forward or reverse driving. All three models utilize duty cycle d and vehicle velocity v_x as input arguments.

$$F_{x_1} = (C_{m_1} + C_{m_2} v_x)\, d - C_r \operatorname{sign}(v_x) - C_d v_x^2 \tag{33}$$

$$F_{x_2} = C_{m_1} d + C_{m_2} v_x - C_r \operatorname{sign}(v_x) \tag{34}$$

$$F_{x_3} = C_1 v_x^2 + C_2 v_x + C_3 v_x d + C_4 d + C_5 d^2 + C_6 \tag{35}$$

Meanwhile, three models are chosen to describe lateral tyre forces. The linear tyre model (24) is suitable for low tyre slip angle values, allowing for the linearization of tyre force characteristics. Nonlinear and more realistic tyre behaviour is then captured with the Pacejka model expressed with (36). At the same time, the reduced Pacejka model (37) offers a potential balance between accuracy and simplicity, while capturing essential non-linearities. The tyre slip angle is an input argument for all three models.

$$F_y = D \sin[C \arctan(B\alpha - E(B\alpha - \arctan(B\alpha)))] \tag{36}$$

$$F_y = D \sin[C \arctan(B\alpha)] \tag{37}$$

After identifying the model parameters, the validation process aims to determine the most suitable model choice.

4. Experiments and Identification

Understanding the dynamic parameters of a vehicle is critical to optimize its performance, ensure safety, and refine control. Longitudinal dynamics, encompassing acceleration and deceleration, play an important role in optimizing propulsion systems and designing effective braking strategies. Simultaneously, lateral dynamics, which involve the interaction between tyres and the road during steering manoeuvres, are fundamental for achieving stability and control during cornering. This chapter describes the selected experiments and the motivation underlying this choice. Furthermore, it addresses the challenges associated with non-measurable variables in the model.

4.1. Experiments Selection

The selection of appropriate experiments plays a significant role in successfully identifying the parameters of the selected models. The significance of proper selection lies in its ability to unveil valuable insights from the data. Choosing suitable experiments allows for the simplification of model equations. By tailoring experiments to specific aspects of the vehicle dynamics, certain forces, interactions and non-measurable variables can be isolated, leading to more interpretable mathematical representations. Moreover, the selected experiments must align with the available sensor set. The chosen setup must mimic the typical operation conditions of the vehicle and ensure that captured data serves for better generalizability of the identified model.

The physical space in which experiments are conducted is a limiting factor that affects the range and nature of experiments. The available space dictates the manoeuvrability of the vehicle, impacting the diversity of data that can be collected. The chosen experiments must balance obtaining comprehensive data and mitigating risks. Safety protocols and measures should be integrated into the experimental design to avoid potential vehicle collision with observers and surroundings, leading to mechanics and sensory equipment malfunction.

The number of measured variables related to the selected models in Chapter 3 is limited to yaw angle ψ, yaw rate $\dot{\psi}$, longitudinal acceleration a_x and lateral acceleration a_y. The onboard sensory system provides a longitudinal velocity estimate v_x based on motor ERPM, duty cycle d and steering angle δ. The values for longitudinal velocity may be reliable only for driving straight. In the case of lateral dynamics measurement, the workaround in terms of experiment selection should be found either to eliminate it from the model or to approximate it using simplified equations.

There is no proper source to measure lateral velocity used to calculate tyre slip angles. This limiting factor is considered for the identification of the lateral vehicle dynamics.

4.2. Preprocessing of Measurement Data

Measurement data preprocessing is a crucial step in the analysis of experimental data. This process involves cleaning, filtering, and organizing raw measurement data before it undergoes further analysis. Preprocessing enhances the quality of measurement data by addressing issues such as noise, outliers, and errors. Cleaning the data ensures that subsequent analyses are based on reliable and accurate information. Preprocessing techniques, such as filtering and smoothing, help reduce noise, provide a clearer representation of the observed dynamics, and help check or make potential model assumptions.

Identifying and removing outliers during preprocessing contributes to the robustness of subsequent analyses and the reliability of repeated identification processes. Outliers, which may result from sensor malfunctions or unexpected events, can significantly impact the accuracy of the identified model if not properly addressed - see Section 5.2.1.

Dealing with missing or incomplete data is a crucial aspect of preprocessing. Techniques such as interpolation or imputation help fill gaps in the data, ensuring that analyses are conducted on a complete dataset.

Preprocessing often involves extracting relevant features from raw data. This step helps in reducing dimensionality, which is particularly important for the identification of accurate and, at the same time, simple models.

Before going further to identification, some essential steps of preprocessing mentioned above and related to applied processes are discussed next.

4.2.1. Missing Data

Using measurement data in its raw form can lead to inaccurate results and wrong conclusions. Thorough cleaning procedures are necessary before incorporating the data into the identification process. A critical aspect of this is examining the data for missing values. When missing values are detected, two common approaches can be employed to handle them. The first approach involves discarding entire rows of data at the time step where the missing value occurs. However, this method may not be suitable if the performance evaluation of a model or a function is time-dependent. In scenarios where time is crucial, an alternative approach is to fill missing data with a default value. Possible default values include zero, the previous value, an interpolated value, or any other physically meaningful value.

In the context of RPAD, the LiDAR measurements often contained invalid data. These measurements were employed in constructing a model for the position estimation. In models where knowledge about the previous step is vital, discarding data points containing missing values is not viable. Consequently, missing values are filled with a default value. For the specified LiDAR, this default value is set to 10.0 meters, representing the maximum recognition range.

4.2.2. Plausibility Check

Ensuring the plausibility of measurement results is a crucial step in data preprocessing. It involves validating that the observed directions align with the expected behaviour in both the model and reality. For instance, in the case of steering angles, it is assumed that a positive steering angle will force the vehicle to turn in a counter-clockwise direction. Therefore, an increase in the servo position is expected to result in the wheels turning counter-clockwise. However, discrepancies may arise, as observed in RPAD. In this context, such inconsistencies can be solved in the VESC settings by adjusting the sign of the corresponding coefficient. It is essential to note that the current settings were initially configured for another project, leaving no option but to address this during data preprocessing.

In addition to this, due to space availability on the upper chassis level, the LP-Research IMU sensor was mounted upside down. Neglecting this fact during data analysis could potentially lead to a misinterpretation of the true motion direction of the RPAD. Therefore, accounting for sensor orientations and calibrations is essential to ensure the accuracy and reliability of the data.

4.2.3. Low-Pass Filter

Applying a low-pass filter to measurement data is advantageous to reduce noise or achieve a smoother representation of the underlying signal. The filtered data provides an initial insight into the dataset and assists in forming first assumptions about the system. A good low-pass filter aims to remove high-frequency components while preserving low-frequency components without distortion, effectively detecting signal changes. The Butterworth low-pass filter is suitable for such purposes [19]. The Butterworth low-pass filter is flat in the passband. It is defined in terms of the square of its transfer function. To calculate the filter numerator and denominator coefficients, one should specify the order and the cutoff frequency [19]. This may be done by analyzing the data or simply by tuning the filter coefficients. Figure 7 shows a comparison between the raw IMU measurements and the filtered values of the lateral acceleration.

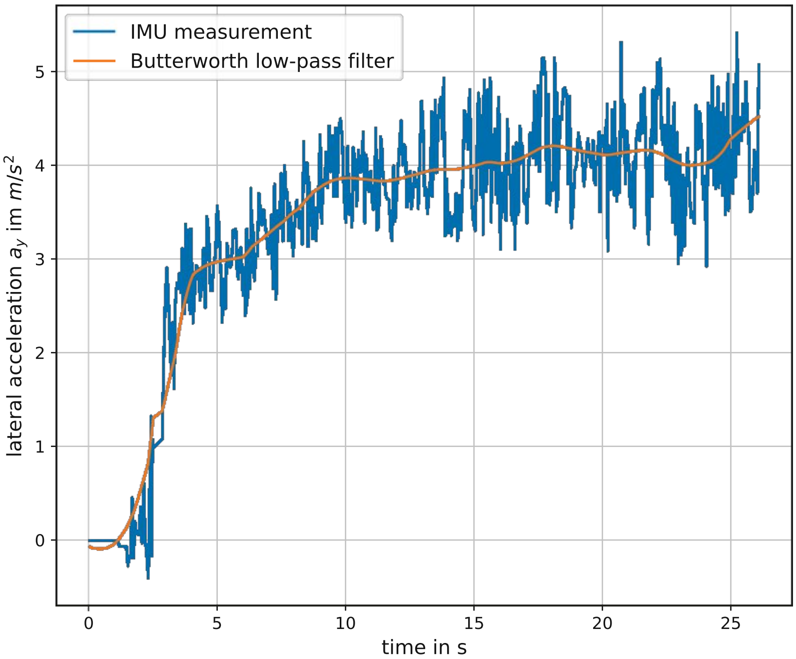

Figure 7: Comparison of the raw IMU lateral acceleration measurements and the filtered values.

The data filtered with the low-pass filter is not used for the identification process directly to avoid the loss of the information. It serves to give the user the first insight into the data.

4.2.4. Outlier Detection

Outlier detection involves identifying data points that significantly deviate from the overall pattern of the dataset. These anomalous values can arise during the measurement process or from data entry mistakes. Detecting and handling outliers is crucial to avoid the results skewing and inaccurate conclusions. Common techniques for outlier detection include statistical methods, such as z-scores and the interquartile range, or machine learning algorithms, like isolation forests and clustering-based approaches. An effective identification of the outliers ensures the reliability of the data analysis process. In this work, outlier detection is applied for the filtering of the tyre slip angle values in a constant region. A data point is identified as an outlier if its value deviates more than 1.5 standard deviations from the expected mean.

The preprocessing steps described above are considered in the automated identification tool described in Appendix A.2.

4.3. Length, Mass and Inertia Measurements

Table 1 provides the directly measured values of certain parameters, except for the moment of inertia, which required a dedicated experiment.

Parameter	**Symbol**	**Value**	**Error**	**Unit**
Length	l	0.555	0.010	m
Width	w	0.286	0.005	m
Height	h	0.160	0.020	m
Front distance to CoG	l_f	0.162	0.031	m
Rear distance to CoG	l_r	0.158	0.027	m
Wheelbase	l_{wb}	0.320	0.029	m
Wheel diameter	d_{wheel}	0.11	0.005	m
Wheel width	w_{wheel}	0.043	0.005	m
Front mass	m_f	1.75	0.130	kg
Rear mass	m_r	1.71	0.15	kg
Mass	m	3.46	0.020	kg
Moment of inertia	I_z	0.04696	0.0092	$\mathrm{kg\,m^2}$

Table 1: Invariant vehicle parameters.

The moment of inertia experiment is detailed in [20]. It employs the pendulum method without considering air resistance. The vehicle is suspended on two ropes attached to its front and rear, ensuring that the centre of gravity divides the distance between the two ropes equally. Small oscillations are induced along the yaw angle axis, and the resulting period of oscillation T is measured. The experiment is repeated 20 times to enhance accuracy.

Equation (38) estimates the parameter value using the relationship between the period of oscillation and the moment of inertia of the suspended mass, resulting in

$$I_z = \frac{mgD^2T^2}{16\pi^2 L} = 0.04696\,\mathrm{kg\,m^2} \tag{38}$$

4.4. VESC Parameters

The VESC can accurately estimate the current vehicle's longitudinal velocity, especially during straight-line driving. Given the absence of an available sensor for measuring velocity, this estimate becomes essential for control and modelling purposes. To attain this estimate, it is necessary to calibrate the odometry of the

RPAD, as outlined in [6]. The calibration process is divided into two parts, focusing on the servo and the motor. These calibrations are essential to ensure the reliability and precision of the velocity estimates obtained through the VESC.

Parameter name in the RPAD configuration	Value
steering_angle_to_servo_offset	0.515
steering_angle_to_servo_gain	-0.769
speed_to_erpm_gain	4265.77

Table 2: The VESC parameters achieved after the calibration of the RPAD odometry.

4.5. Longitudinal Model Parameters

Building upon the selection of the longitudinal model in the preceding chapter, the focus now shifts to the practical aspects of understanding and modelling the vehicle's forward motion. This section's primary goal is to choose appropriate experiments, define an objective function, and identify the parameters that define how the vehicle accelerates and decelerates. Capturing these nuances is necessary to construct reliable models. The achieved accuracy and reliability are crucial in developing effective control algorithms, enhancing safety, and improving overall performance in real-world scenarios.

Each driving scenario should serve a specific purpose, aiming to extract valuable information about the vehicle's behaviour under different longitudinal conditions.

4.5.1. Design of Experiment

The longitudinal identification process aims to characterize the vehicle's behaviour during acceleration, braking, and constant velocity. Due to the spatially limited test ground, the focus was on pairs of driving stages rather than incorporating all stages simultaneously. The following test manoeuvres were chosen:

- Coasting: the vehicle is allowed to coast with zero duty cycle until it comes to a complete stop. This manoeuvre aims to identify the initial values of the parameters unrelated to the duty cycle.
- Acceleration and constant velocity profile: acceleration to the desired motor speed and maintaining a constant velocity for several seconds. This profile is repeated for motor speeds ranging from 4000 RPM to 13000 RPM in increments of 500 RPM.

- Acceleration and braking stages: acceleration to higher motor speeds (e.g., 13000 or 15000 RPM) and subsequent deceleration to the braking motor speed of 2000 RPM aiming to capture more dynamics features.

No steering is applied during each test manoeuvre.

Moreover, combining the identification process for the lateral and longitudinal dynamics in one experiment may be advantageous. If such a solution exists, it may reduce the time needed for the identification process. The initial seconds of the quasi-steady-state manoeuvre (See Subsection 4.6.1) for the lateral dynamics parameter identification may serve as an alternative for the second manoeuvre in the list. During this part of the experiment, the constant speed is maintained after the acceleration stage.

Each driving profile will result in different parameter sets, reflecting the unique characteristics of the vehicle's response to different inputs. Comparing the performance of the model across the test manoeuvres during the validation can help to determine which experiment captures the vehicle dynamics better.

4.5.2. Identification

Before initiating the identification of model parameters, it's crucial to select an appropriate optimization goal. Similar to the approach used for finding the parameters of the Pacejka model (Subsection 4.6.1), a curve-fitting method can be applied. In this case, the output variable is a longitudinal acceleration, and the input variables are the velocity and duty cycle of the electric motor. However, considering the integration of (29), where velocity and duty cycle at the current time step are used to calculate the velocity for the next step, fitting velocity and duty cycle alone for acceleration might lead to inaccurate results during integration. Instead, an identification based on the velocity error optimization is employed. The objective function to be minimized for the model parameters is the sum of the squared velocity errors

$$v_{\text{est}}(k) = v_{\text{est}}(k-1) + dt\, f(v_{\text{est}}(k-1), u(k-1)), \tag{39}$$

$$\min_{p} \sum_{i=1}^{n} (v_{\text{meas},i} - v_{\text{est},i})^2. \tag{40}$$

Here, $v_{\text{est}}(k)$ is an estimate of the velocity for the time step k, and $v_{\text{meas}}(k)$ is an actual velocity measurement. The Ridge regularization technique is applied to the objective function to penalize the large parameter values:

$$\min_p \left(\sum_{i=1}^{n}(v_{\mathrm{meas},i} - v_{\mathrm{est},i})^2 + \alpha \sum_{j=1}^{k}(p_j^2)\right) \tag{41}$$

with α as a hyperparameter, and p_j as a one of the model parameters. The longitudinal acceleration is used to evaluate the prediction accuracy, considering the higher accuracy of the IMU sensor for acceleration measurement compared to the velocity estimation using the VESC.

Specific parameters, such as drag coefficient or rolling resistance for the physical and linear models, are constrained due to their physical meanings. Therefore, the Trust Region Reflective algorithm is applied to solve the minimization problem of the objective function, providing an efficient solution to large constrained minimization problems [21].

First, the initial values of the parameters unrelated to the duty cycle variable are identified using a coasting driving profile for physical and linear models. The polynomial model, lacking physical significance, is ignored in this case. For this profile, the vehicle is driven at a constant speed, and then the duty cycle is set to zero, forcing the vehicle to move forward by inertia until it comes to a complete stop. In this scenario, the physical model is simplified to

$$m\, a_x = -C_r\, \mathrm{sign}(v_x) - C_d\, v_x^2 \tag{42}$$

and the linear model is simplified to

$$m\, a_x = C_{m_2}\, v_x - C_r\, \mathrm{sign}(v_x). \tag{43}$$

Table 3 illustrates the negative correlation between the longitudinal acceleration and velocity, including its squared value. The correlation of the acceleration with velocity is greater than with squared velocity, implying that the term with squared velocity does not significantly contribute to the accuracy in the presence of the velocity term. However, it is not the case for (42), where the velocity term is absent. Since the correlation is negative, C_d should be positive, and C_{m_2} should be negative. This aligns with expectations, considering the negative sign for air drag force in (42) and parameter C_{m_2} representing how motor force decreases with increasing velocity. Table 4 shows the found parameters for both models.

Variable	a_x	v_x	v_x^2
a_x	1.00000	-0.95531	-0.90586

Table 3: Part of the correlation matrix between the acceleration, velocity and squared velocity.

Physical		**Linear**	
Parameter	**Value**	**Parameter**	**Value**
C_d	9.72722	C_{m_2}	-14.64054
C_r	4.85345	C_r	0.91902

Table 4: Specific parameter values of the physical (left) and linear (right) models based on the coasting measurement.

Subsequently, these initial parameters are utilized for the identification of the physical and linear model parameters during the acceleration profile measurement as starting values for optimization problem. Table 5 presents the results of the optimization, showing that the previously identified initial values of rolling resistance, air drag coefficient and motor parameter for physical and linear models are not close to the parameters identified with the minimization algorithm.

Physical		**Linear**		**Polynomial**	
Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
C_{m_1}	104.0	C_{m_1}	160.0	C_1	-2.98935
C_{m_2}	0.0	C_{m_2}	-8.76896	C_2	-5.6959
C_r	1.46454	C_r	1.23296	C_3	47.94791
C_d	2.38644			C_4	80.0
				C_5	160.0
				C_6	0.44114

Table 5: Parameter values of the physical (left), linear (middle) and polynomial (right) models based on the acceleration stage measurements (motor speed 11000 RPM).

Figure 8 displays the IMU measurement of the longitudinal acceleration and the model estimations for the training dataset. All three models exhibit similar results throughout the measurement duration.

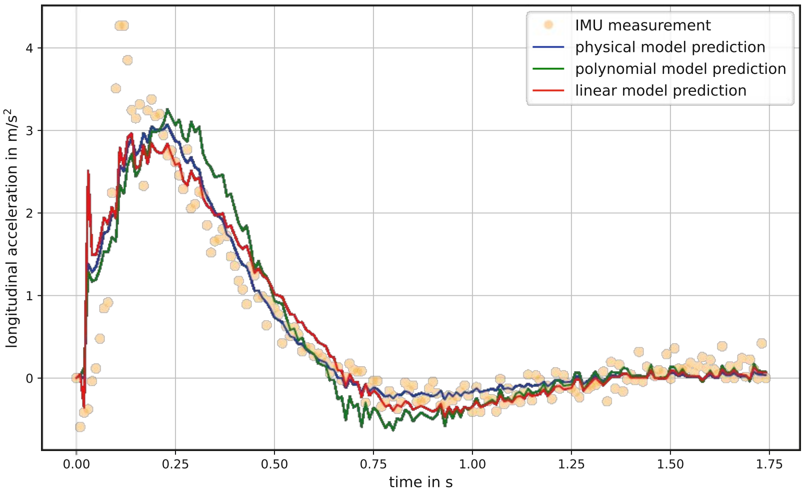

Figure 8: Comparison of the measurement data and physical, polynomial and linear models for the acceleration stage.

Next, the model parameters are identified for the test manoeuvre, which is described by the acceleration followed by the braking. Table 6 shows parameter values for this test manoeuvre. As previously, the initial values identified during the coasting manoeuvre do not correspond to the parameters found with the minimization algorithm. This suggests that the coasting manoeuvre does not sufficiently capture this part of the dynamics.

Physical		**Linear**		**Polynomial**	
Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
C_{m_1}	104.0	C_{m_1}	159.93144	C_1	-0.72341
C_{m_2}	0.0	C_{m_2}	-8.86718	C_2	-7.94118
C_r	2.10426	C_r	1.00873	C_3	16.79812
C_d	1.81916			C_4	80.0
				C_5	160.0
				C_6	2.36464

Table 6: Parameter values of the physical (left), polynomial (middle) and linear (right) models based on the acceleration-deceleration profile measurements.

Compared to the parameters shown in Table 5, the new parameter set looks similar for physical and linear models. This similarity is due to the corresponding upper bound of motor coefficient C_{m_1}. The same reason is behind the equality of C_4 and C_5 coefficients of the polynomial model. The upper bound is identified by tuning the parameters to avoid the oscillations during the integration step for high-duty cycle and low-velocity values. These upper bounds were checked for the same maneuvers, but on different surfaces and with different velocities.

Figure 9 illustrates again that the linear model captures the variations in acceleration better than the other two models. In contrast, the physical model struggles to reconstruct the braking phase well enough.

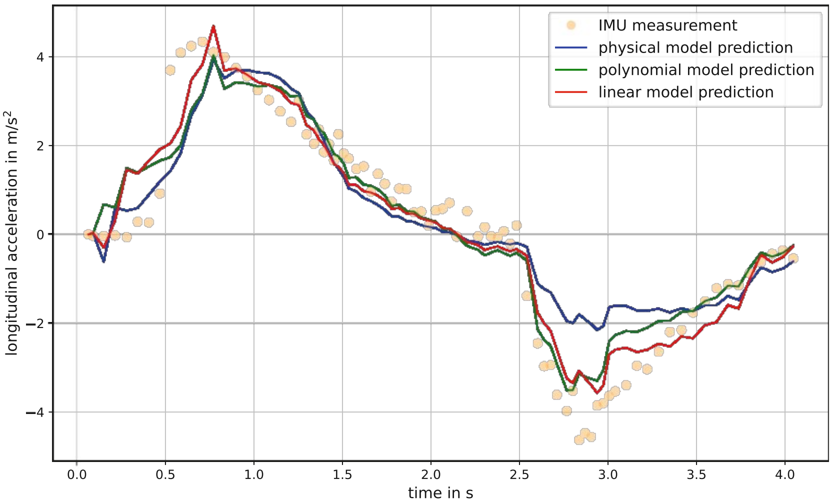

Figure 9: Comparison of the measurement data and physical, polynomial and linear for acceleration-deceleration profile.

Table 7 contains the identified parameters for each model, and Figure 10 demonstrates the comparison of IMU measurements with the model predictions for the acceleration stage of the test manoeuvre used for the lateral dynamics identification process. The only significant difference in parameter values may be observed for the rolling resistance coefficient C_r for linear and physical models. The linear model shows visually better prediction results than the other two manoeuvres. However, the final evaluation should be made during the validation.

Physical		Linear		Polynomial	
Parameter	**Value**	**Parameter**	**Value**	**Parameter**	**Value**
C_{m_1}	104.0	C_{m_1}	160.0	C_1	-3.09194
C_{m_2}	0.0	C_{m_2}	-8.86426	C_2	-11.56648
C_r	0.69779	C_r	2.2332	C_3	80.73125
C_d	2.05897			C_4	70.34928
				C_5	157.54365
				C_6	1.09621

Table 7: Parameter values of the physical (left), polynomial (middle) and linear (right) models based on the acceleration stage of lateral measurements.

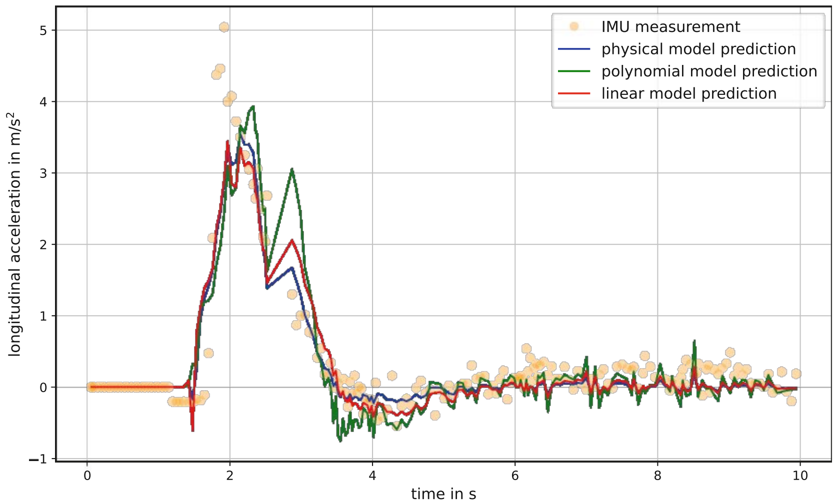

Figure 10: Comparison of the measurement data and three models for acceleration stage of lateral measurement.

In conclusion, the linear model appears to be the most promising of the three suggested models. If this assumption is proved during the validation, the polynomial function loses its value due to the higher order and non-linearity. Additionally, the coasting manoeuvre does not provide a good initial guess for the mentioned parameters and does not correspond to the final values. The validation process will provide answers regarding model choice and identification manoeuvre selection to obtain the best parameter set.

The preprocessing and identification steps are implemented in the automated identification toolbox described in Appendix A.2.

4.6. Tyre Model Parameters

In this section, the main focus is on identifying the parameters crucial for understanding the tyre behaviour, while aiming to simplify this complex process through well-designed experiments and manoeuvres. Preprocessing steps are vital to filter out the outliers of estimated tyre slip angles, ensuring reliability in the identified parameters.

4.6.1. Design of Experiment

The quasi-steady-state ramp steer manoeuvre [22] is selected for the identification of the tyre parameters in consideration of the available sensors and the goal of simplifying the model equations. In this manoeuvre, the vehicle maintains a constant velocity while the steering angle of the front wheels is gradually increased at a rate of 0.5 deg/s. This approach assumes that the vehicle is in a steady-state condition, leading to the elimination of the lateral velocity and yaw rate derivatives:

$$\dot{v}_y = 0, \tag{44}$$

$$\ddot{\psi} = 0. \tag{45}$$

As the steering angle is indirectly proportional to the curvature radius, a large curvature radius should be considered at the beginning of the test manoeuvre, when the velocity and the steering angle are small. Considering the spatially limited test ground, the front wheels are steered to 3 or 4 degrees at the beginning of the test manoeuvre. In addition, the steering velocity was also increased to 1 deg/s compared to [22]. Despite these adjustments, as shown in Figure 18, the steady-state assumption of this manoeuvre is maintained. To determine the velocity at which identified parameters yield better estimation results, the experiment is executed for motor speeds ranging from 4000 to 13000 RPM with a step size of 500 RPM.

4.6.2. Identification

Given (46) for the lateral acceleration and considering (44), the lateral acceleration can be approximated solely by the longitudinal velocity and yaw rate, leading to

$$a_y = \dot{v}_y + v_x \, \dot{\psi} = 0 + v_x \, \dot{\psi}. \tag{46}$$

This assumption allows the longitudinal velocity estimation, given the absence of other reliable sources for measuring it other than VESC. The model equations for lateral acceleration and yaw rate derivative can be simplified into a system of linear equations with two unknowns, $F_{y,f}$ and $F_{y,r}$:

$$a_y = \frac{1}{m}(F_{y,r} + F_{y,f} \cos \delta) \tag{47}$$

$$\ddot{\psi} = 0 = \frac{1}{I_z}(-F_{y,r}\, l_r + F_{y,f}\, l_f \cos \delta) \tag{48}$$

Solving the system of the algebraic equations for $F_{y,f}$ and $F_{y,r}$ results in the output variables of the tyre models:

$$F_{y,f} = \frac{l_r}{l_{wb} \cos \delta} m a_y \tag{49}$$

$$F_{y,r} = \frac{l_f}{l_{wb}} m a_y. \tag{50}$$

The issue with the tyre slip angles is more complex due to the absence of available sensors to measure the vehicle slip angle or lateral velocity needed for (8) and (9). To solve this problem, two potential solutions were found:

- Neglect the lateral velocity, identify the tyre parameters and simulate the measurements using the lateral dynamics part of the non-linear state space system introduced in Subsection 3.7. Finally, use the simulated lateral velocity together with the IMU measurements to repeat the identification. However, this solution assumes that the vehicle model used in the simulator produces accurate results.

- Utilize the longitudinal acceleration equation (51) under the assumption of a constant longitudinal velocity to approximate lateral velocity. This approach is applied starting from the specific yaw rate values ($|\dot{\psi}| > 0.1$) and after reaching the constant longitudinal velocity. Otherwise, the lateral velocity is considered to be zero.

$$a_x = \dot{v}_x - v_y \dot{\psi} \tag{51}$$

$$v_y = -\frac{a_x}{\dot{\psi}} \tag{52}$$

Regardless of the chosen solution, the calculated tyre slip angles still partly contain false data or outliers - Figure 11. Before starting the parameter identification, the tyre slip angle and lateral force data must undergo filtering. The following steps eliminate most of the false data:

- Set the lateral forces and slip angles to zero if the steering angle is zero.

- Drop the data points where the slip angle is zero, but the corresponding lateral force is not (allowed as the time variability is not important in this case).

- Drop the data where the slip angles do not have the same sign as expected for the current vehicle rotation (only one quadrant is considered).

- Drop the outliers in the last third of the slip angle data points (constant region) that are further than 1.5 standard deviations from the expected mean. Figure 11 shows the outliers detected with this approach for the front tyre slip angle values.

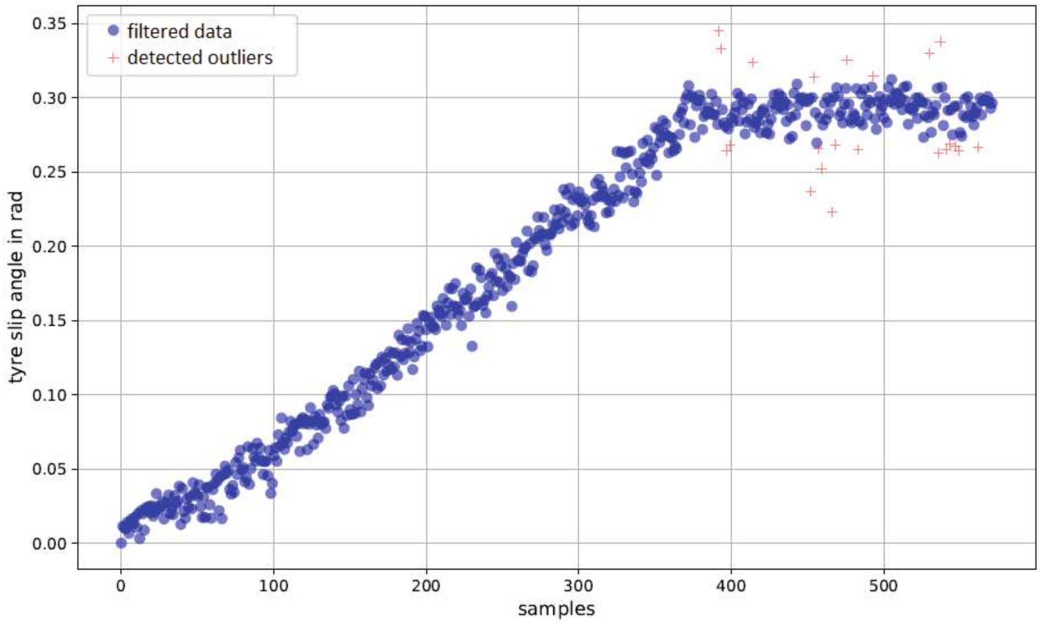

Figure 11: Visualization of the outliers for the tyre slip angle data.

After these steps, the identification process can be started.

Firstly, the initial parameter values of the Pacejka model must be found to make a first guess for the curve-fitting algorithm. According to [17], the value of D in (21) is the peak value of the lateral tyre force:

$$D = \max(F_y). \tag{53}$$

The initial value of the shape factor C can be calculated from the value of D and the horizontal asymptote of the tyre forces $f_{y,a}$:

$$C = 1 + (1 - \frac{2}{\pi} \arcsin(\frac{F_{y,a}}{D})). \tag{54}$$

Since parameter B is responsible for the stiffness factor, the slope around the origin is a suitable initial guess. The reduced model (37) is the basis for this derivation. The slip angles under 0.025 radians and their corresponding lateral forces define the area around the origin. It is important that no unexpected outlier falls into that region. This ensures that the value inside of the arcsin function in (56) is not equal to 1, which would force the value of B to jump to infinity. To find B, it has to be first expressed as a slope of the linear function based on (37):

$$y = B\alpha = \tan(\frac{1}{C} \arcsin \frac{F_y}{D}) \tag{55}$$

$$B = \frac{1}{\alpha} \tan(\frac{1}{C} \arcsin \frac{F_y}{D}). \tag{56}$$

Finally, the initial value of the curvature factor E can be calculated knowing B, C and the slip angle value at the peak force α_m with

$$E = \frac{B\alpha_m - \tan \frac{\pi}{2C}}{B\alpha_m - \arctan B\alpha_m} \tag{57}$$

if $C > 1$ or

$$E = \frac{B\alpha_m}{B\alpha_m - \arctan B\alpha_m} \tag{58}$$

otherwise.

To ensure that the parameter identification using the curve-fitting is not influenced by the unexpected outliers, the upper and lower boundaries for the parameter values are set to be in the range of ±25% of initial guess, except for the parameter D of the rear tyre, where the range is reduced to ±10% to avoid overfitting (More in Section 5.2.1). Table 8 shows the initial parameters of the front and rear tyres calculated using Equations (53)-(58).

4.6.3. Linear Model

Figure 11 indicates that the linear model (24) is applicable only within the specific ranges of the tyre slip angle. Extending the usage of this model beyond these ranges results in inaccurate predictions. However, a potential solution is to employ a saturation function, setting the maximum value of the limited region as the upper bound and its opposite value as the lower bound.

As the experiment covered various motor speeds (4000 to 13000 RPM), there

	Approximated v_y		Simulated v_y	
Parameter	**Front tyre**	**Rear tyre**	**Front tyre**	**Rear tyre**
B	22.09339	13.91822	23.10926	9.44206
C	1.41709	1.42175	1.42799	1.42799
D	10.15078	9.67439	10.15078	9.67002
E	0.89143	-1.1	0.90373	-1.1

Table 8: Initial front and rear tyre parameters for Pacejka model.

is sufficient data to approximate the linear region based on the tyre slip angle versus the lateral force figures. For the front tyre, the region is chosen between -0.05 and 0.05 radians, while for the rear tyre, it is between -0.1 and 0.1 radians. Applying these limits to the cleaned data and utilizing a linear regressor helps estimate the slope of the function. Since the function should pass through the origin (indicating no lateral force when there is no tyre slip angle), the y-intercept is set to 0. The linear regressor employs the Least Squares Method to determine the unknown slope coefficient for the specified input and output variables. As mentioned before, two approaches are applied to get its approximation without a source for lateral velocity measurement.

Table 9 compares slopes for front and rear tyres achieved using both methods. In addition to this, Figure 12 visually compares the model estimations with the filtered measurement data. Notably, the slope of both front and rear tyres is slightly lower when lateral velocity is simulated. However, the approximation using (52) cannot be applied during the acceleration stage, coinciding with the low steering angle values. This impacts the region of the low values of tyre slip angles for the front tyre (blue circles), seemingly contradicting the constraint of a zero y-intercept. On the other hand, the results look different for the rear tyre, where

	Approximated v_y		Simulated v_y	
Parameter	**Front tyre**	**Rear tyre**	**Front tyre**	**Rear tyre**
C_α	138.89766	94.83187	136.74672	92.56262

Table 9: Front and rear tyre parameters for linear model.

the correlation between the slip angle and lateral force as a linear dependency is observed. This is further supported by comparing the accuracy metrics provided in Table 10, especially when the lateral velocity is simulated. The Root Mean Square

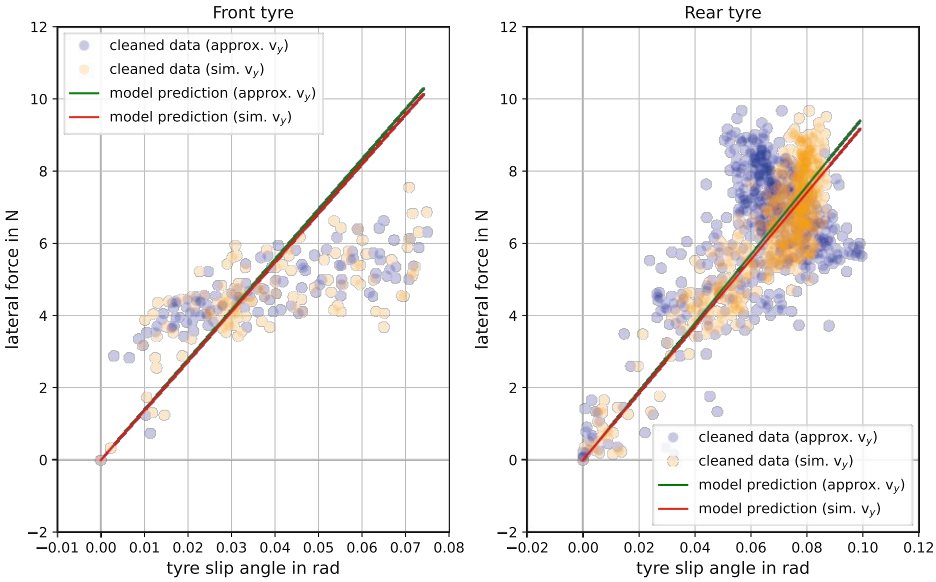

Figure 12: Comparison of the measurement data and linear model estimation for the front(left) and rear(right) tyres.

Error (RMSE) in Table 10 describes an average magnitude of the estimated error

$$\mathrm{RMSE} = \sqrt{\sum_i^n \frac{(y_i - y_{i,\mathrm{est}})^2}{N}}, \tag{59}$$

while the R-squared value (lies between 0 and 1) describes how much variation in the data is explained by the model

$$\mathrm{R}^2 = 1 - \frac{\sum_i^n (y_i - y_{i,\mathrm{est}})^2}{\sum_i^n (y_i - \overline{y})^2}, \tag{60}$$

with $\overline{y}$ as a mean value and subindex est stands for the model estimate. Since the R-squared values for both front tyre linear models are negative (not valid), the metric cannot accurately depict how well the variance of the slip angle describes the variance in lateral force, likely due to the nonlinear nature of the selected region. However, for the rear tyre, this metric is positive. In the case of the approximated lateral velocity, the R-squared value is low due to the dispersion of the data points. In contrast, simulating the lateral velocity provides more compact data points, resulting in a higher R-squared score.

	Approximated v_y		Simulated v_y	
	Front tyre	**Rear tyre**	**Front tyre**	**Rear tyre**
RMSE	1.24942	1.70536	1.12064	0.8615
$\mathbf{R^2}$	-0.7801	0.11205	-0.32133	0.75959

Table 10: Estimation accuracy metrics for training data of linear model.

Furthermore, both approaches for the estimation of the lateral velocity exhibit the same trend: the slope of the rear tyre is lower than that of the front one, leading to understeering behaviour, which may be observed during the vehicle's operation.

4.6.4. Reduced Pacejka Model

Compared to the linear model (Figure 12), the filtered measurement data for the approximated lateral velocity with (52) and the simulated lateral velocity exhibit similar trends in the nonlinear region, as shown in Figure 13. Consequently, the resulting curves for the front tyre almost coincide. However, this is not the case for the rear tyre, where the nonlinear region of the tyre slip angle is not sufficiently represented. The asymptote is much higher for the parameters based on the simulated lateral velocity, mainly due to the absence of some outliers of the slip angle values, as in the case of approximating lateral velocity. In addition, the identified parameter B, responsible for the slope near the origin, differs significantly due to the higher spread of data points in the linear region for the approximation approach. These observations are further supported by examining the estimation accuracy metrics provided in Table 12.

	Approximated v_y		Simulated v_y	
Parameter	**Front tyre**	**Rear tyre**	**Front tyre**	**Rear tyre**
B	20.70597	15.31005	19.92063	9.24421
C	1.06282	1.10076	1.07099	1.31299
D	7.46850	8.70696	7.38982	9.67002

Table 11: Front and rear tyre parameters of the reduced Pacejka model.

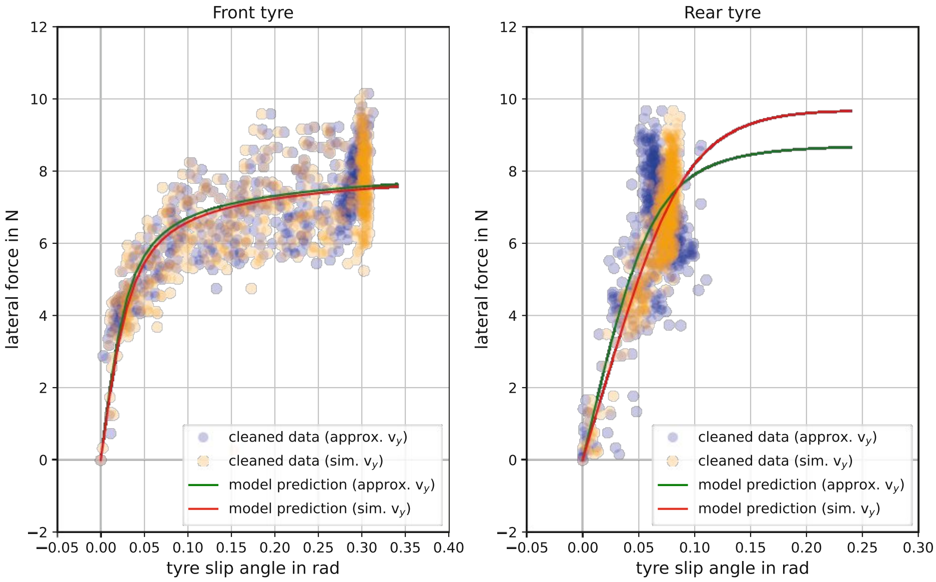

Figure 13: Comparison of the measurement data and reduced Pacejka model estimation for the front(left) and rear(right) tyres.

	Approximated v_y		**Simulated v_y**	
	Front tyre	**Rear tyre**	**Front tyre**	**Rear tyre**
RMSE	0.96729	1.35215	1.06735	0.8782
$\mathbf{R^2}$	0.59949	0.44156	0.53383	0.75017

Table 12: Estimation accuracy metrics for the training data of the reduced Pacejka model.

4.6.5. Complete Pacejka Model

The identified parameters for the complete Pacejka model are similar to the ones achieved for the reduced model, as shown in Table 13 and Figure 14. A comparison of the estimation accuracy metrics in Tables 12 and 14 indicates that the complete Pacejka model can provide slightly more accurate results, with the lower average error and better description of variance in the lateral force.

	Approximated v_y		Simulated v_y	
Parameter	**Front tyre**	**Rear tyre**	**Front tyre**	**Rear tyre**
B	23.08195	14.30502	21.04639	9.24421
C	1.06282	1.06631	1.07099	1.17231
D	8.03754	8.70696	7.9703	9.67002
E	0.84695	-0.825	0.83988	-1.37500

Table 13: Front and rear tyre parameters for the complete Pacejka model.

	Approximated v_y		Simulated v_y	
	Front tyre	**Rear tyre**	**Front tyre**	**Rear tyre**
RMSE	0.89989	1.37563	1.01522	0.86309
$\mathbf{R^2}$	0.65335	0.422	0.57825	0.7587

Table 14: Estimation accuracy metrics for the training data of the complete Pacejka model.

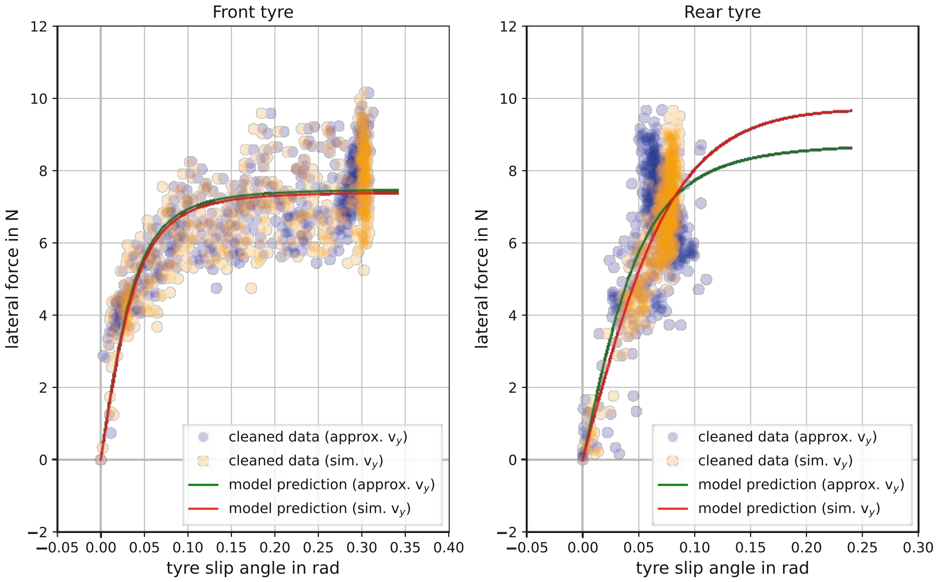

Figure 14: Comparison of the measurement data and complete Pacejka model estimation for the front(left) and rear(right) tyres.

4.6.6. Conclusion

To sum up, the choice between two approaches for estimating lateral velocity in the absence of a corresponding sensor affects the resulting slope in the low slip angle region. The final decision on the preferred approach depends on the outcomes of the validation process. Considering the conditions under which the measurements were conducted, particularly the speed, it could be assumed that utilising a linear model to characterize the behaviour of the rear tyre is more fitting. The validation phase is crucial for addressing any model accuracy and selection uncertainties.

The preprocessing and identification steps described in this chapter are implemented in the automated identification tool described in Appendix A.2.

5. Validation

Validation is a crucial phase in the pursuit of accurate vehicle dynamics models. Its significance lies in building the credibility of the derived models. It is essential to emphasize that identified models must generalize sufficiently well in the intended domain of application and avoid over-fitting the training data on which they were identified. In this chapter, selected models and their parameters are validated by comparing the estimations with data measured using available sensors. The goal is to determine which model performs better and identify experiments that provide more information about the system. Additionally, the LiDAR model is introduced as a basis for the validation of the RPAD position estimation.

5.1. Longitudinal Model

The validation of the longitudinal model aims to address two key questions:

- which manoeuvre captures more information on the underlying vehicle dynamics and
- which model yields the most accurate results on all datasets.

Four unseen measurements are used to validate the selected models and their parameters. These four validation measurements consist of

- two test manoeuvres for the acceleration followed by constant speed and
- two test manoeuvres for the acceleration followed by braking.

Considering the available hardware setup, the most reliable way to evaluate the estimation accuracy is to compare longitudinal accelerations measured with the IMU with the accelerations estimated by the model.

To assess estimation accuracy, the accuracy metrics, RMSE and R-squared value, are calculated. As explained in Subsection 4.6.2, RMSE measures the average magnitude of the prediction errors (a lower value is better) and is described by

$$\mathrm{RMSE} = \sqrt{\sum_i^n \frac{(y_i - y_{i,\mathrm{est}})^2}{N}}. \tag{61}$$

The R-squared value (lies between 0 and 1, with a higher value being better) describes how much variation in the data is explained by the model and is expressed by

$$\mathrm{R}^2 = 1 - \frac{\sum_i^n (y_i - y_{i,\mathrm{est}})^2}{\sum_i^n (y_i - \overline{y})^2}, \tag{62}$$

with $\overline{y}$ as a mean value and subindex est stands for the model estimate. The comparison involves evaluating how well different identification manoeuvres (corresponding parameter sets) described the system behaviour and determining which model type produced more accurate predictions. To assess which identification manoeuvre describes the system behaviour better, the accuracy metrics of the respective model types across the identification manoeuvres used for parameter identification are compared. To evaluate which model type yields more accurate predictions, accuracy metrics are compared across the same manoeuvre separately.

Each identification manoeuvre described in Chapter 4 is assigned a symbolic type number to describe it more compactly:

- Acceleration stage with constant speed manoeuvre - Type 1
- Acceleration and braking manoeuvre - Type 2
- Acceleration stage of lateral dynamics identification measurement - Type 3

This means that three different parameter sets corresponding to the chosen identification manoeuvres have been found for each model type.

Tables 15-17 present accuracy metrics for the physical, linear and polynomial models across different identified parameter sets. According to the average RMSE and R-squared values across validation measurements for each identification manoeuvre (parameter set) listed in the tables, the physical model achieved the most accurate results with the parameter set found during the Type 2 manoeuvre. The accuracy for this identification manoeuvre outperforms the other two only by approximately 1.4%. The accuracy metrics for Type 1 and Type 3 parameter sets do not differ significantly. Considering the typical variance of the longitudinal acceleration (IMU measurement) in the constant speed region being $0.1827\,\mathrm{m/s^2}$, the difference across the identification manoeuvres for this model is insignificant.

	Physical		**Linear**		**Polynomial**	
Recording	**RMSE**	$\mathbf{R^2}$	**RMSE**	$\mathbf{R^2}$	**RMSE**	$\mathbf{R^2}$
1	1.09479	0.78391	0.62105	0.93046	0.79959	0.88473
2	1.12455	0.75114	0.8527	0.85692	1.06499	0.77681
3	0.91965	0.82895	0.66463	0.91066	0.77186	0.87951
4	0.72026	0.85401	0.48545	0.93368	0.58106	0.90499

Table 15: Accuracy metrics of the models identified with Type 1 manoeuvre.

A similar situation holds for the linear model. However, the parameters found with all three identification maneuvers produce almost identical results. In general, the accuracy of the linear model is higher than that of a physical model, both for

	Physical		Linear		Polynomial	
Recording	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**	**R^2**
1	1.02262	0.81146	0.63372	0.92759	0.69087	0.91395
2	1.1066	0.75903	0.85623	0.85573	0.96155	0.81806
3	0.93655	0.8226	0.66134	0.91154	0.74437	0.88794
4	0.67021	0.87359	0.516	0.92507	0.48753	0.93311

Table 16: Accuracy metrics of the models identified with Type 2 manoeuvre.

	Physical		Linear		Polynomial	
Recording	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**	**R^2**
1	1.05847	0.79801	0.63162	0.92807	0.88641	0.85834
2	1.13037	0.74856	0.8481	0.85846	1.22906	0.70274
3	0.89783	0.83697	0.66243	0.91125	0.94054	0.82109
4	0.72641	0.85151	0.46957	0.93795	0.66982	0.87374

Table 17: Accuracy metrics of the models identified with Type 3 manoeuvre.

RMSE and R-squared value independent of the identification manoeuvre. On average R-squared value is about 10% higher, while RMSE is about $0.29\,\mathrm{m/s^2}$ lower.

While the polynomial model exhibits poor generalization performance and its results were inconsistent and generally less accurate, the simplicity and superior accuracy of the linear model led to the elimination of the polynomial model from further discussion.

Figures 15-17 provide visual comparisons of each model estimation using their respective parameter sets for one of the four validation measurements. These figures underscore the great performance of the linear model in capturing longitudinal dynamics across the investigated identification manoeuvres and in the generalization with the unseen data.

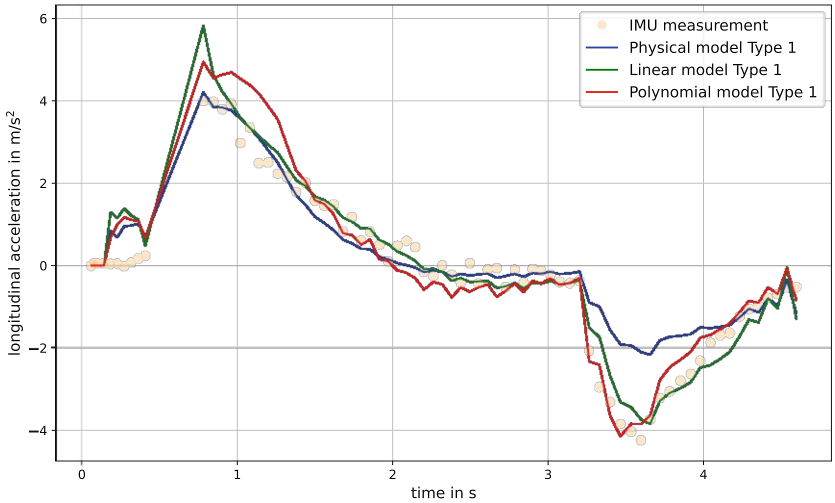

Figure 15: Comparison of IMU measurements and predictions of the model identified with Type 1 manoeuvre.

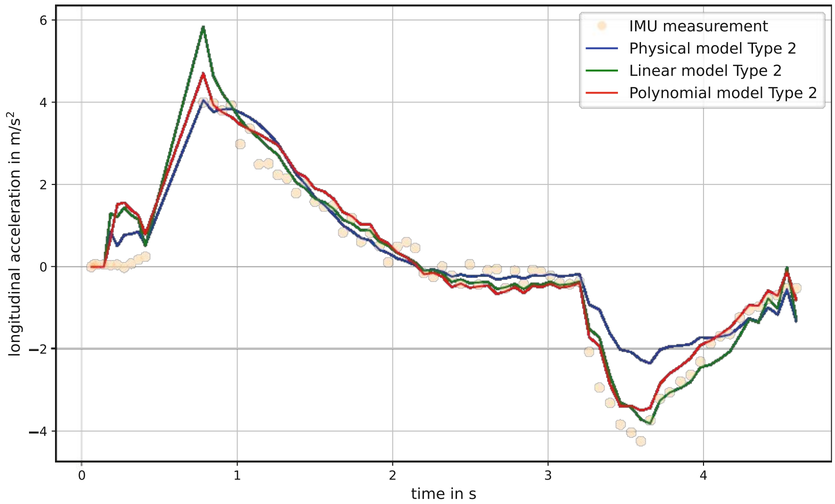

Figure 16: Comparison of IMU measurements and predictions of the model identified with Type 2 manoeuvre.

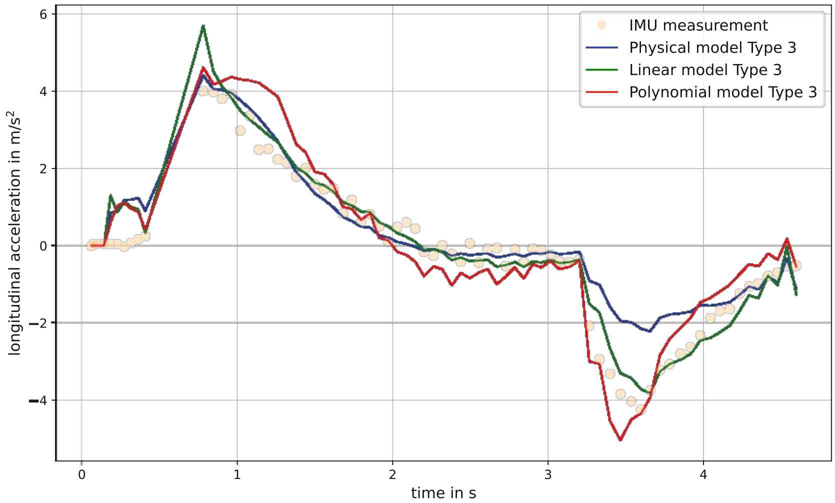

Figure 17: Comparison of IMU measurements and predictions of the model identified with Type 3 manoeuvre.

In summary, the linear model consistently outperforms the other two models in reproducing longitudinal dynamics independent of the identification manoeuvre (parameter set). The comparison of the identification manoeuvres reveals that the linear model parameters can be obtained simultaneously with lateral dynamics identification without sacrificing prediction accuracy. In contrast, the performance of the polynomial model is highly dependent on the identification manoeuvre and the model itself failed to reach its main purpose, which is the highest accuracy among all three models.

5.2. Lateral Model

In the pursuit of precision in lateral dynamics modelling, the estimation of lateral velocity plays a pivotal role. This section evaluates two distinct approaches for approximating lateral velocity for quasi-steady-state manoeuvre and subsequently compares the predictive capabilities of the resulting lateral dynamics models. Furthermore, the fundamental question has to be answered: which model yields the best results?

The tyre models use slip angle as an input, calculated using yaw rate as well as longitudinal and lateral velocities. The estimation of the longitudinal velocity made with the VESC may be unreliable for describing the lateral dynamics of the vehicle. Thus, it should be checked first using the basic assumption of quasi-steady-state manoeuvre for the lateral acceleration (46). The lateral acceleration measured with an IMU is compared to the product of the yaw rate measured with the same IMU and longitudinal velocity estimated by VESC:

$$a_{y,\mathrm{imu}} = \dot{\psi}_{\mathrm{imu}} v_{x,\mathrm{vesc}}. \tag{63}$$

Figure 18 shows close agreement between the measured lateral acceleration and the product of the measured yaw rate and the estimated longitudinal velocity.

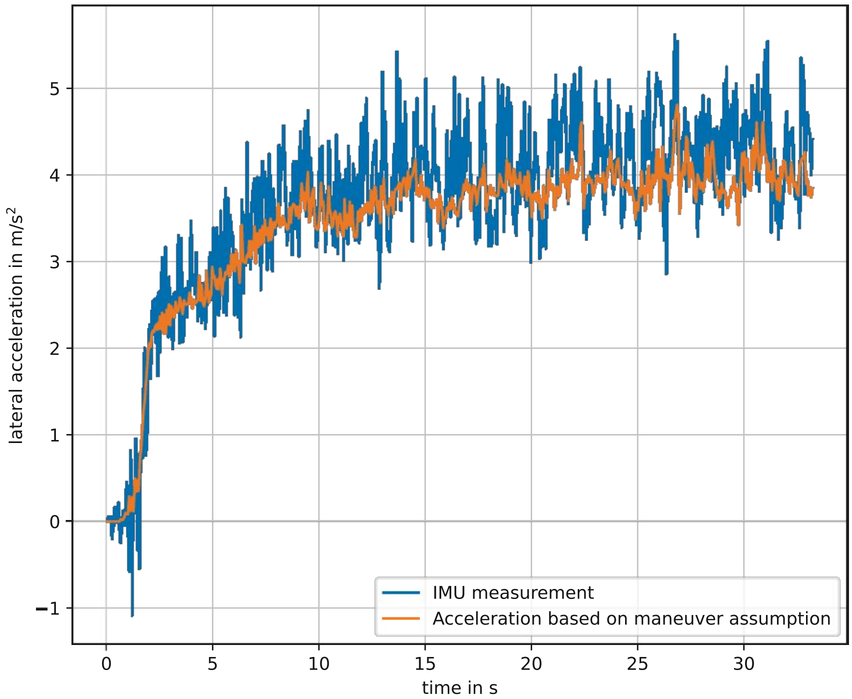

Figure 18: Comparison of the lateral acceleration IMU measurement and quasi-steady state approximation.

This means that the longitudinal velocity estimation of VESC can be used for the calculation of the tyre slip angles.

The lateral model identification process presented three front and rear tyre models and two approaches to estimate lateral velocity. Given the absence of a direct measurement source of the lateral velocity, the models and the two lateral velocity estimation approaches were subjected to validation. Four quasi-steady-state manoeuvre measurements, unseen during the model training, were employed for this purpose. Since lateral forces and slip angles cannot be directly measured, the measurable states that are directly related to the lateral dynamics are used for the validation process. RMSE and R-squared values are used as the accuracy metrics of the model predictions.

The designed simulator (Appendix A.2) was employed for the predictions using each model and parameter set. The simulator uses the duty cycle and steering angle as inputs and returns seven vehicle states and their derivatives. In order to focus only on lateral dynamics, the two states describing lateral velocity and yaw rate were considered. The inputs of the decoupled simulator are a longitudinal velocity estimated by VESC and a steering angle of the front wheels. The validation of the mentioned system states requires the simultaneous use of the front and rear tyre models.

At the first validation stage, front and rear tyres were configured with the same model and corresponding parameter set found in Subsection 4.6.2. For example, front and rear tyres are simulated with a linear model (Table 9). To simplify naming conventions, the models with parameters identified using lateral velocity estimated by (52) (after reaching constant longitudinal speed) are called Type A models. Meanwhile, the models with parameters identified using simulated lateral velocity are called Type B models.

Tables 18 and 19 show the prediction accuracy scores for yaw rate and lateral acceleration, respectively, for Type A models. In addition, Figure 19 compares one of four measurements and three model predictions. Both tables and the figure revealed that exploiting the linear model for both tyres leads to unstable results and fails to describe the system dynamics due to their parameter identification for low values of tyre slip angle and corresponding lateral forces. The oscillations occurring in the linear model predictions are a product of lateral force differences, which will be described at the end of this section. For this reason, the linear model will not be visualized to allow a better look at the reliable reduced and complete Pacejka models of both Type A and B.

	Linear		**Reduced Pacejka**		**Complete Pacejka**	
Recording	**RMSE**	$\mathbf{R^2}$	**RMSE**	$\mathbf{R^2}$	**RMSE**	$\mathbf{R^2}$
1	3.72562	-102.2148	0.1202	0.89256	0.1016	0.92323
2	3.72133	-92.56765	0.11945	0.90359	0.10429	0.92651
3	3.28089	-54.4534	0.12726	0.91656	0.1162	0.930444
4	3.44071	-75.10764	0.12082	0.90616	0.10443	0.92989

Table 18: Accuracy metrics of Type A models for the yaw rate, where both tyres use an identical model. The linear model fails to describe the system dynamics.

	Linear		Reduced Pacejka		Complete Pacejka	
Recording	**RMSE**	$\mathbf{R^2}$	**RMSE**	$\mathbf{R^2}$	**RMSE**	$\mathbf{R^2}$
1	2.35246	-2.8501	0.57936	0.76648	0.54678	0.792
2	2.32362	-2.61304	0.54697	0.7998	0.52476	0.81573
3	2.18279	-1.99833	0.50525	0.83936	0.48475	0.85213
4	2.27853	-2.4595	0.54039	0.80541	0.50669	0.82892

Table 19: Accuracy metrics of Type A models for the lateral acceleration, where both tyres use an identical model. The linear model fails to describe the system dynamics.

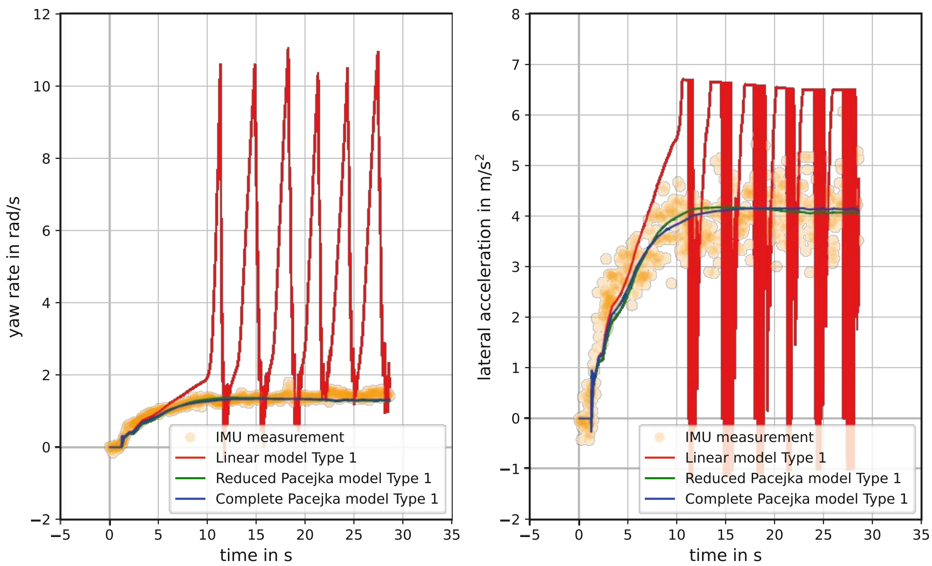

Figure 19: Comparison for yaw rate and lateral acceleration with Type A models. An occurring limit cycle in the case of the linear model can be seen.

Figure 20 demonstrates now clearly the comparison of model predictions with IMU measurements. It is evident from the figure that both models follow the dataset trends for yaw rate and acceleration. In the absence of obvious deviations in the figure, it is more beneficial to analyze RMSE and R-squared values described in Tables 18 and 19. One may notice that the RMSE values for yaw rate prediction accuracy are almost five times lower than for lateral acceleration, while R-squared values are slightly more than 10% higher. This can be explained by a larger variance of lateral acceleration data points (Figure 20). However, the accuracy scores are similar across all recordings for RMSE and R-squared values, indicating

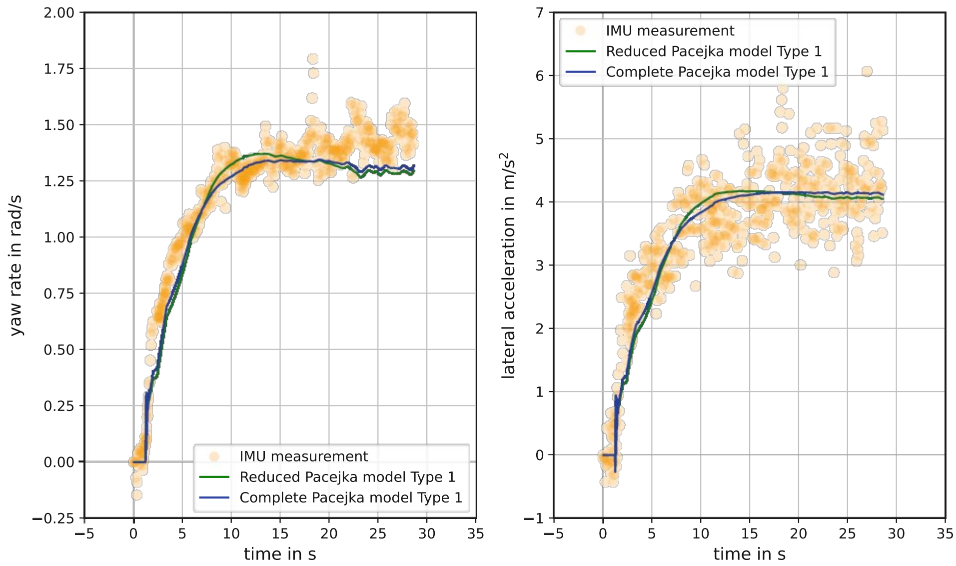

Figure 20: Comparison for yaw rate and lateral acceleration with Type A Pacejka models.

a satisfactory degree of robustness of the models, at least for the same driving profile on which the models were trained. In addition, one may notice that using the complete Pacejka model for both tyres outperforms the reduced model for all validation recordings. However, the difference does not exceed 4% on average.

On the other hand, Tables 20 and 21 describe the model prediction accuracy scores for the Type B models (simulated lateral velocity). As for Type A models, the scores do not spread wide and show stable and consistent results (Figure 21). The trend of better performance of the complete Pacejka model compared to the reduced model also holds. For every validation measurement, the complete Pacejka model outperforms the reduced model. What is more important is that Type B models achieve higher accuracy scores compared to the alternative models of Type A. This leads to the conclusion that the second approach to estimating the lateral velocity by simulating the vehicle dynamics should be used in the absence of a lateral velocity measurement source.

	Reduced Pacejka		Complete Pacejka	
Recording	**RMSE**	**R^2**	**RMSE**	**R^2**
1	0.11664	0.89883	0.09851	0.92784
2	0.11542	0.90999	0.10037	0.93193
3	0.12554	0.9188	0.11477	0.93214
4	0.11662	0.91256	0.10127	0.93406

Table 20: Accuracy metrics of Type B models for yaw rate, where both tyres use an identical model.

	Reduced Pacejka		Complete Pacejka	
Recording	**RMSE**	**R^2**	**RMSE**	**R^2**
1	0.56786	0.77566	0.53732	0.79914
2	0.53445	0.80886	0.51423	0.82305
3	0.48972	0.84908	0.4695	0.86128
4	0.5219	0.8185	0.49116	0.83925

Table 21: Accuracy metrics of Type B models for lateral acceleration, where both tyres use an identical model.

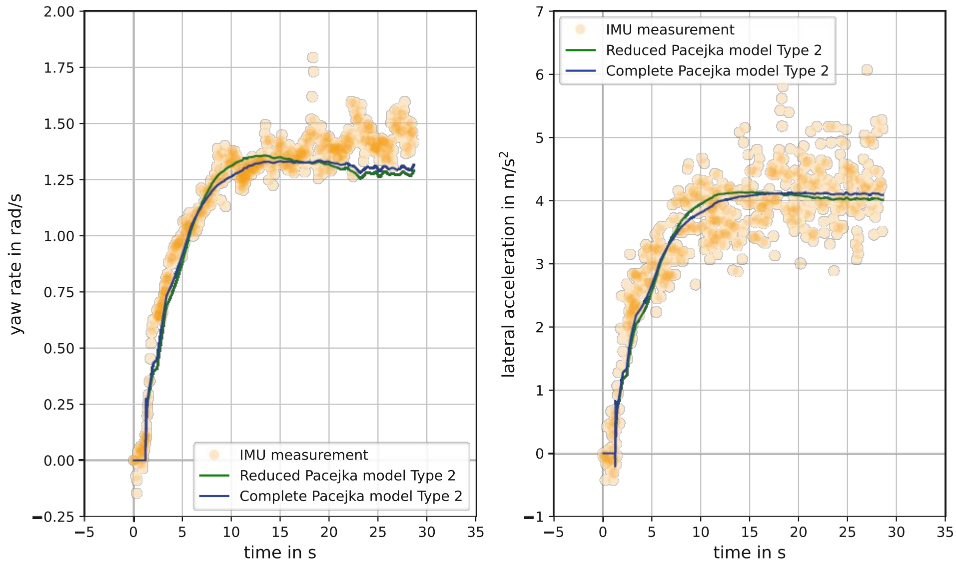

Figure 21: Comparison for yaw rate and lateral acceleration with Type B Pacejka models.

Another critical question is whether both tyres should share the same model or whether combining different models for front and rear tyres produces better results. This question essentially occurs by observing lateral force vs. slip angle figures for rear tyres (Figures 12-14). While the slip angle of the front tyre reaches values up to 0.30 radians, the values of the rear tyre barely reach 0.10 radians. It is assumed that the rear tyre does not reach saturation, so the linear model for rear tyres may fit better. The validation is now extended to check if combining Pacejka models for front tyres and linear models for rear tyres produces better model prediction accuracy.

Tables 22 and 23 show that for Type A models, the combination of a Pacejka model (reduced or complete) and a linear model produces better results than using the same Pacejka model variant for both tyres simultaneously. In addition, such a model combination produces results that are the same or sometimes even of greater accuracy than if one uses the complete Pacejka model of Type B for both tyres, which showed the best results previously.

	Reduced Pacejka		**Complete Pacejka**	
Recording	**RMSE**	$\mathbf{R^2}$	**RMSE**	$\mathbf{R^2}$
1	0.10641	0.9158	0.09217	0.93683
2	0.10439	0.92638	0.0936	0.9408
3	0.11147	0.93599	0.10397	0.94431
4	0.10436	0.92999	0.0926	0.94488

Table 22: Accuracy metrics of yaw rate with Type A Pacejka models for the front tyre and linear model for the rear tyre.

	Reduced Pacejka		**Complete Pacejka**	
Recording	**RMSE**	$\mathbf{R^2}$	**RMSE**	$\mathbf{R^2}$
1	0.5532	0.78709	0.53054	0.80418
2	0.51881	0.81988	0.50773	0.82749
3	0.46808	0.86212	0.46001	0.86683
4	0.50345	0.83111	0.4817	0.84538

Table 23: Accuracy metrics of lateral acceleration with Type A Pacejka models for the front tyre and linear model for the rear tyre.

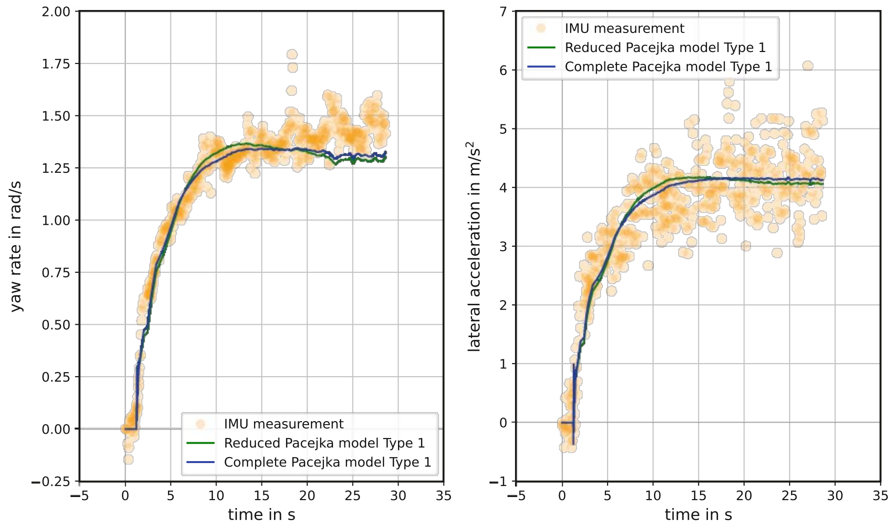

Figure 22: Comparison for yaw rate and lateral acceleration with Type A Pacejka models for the front tyre and linear model for the rear tyre.

The improvement is also seen for Type B Pacejka models - Table 24 and 25. However, this improvement is not as significant as for Type A models. Type A models now produce slightly better results in comparison with Type B. Considering the limited space during the measurement process and safety reasons, extending the velocity to higher ranges to achieve larger rear tyre slip angle values was not possible. However, the risks of using a linear model for high slip manoeuvres to describe the rear tyre behaviour outside the investigated ranges should be considered. Considering the saturation of the front tyre, the further increase of velocity may lead to an increase of the rear tyre slip angle and, as a result, to unstable and unmeasurable states of the vehicle.

	Reduced Pacejka		**Complete Pacejka**	
Recording	**RMSE**	**R^2**	**RMSE**	**R^2**
1	0.11007	0.90991	0.0956	0.93204
2	0.10922	0.9194	0.09775	0.93544
3	0.11853	0.92762	0.11161	0.93583
4	0.1088	0.9239	0.09771	0.93862

Table 24: Accuracy metrics of yaw rate with Type B Pacejka models for the front tyre and linear model for the rear tyre.

	Reduced Pacejka		Complete Pacejka	
Recording	**RMSE**	$\mathbf{R^2}$	**RMSE**	$\mathbf{R^2}$
1	0.55424	0.78629	0.53112	0.80375
2	0.51975	0.81922	0.50729	0.82779
3	0.47071	0.86057	0.46011	0.86678
4	0.50208	0.83202	0.48137	0.8456

Table 25: Accuracy metrics of lateral acceleration with Type B Pacejka models for the front tyre and linear model for the rear tyre.

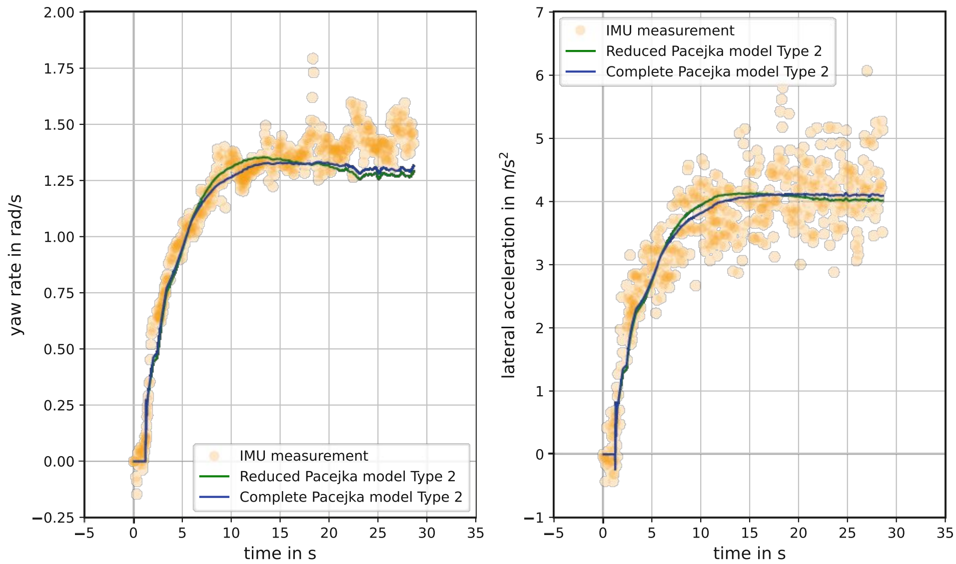

Figure 23: Comparison for yaw rate and lateral acceleration with Type B Pacejka models for the front tyre and linear model for the rear tyre.

To summarize, using a simulator to estimate lateral velocity (Type B models) yields slightly better results overall. Based on the investigated data ranges for tyre slip angles, it is reasonable to use the complete Pacejka model for the front tyre and the linear model for the rear tyre. However, to mitigate risks for more extreme operating conditions, it is strongly advised to use the Pacejka model for both tyres.

5.2.1. Simulation Oscillations and Importance of Sufficient Preprocessing

As was mentioned in Section 4.6.2, preprocessing of the raw measurement data is a crucial step prior to the parameter identification to achieve an accurate and reliable model. On the one hand, the data for the identification should be prepared without losing vital information or at least reducing information loss to a minimum. On the other hand, the outliers and undesirable deviations should be filtered out to avoid undesired over-fitting of the model to problematic or flawed data points. The identification process described in the previous chapter results from continuous improvements, data analysis, trials and errors. It shows the consistent and reproducible identification results in the presence of noise and unexpected outliers. Ignoring preprocessing or neglecting it to some degree can lead to issues such as the oscillations illustrated in Figure 19, but not only for the linear models.

Subsection 4.6.2 describes how to find the initial values of parameters for the Pacejka model. These initial values serve as a starting point for subsequent curve-fitting algorithms. The algorithm minimizes the squared error between the model predictions and the training data by adjusting the model parameters. In the case of the rear tyre, slip angle outliers can result in over-fitting if not filtered out beforehand. Figure 24 shows how several outliers of the rear tyre slip angles force the identified function to curve in their direction, leading to model over-fitting. Since it is evident from the figure that the rear tyre does not reach saturation in lateral force, the identified model curvature is wrong.

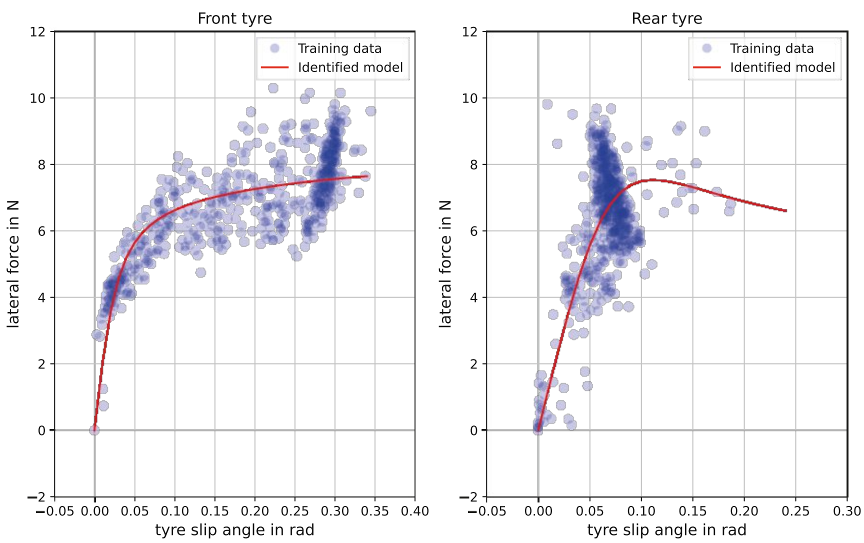

Figure 24: Comparison of training data and model prediction with complete Pacejka model.

Figure 25 shows the rear tyre slip angle evolution and the identified outliers. These outliers could not be dropped during filtering based on the physical plausibility, so statistical filtering was applied. A data point is identified as an outlier and is filtered out if its value deviates more than 1.5 standard deviations from the expected mean. The impact of this filtering on the model's curvature is demonstrated in Figure 26. This approach ensures a more accurate representation of the tyre behaviour, particularly for the rear tyre.

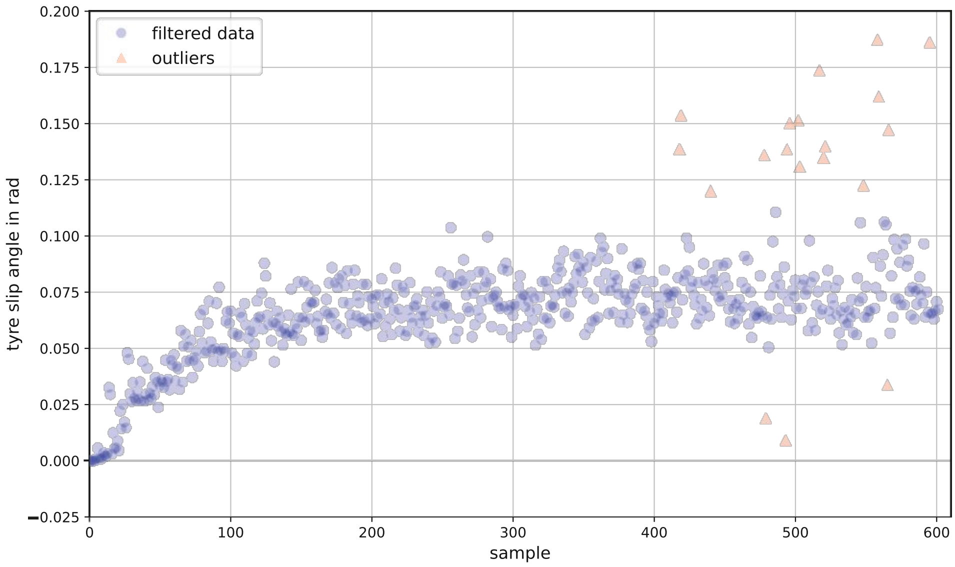

Figure 25: Filtered rear tyre slip angle values and outliers.

The front tyre is also present in the figures to show how the lateral force spreads by high slip angle values. Since the rear tyre does not reach saturation, it is impossible to identify or estimate the maximum lateral force it can achieve. The highest estimated lateral force value (blue circles in Figure 26) is then taken as an initial value, and the lower boundary is set to 90% of it. At the same time, such a boundary is not acceptable for the front tyre because of the mentioned spread of lateral forces and the reached saturation. The lower boundary is reduced to 75% for the front tyre. This combination ensures the desired characteristic curve shape of identified models for both tyres. In the case of the unreached saturation of the rear tyre, the expected shape is a shape of the monotonously increasing function.

Neglecting these steps may lead to oscillation problems during simulation, especially for higher tyre slip values. Figure 27 illustrates such undesired behaviour. The same oscillations occurred during the validation of linear models earlier in

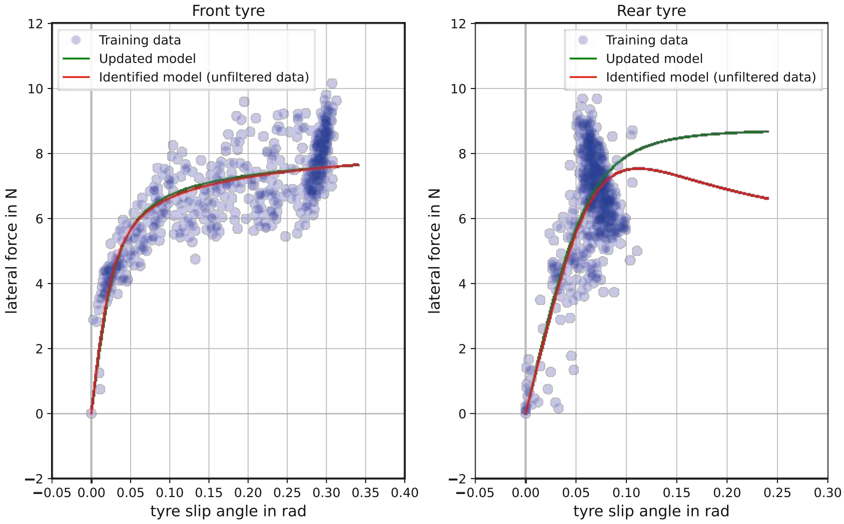

Figure 26: Comparison of filtered training data and model prediction with complete Pacejka model.

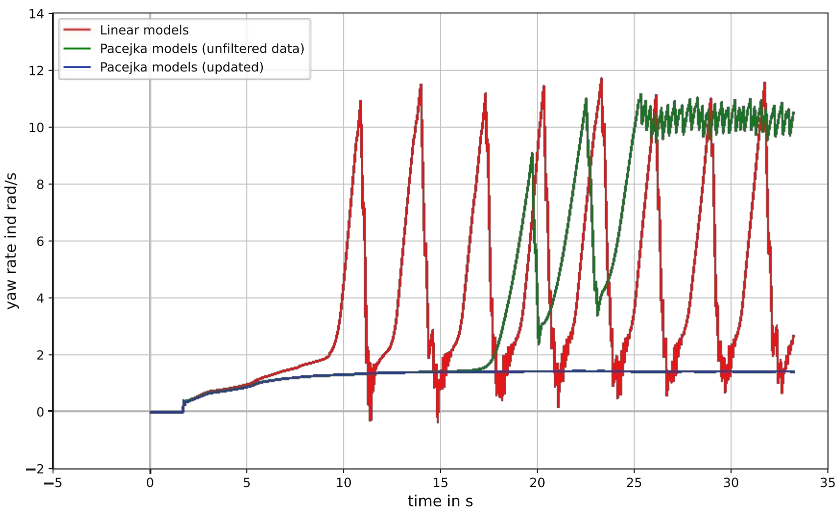

Figure 27: Comparison of the yaw rate predictions.

this chapter. Oscillations arise due to differences in the lateral forces produced by the front and the rear tyres. This phenomenon is more pronounced in linear models due to their limited application range (tyre slip angle values). Figure 28 describes the difference in generated forces for the linear model, the Pacejka model with the wrong curve shape by the rear tyre due to the influence of unfiltered outliers and the Pacejka model with the correct curve shape for the rear tyre. Due to oscillations in Figure 28, it is hard to identify the source of such behaviour. To avoid the oscillations, simulation results achieved with the filtered data were

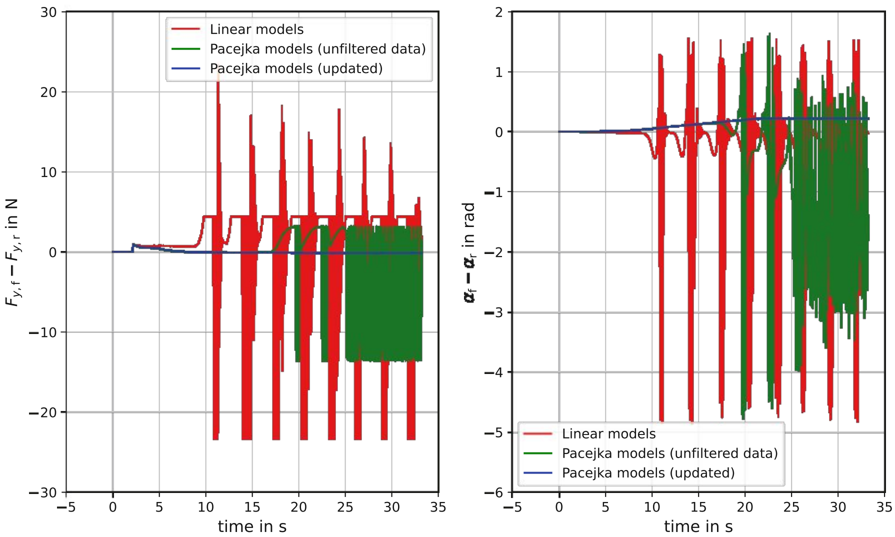

Figure 28: Comparison of lateral force and slip angle differences between the front and rear tyres.

fed to the linear models and the Pacejka models with identified parameters based on the unfiltered data to achieve the function values shown in Figure 29 and 30. The first figure shows the evolution of the lateral forces for front and rear tyres compared to each other for all three models over the slip angle values. At the same time, Figure 30 shows how the difference in lateral forces evolves over slip angle values. The difference in linear model function slopes of front and rear tyres affects continuously increasing differences because the rear tyres reach lower slip angle values. One may also observe a slightly more significant area between the curves of the front and rear tyres in the linear region for Pacejka models identified with unfiltered data in comparison to Pacejka models identified with filtered data - Figure 29. This greater area can also be observed in Figure 30 for slip angle values between 0.0 and 0.05 radians, as the green line has the same trend as the blue, but higher function values. In addition to this, there is a point where both tyre forces are equal (ca. 0.10 radians) for both Pacejka models. However, after reaching this point, the front tyre lateral force starts to dominate over the rear one (in terms of having larger values) by the model based on unfiltered data - the force difference is positive. The opposite happens for the Pacejka model based on filtered data - the force difference is negative. Considering static parameters of the vehicle shown in Table 1 and Equation 31, the positive force difference will lead to even greater values of yaw rate derivative during the integration and, as a result, a jump in yaw rate value. The jump in yaw rate triggers oscillations in tyre slip angle values, which simultaneously triggers lateral force oscillations, as the tyre force depends on slip angle as the only argument.

The mentioned preprocessing (filtering) steps are also described in Subsection 4.6.2 to systematize the process and are implemented for the identification toolbox described in A.2.

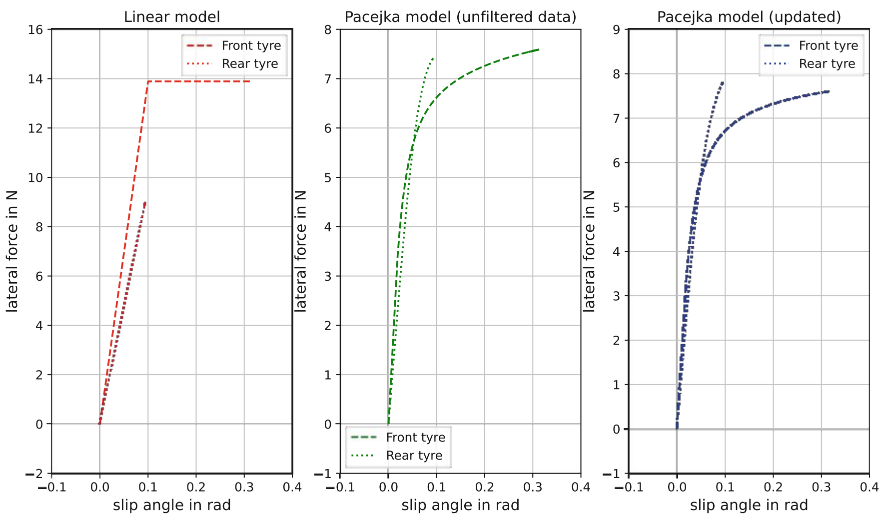

Figure 29: Generated lateral forces of front and rear tyres based on slip angle achieved during simulation with filtered data.

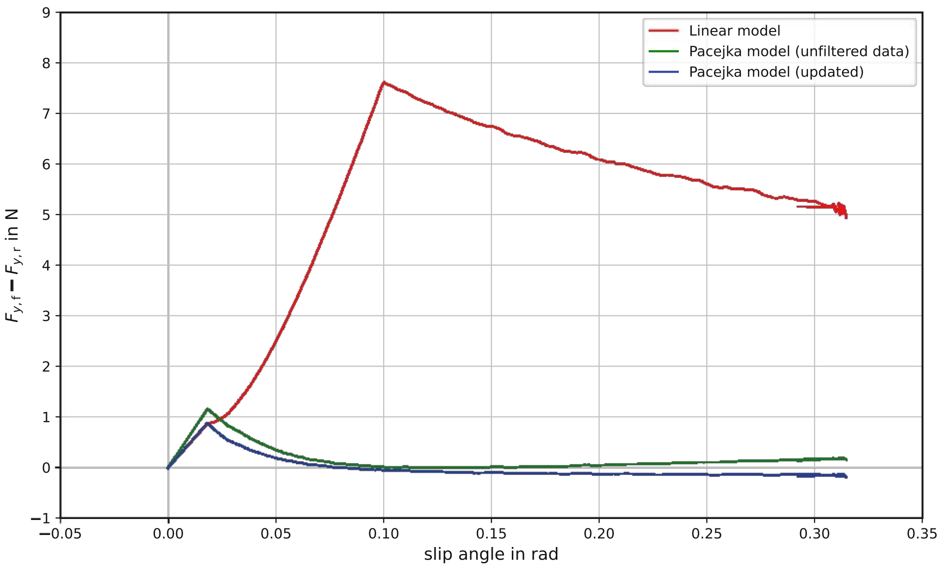

Figure 30: Comparison of lateral force difference between the front and rear tyres.

5.3. Complete Model

In the preceding sections, the longitudinal and lateral dynamics models underwent separate (decoupled) validations, each drawing conclusions regarding efficient identification test manoeuvres and model accuracy. However, the true capability of the complete vehicle model lies in its ability to couple longitudinal and lateral dynamics correctly. Ground truth data from the IMU measurements, encompassing acceleration, yaw angle and yaw rate, serve as benchmarks for this validation. This expands the scope of the validation to four variables: longitudinal and lateral accelerations, yaw angle and yaw rate.

The RPAD's primary function is to follow a predefined driving profile, requiring an accurate measurement of the vehicle's displacement. While SLAM algorithms are typically employed, specific environmental conditions during the conducted measurement campaigns required an alternative solution. Operational challenges, including overload and compromised sampling time, forced to find a workaround by constructing a controlled environment. LiDAR data recorded in this environment is used to estimate the vehicle's position offline using an extended Kalman filter and a particle filter. These filters leverage the dynamics model as a basis for predictions, yielding position estimates assumed as pseudo-ground truth for the vehicle's location. This pseudo-ground truth serves as a basis for validating the vehicle's position prediction. The subsequent sections provide insights into the implementation of both filters and are followed by the validation process of the complete model.

5.3.1. LiDAR Model

Due to the specific conditions of the measurement environment and the need for computational time reduction, an alternative to using the SLAM algorithm was found. Namely, a simplified model was designed for the position and orientation estimation offline. While developed for a specific test environment, this model can be adapted for similar use cases due to its straightforward setup. This approach allows for the position and orientation estimation without overloading computational resources or causing data loss due to delays.

The basic concept of this model is illustrated in Figure 31. Initially, four obstacles (cylinders with a diameter of approximately 95 mm and a height of 40 mm) are positioned in predefined locations to form a rectangle of predefined shape, acting as beacons. The vehicle starts at the known coordinates and with the known orientation. As the vehicle moves, LiDAR measurements are taken, and the potential location of the vehicle is estimated using the vehicle dynamics model at each sample time. The estimated position and orientation are then used to calculate the potential coordinates of each predefined obstacle using the beam ranges and their corresponding angles at each sample time, as defined by Equations (64) and (65).

$$O_{i,x} = x_p + r_j \cos(\psi_p + \gamma_j) \tag{64}$$

$$O_{i,y} = y_p + r_j \sin(\psi_p + \gamma_j) \tag{65}$$

Here, O_i describes the i-th of four obstacles, x_p, y_p and ψ_p are the vehicle dynamics model position and orientation predictions, r_j is the j-th beam range (of a total of 1081 beams) and γ_j its corresponding beam angle.

The calculated coordinates of potential obstacle locations are then compared with the actual coordinates of the four predefined obstacles (or beacons). If the difference is smaller than the reference value of 1.0 meters, the current beam range and its angle are used to calculate the position of the vehicle using

$$x_L = B_{i,x} - r_{avg} \cos(\psi_p + \gamma_j) \tag{66}$$

$$y_L = B_{i,y} - r_{avg} \sin(\psi_p + \gamma_j) \tag{67}$$

where B denotes the true obstacle coordinate.

The calculated position of the vehicle and the difference between the potential and actual obstacle locations η_k are saved. The mentioned difference will be referred to as an accuracy score. If the accuracy score exceeds 1.0 meters, the current

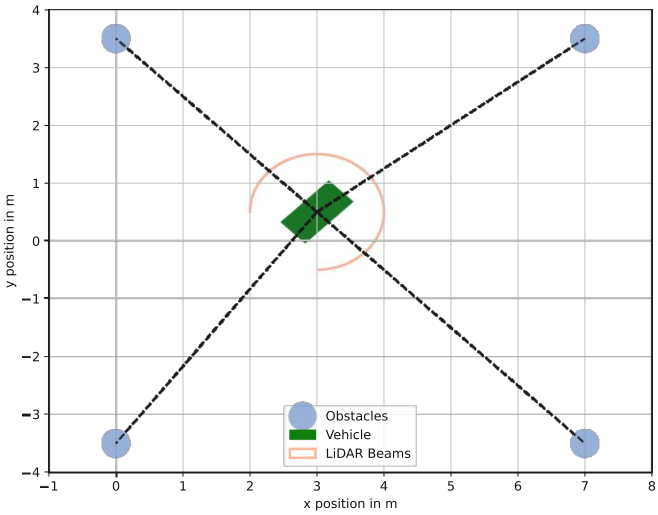

Figure 31: LiDAR model.

beam is skipped. Since there are 1081 beams per time step, multiple beams (typically two to eight) can detect the same obstacle. The accuracy score for each beam (which detected the obstacle) is used to calculate the weighted average of position coordinates at the current time step using:

$$\kappa_k = \frac{1}{\eta_k} \tag{68}$$

$$x_{avg} = \frac{\sum \kappa_k \, x_k}{\sum \kappa_k} \tag{69}$$

$$y_{avg} = \frac{\sum \kappa_k \, y_k}{\sum \kappa_k} \tag{70}$$

where k is an index of the beam which detected the obstacle.

This algorithm is repeated at every time step. The current position estimation is skipped if none of the four predefined obstacles is recognized during the current sample. For example, for the measurement of approximately 370 samples, only two did not produce a valid position estimation.

5.3.2. Kalman Filter

The Kalman filter is an optimal filter for dynamic system state estimation under the influence of white-noise random excitation and measurement noise. The filter is mighty in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modelled system is unknown [23]. Integrating the Kalman Filter with the dynamics model of the vehicle offers a solution to smoothen noise in the output data of the LiDAR model shown in Section 5.3.1.

The Kalman filter addresses the general problem of estimating the state of a discrete-time controlled process governed by the linear stochastic difference equation (71) with measurement vector described with (72) [23], where the variables w_k and v_k are process and measurement white noise (zero-mean, uncorrelated, known covariances Q and R, respectively).

$$x_{k+1} = F\,x_k + B\,u_k + w_k \tag{71}$$

$$z_k = H_k\,x_k + v_k \tag{72}$$

Kalman filter algorithm consists of two steps: prediction and update [24]. During prediction, the state estimate and the estimated error covariance matrix are calculated using (73) and (74):

$$\hat{x}^-_{k+1} = F\,\hat{x}^+_k + B\,u_k \tag{73}$$

$$P^-_{k+1} = F\,P^+_k F^T + Q \tag{74}$$

The hat operator signifies that the variable is estimated, while superscripts $-$ and $+$ describe predicted and updated estimates.

The update step begins with the calculation of the measurement residual representing the difference between the true measurement and the estimated measurement:

$$\tilde{y} = z_k - H\,\hat{x}^-_{k+1} \tag{75}$$

Subsequently, the Kalman gain K_{k+1} used to determine the optimal correction of the predicted state estimate is calculated with

$$K_{k+1} = P^-_{k+1}\,H^T(R + H\,P^-_{k+1}H^T)^{-1}. \tag{76}$$

Finally, the state estimate and error covariance matrix can be updated:

$$\hat{x}^+_{k+1} = \hat{x}^-_{k+1} + K_{k+1}\tilde{y} \tag{77}$$

$$P^+_{k+1} = (I - K_{k+1}\,H_{k+1})\,P^-_{k+1} \tag{78}$$

In the case of the vehicle model described in this work, it is impossible to express the process model with linear equations. The problem can be described with (79) and (80).

$$x_{k+1} = f(x_k, u_k) + w_k \tag{79}$$

$$z_{k+1} = h(x_{k+1}) + v_{k+1} \tag{80}$$

To apply the Kalman filter, it is necessary to linearize the functions f and h around respective state estimates:

$$F_k = \frac{\partial f}{\partial x}\Big|_{\hat{x}_k^+, u_k} \tag{81}$$

$$H_{k+1} = \frac{\partial h}{\partial x}\Big|_{\hat{x}_{k+1}^-} \tag{82}$$

This specific approach for nonlinear system estimation is known as the extended Kalman filter.

5.3.3. Particle Filter

The particle filter is an alternative method for determining the vehicle's position based on LiDAR measurements. The particle filter method is a sequential Monte Carlo technique for solving the state estimation problem, using the Sequential Importance Sampling algorithm and including a resampling step at each instant [25]. Particle filter represents the possible states of the system at the current time step with a set of random samples, called particles. Each particle represents a hypothesis about the state of the system. The particles which match the actual measurements better receive higher weights. These particles are more likely to be selected to form the set for the next time step.

Considering specific conditions outlined in Subsection 5.3.1, the algorithm can be described with the following steps:

1. Draw of N particles with a uniform distribution, each with predefined position and orientation noise. Each particle comprises a particular tuple of estimated quantities ((X, Y)-coordinates and yaw angle ψ). Each particle is assigned the same initial weight $\frac{1}{N}$.

2. Prediction step based on the system dynamics model for given input variables at the current sample time for every particle, obtaining a new estimate of the position and orientation for each particle.

3. Based on position and orientation found in the previous step, a simulation of LiDAR readings to detect four predefined obstacles (See Subsection 5.3.1) is made for each particle.

4. The measured distances to four predefined obstacles are found with the LiDAR model described in Subsection 5.3.1.

5. Simulated measurements and actual measurements of LiDAR data are compared (distances to obstacles), and the corresponding weights for each particle are calculated based on this comparison. The weights represent the likelihood of measuring the considered data point under the assumption that the current particle estimates the state correctly.

6. Resampling of particles based on the calculated weights, while adding position and orientation noise to a new set of particles. The current position of the vehicle is then calculated as the weighted average of the resampled particle set.

7. Repeat the process, starting with the system dynamics model update (step 2).

The number of particles is chosen to be $N = 1000$, while position and orientation noise are modelled as a normal distribution with zero mean and standard deviation of 1 meter and 0.1 radians, respectively.

5.3.4. Validation Process

A linear model was selected for longitudinal dynamics and a complete Pacejka model for the front and rear tyres to validate the RPAD model. Parameters for all models were determined during lateral identification with a quasi-steady-state manoeuvre, specifically, Type 3 parameters for the longitudinal model and Type B parameters for the tyre models.

The Look-Up Table (LUT) controller, as described in [26], was designed to guide the RPAD to follow the predefined path in terms of X and Y coordinates for a given longitudinal velocity. Due to safety considerations in the measurements environment, the velocity was constrained to approximately 3.0 m/s by controlling motor speed commands. Under these conditions, the estimated tyre slip angle of the front tyre reaches absolute values of 0.33 radians, indicating high slip values.

The states and their derivatives were compared to IMU measurements to assess the accuracy of the model estimations. Table 26 presents the RMSE and R-squared scores for longitudinal and lateral accelerations, yaw angle and yaw rate. Notably, the R-squared values for yaw rate and lateral acceleration predictions surpass those observed during the validation of tyre models. This could be attributed to the vehicle spending less time in high-slip states and to the validation curve simulating the potential environment in which the RPAD would operate. Furthermore, yaw angle scores demonstrate exceptional accuracy, with a low average magnitude of prediction errors and R-squared values close to 1.

	Yaw Angle	Yaw Rate	Long. Accel.	Lat. Accel.
RMSE	0.06215	0.16641	0.36242	0.55810
R-squared	0.99799	0.97777	0.77373	0.94921

Table 26: Accuracy metrics for validation data - states measured with external IMU.

However, the prediction accuracy of the longitudinal model reveals contrasting results. The R-squared value is significantly lower compared to tyre model related variables and to validation results for longitudinal dynamics only. This discrepancy may be attributed to the coupling of lateral dynamics and greater noise during the constant velocity stage, as illustrated in Figure 32. Yet, the extended constant velocity stage results in lower RMSE values than any achieved during the validation process before.

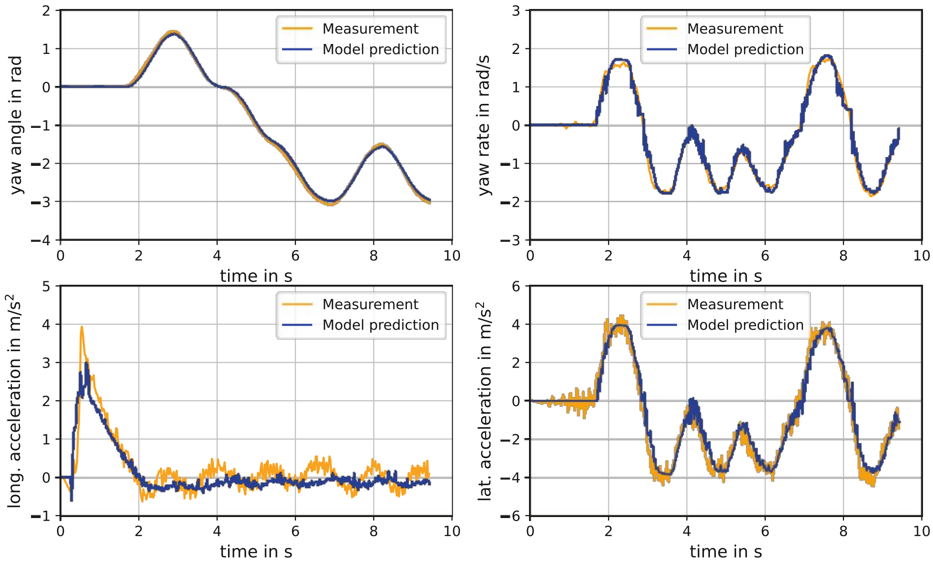

Figure 32: Comparison of IMU measurements and corresponding model predictions.

Upon further analysis of Figure 32, larger deviations in the yaw angle are noticeable at seconds 3 and 8. To determine the source of such differences and their potential impact on the model results, a comparison was made between the prediction of position and the corresponding data estimated with the LiDAR model using the described extended Kalman filter or particle filter. Figures 33 and 34 reveal that the prediction of vehicle displacement aligns closely with filter estimates. However, deviations in yaw angle occur during turns or transitions between turns.

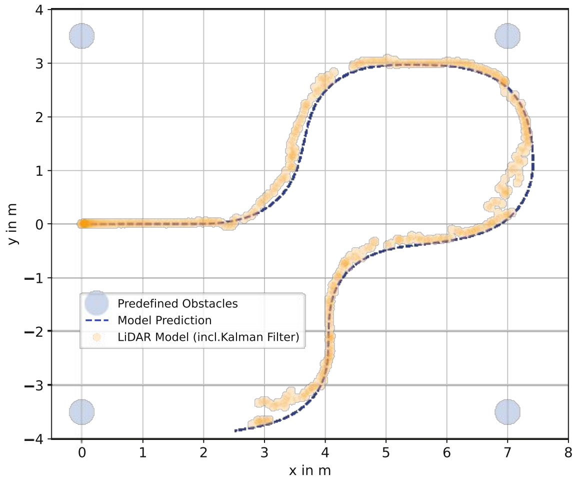

Figure 33: Comparison of LiDAR model estimation and model prediction.

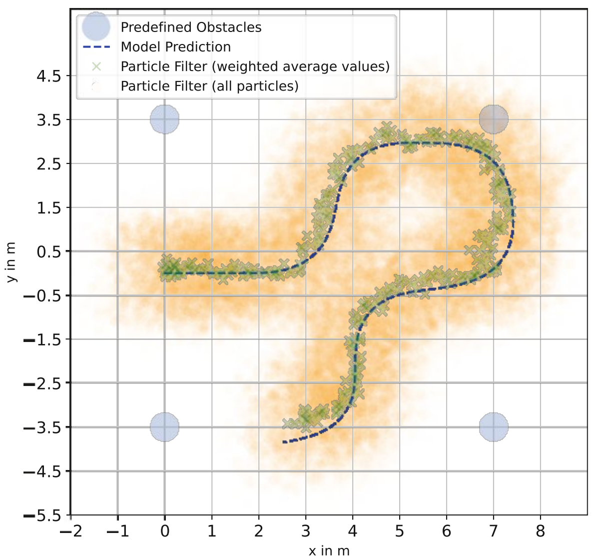

Figure 34: Comparison of particle filter estimation and model prediction. Orange circles show the distribution of particles during the whole measurement.

Given that the yaw angle derivative is solely described by the yaw rate in the system dynamics model (28), it was hypothesized that insufficient cornering in turns or transitions between the turns is associated with the cosine term in the equation of yaw rate (31). Since the same term is also employed in the differential equation of lateral velocity (30), the cosine term was eliminated from both equations for consistency reasons, and the simulation was repeated. Table 27 summarizes prediction accuracy scores for validating IMU signals. The RMSE value for all variables, except for yaw angle, worsened, although insignificantly and can be disregarded.

	Yaw Angle	**Yaw Rate**	**Long. Accel.**	**Lat. Accel.**
RMSE	0.05135	0.17526	0.364	0.56388
$\mathbf{R^2}$	0.99863	0.97535	0.77176	0.94815

Table 27: Accuracy metrics for the prediction of (X, Y) coordinates - states measured with the IMU.

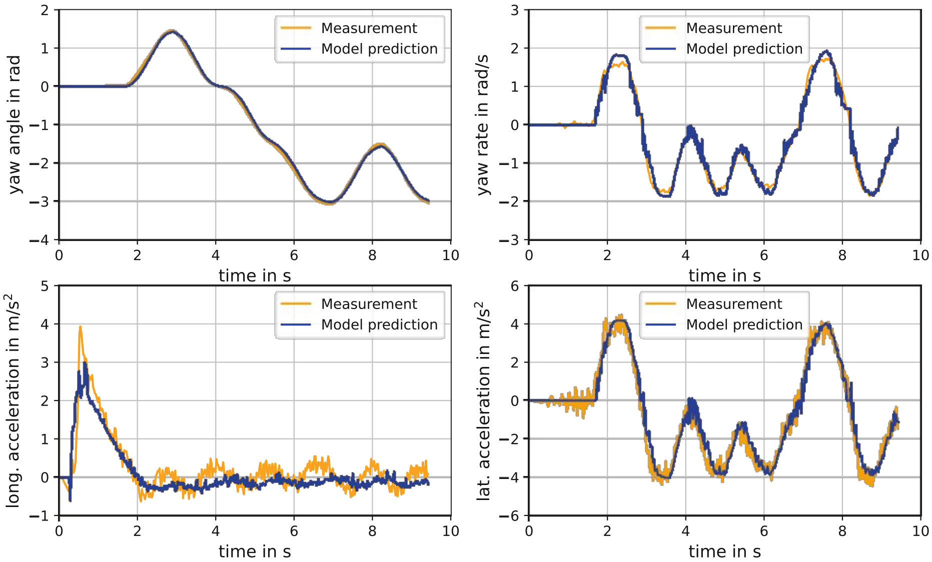

Figure 35: Comparison of IMU measurements and corresponding model predictions.

The LiDAR model and particle filter estimations were compared to model predictions before and after the yaw rate equation update to assess the impact on (X, Y) estimations (Figure 36). The cornering became slightly sharper with the update. Table 28 validates this observation by comparing metrics of the original and updated models. The RMSE value improved by around 7% for the (X, Y) estimations (1 centimetre on average).

	RMSE	$\mathbf{R^2}$
Considering cosine term	0.12143	0.99810
Neglecting cosine term	0.11295	0.99836

Table 28: Accuracy metrics for the prediction of (X, Y) coordinates - compared to LiDAR Model estimation with integrated Kalman Filter.

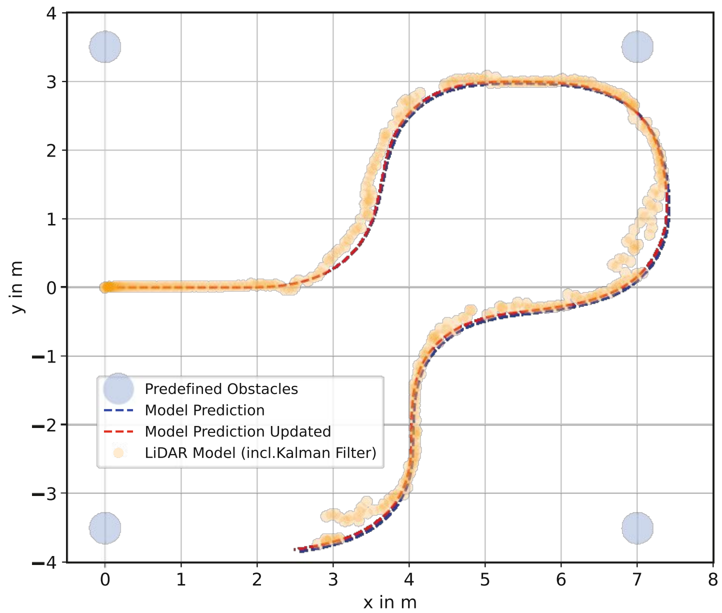

Figure 36: Comparison of LiDAR model estimation and model prediction.

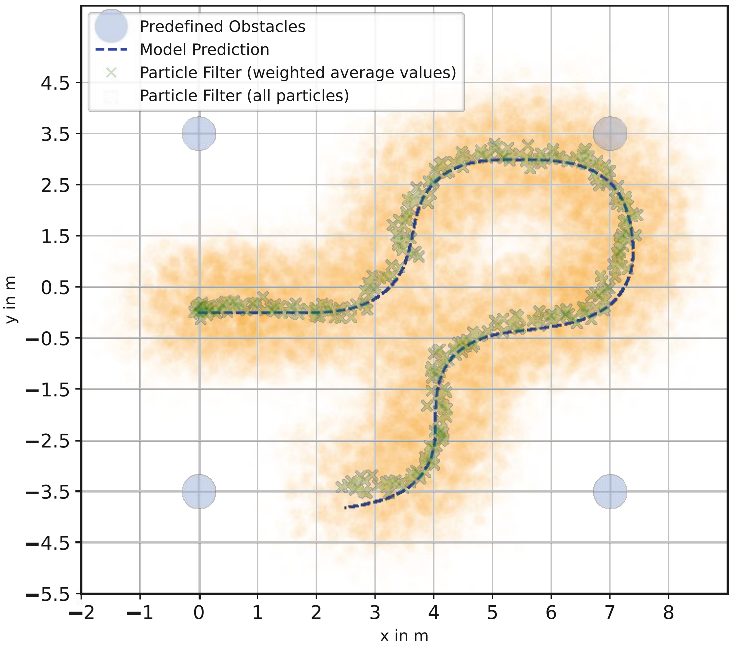

Figure 37: Comparison of particle filter estimation and prediction of updated model. Orange circles show the distribution of particles during the whole measurement.

In summary, the model exhibits accurate results even under the high slip conditions for the front tyre. Despite the simplicity of the linear model, it proves sufficient to capture the longitudinal dynamics of the vehicle. Moreover, the validation showed that performance during turns can be enhanced by eliminating the cosine term from the yaw rate and lateral velocity equations. In this case, the predicted location of the vehicle deviates by around 11 centimetres on average.

6. Conclusion

This thesis provides a comprehensive overview of the identification process for the vehicle dynamics based on the RPAD platform. The detailed description of the RPAD's hardware setup, including sensors, actuators, and computational units, established the constraints and the range of measurable system states. The dynamic bicycle model was chosen for the description of the vehicle. To address non-linearity in lateral dynamics, the Pacejka model, in its complete and simplified versions, was selected to explain tyre behaviour. The DC-motor model and the resistance forces, along with their extended and simplified versions, were employed to describe the longitudinal motion of the RPAD.

The identification experiments were designed to meet constraints, simplify model equations, and accurately capture vehicle dynamics. The subsequent identification process revealed both the advantages and disadvantages of experiments and models. The validation process compared model predictions with the IMU measurements and introduced LiDAR models with an extended Kalman filter and a particle filter for position estimation validation in the absence of an objective ground truth, leveraging the available LiDAR sensor. The validation stage demonstrated that the same identification manoeuvre (quasi-steady-state) could be applied to combine the identification of longitudinal and lateral dynamics parameters without sacrificing model accuracy. For the conducted driving profile and conditions chosen for the identification and validation processes (speed up to $3\,\mathrm{m/s}$ with front tyre slip angle up to $0.33\,\mathrm{radians}$), the identified model exhibited highly accurate results for predicting lateral states (yaw angle, yaw rate, lateral acceleration) and adequately captured longitudinal dynamics described by the linear model despite the presence of significant noise in measurable variables. The step-by-step procedure was automated and encapsulated in a ROS 2 node for data measurement and an identification Python package for parameter estimation.

Future works could explore the validity of the identification process and resulting model for more challenging conditions. The influence of pitch and roll on vehicle dynamics could be analyzed for the load transfer consideration in the double-track model. Additionally, an analysis of the potential improvement of model accuracy through data-driven models using subspace system identification or neural networks could be considered.

A. Toolbox

To automate the parameter identification process described in Chapter 4, a ROS 2 node and an offline Python package were created. The following sections describe how both node and package can be used to create motor and servo input commands, start the measurement, record topics in *.csv-file and identify longitudinal and lateral tyre parameters. For a quick review of ROS 2 basics, one may address the corresponding subsection in Chapter 2.

A.1. ROS 2 Node

As was mentioned in Chapter 2, the RPAD may be controlled using duty cycle or motor speed for motion in the longitudinal direction and via servo position for motion in the lateral direction. To execute the quasi-steady-state manoeuvre described in Chapter 4 (constant RPAD speed with specific steering velocity), it is better to prepare the input commands in terms of duty cycle or motor speed and servo position first. Then, they are being fed into the predefined node while subscribing to topics of interest and saving them at the end of manoeuvre execution. Exactly this role is played by the ROS 2 Node, called `tester_node`. However, this node can also be used only for topic recording.
`tester_node` consists of the following folders and files:

- `tester_node` — Package folder
  - `input_commands` — Folder for input commands
    - `model_inputs.csv` — Input commands file
  - `recordings` — Folder for saved recordings files
    - `rec_file_11_29_T18_35_35.csv` — Example of recordings file
  - `resource`
    - `tester_node` — Marker file for the package
  - `test` — Folder for unit testing
  - `tester_node` — Used by ROS 2 to find the package
    - `__init__.py`
    - `node_function.py` — Executable of the package
  - `package.xml` — Meta information about the package
  - `setup.cfg` — Required in case the package has executable(s)
  - `setup.py` — Information about installation instructions

The input commands file is a *.csv-file, which describes the motor speed and servo position for each sample time. The column names may be arbitrary, but keeping the correct order of time, motor speed, and servo position is important. This file can be created by the user or generated using the corresponding function of the offline package (See Section A.2.1). The file must be saved with ";" as a delimiter. Motor speed is specified in Revolutions Per Minute (RPM) and servo

input in radians. Servo input should consider the offset specified in the `vesc.yaml` file. For example, to set the steering angle to x degree, one must add or subtract (depending on VESC configuration) the servo offset (ca. 0.515 radians).
The node is used only as a recorder if no values other than column names are given in the table. In this case, it makes it possible for the user to execute custom manoeuvres using a joystick while recording topic messages in *.csv-file.
Before running the node, a user has to make sure that the node is located in `src` source folder of ROS 2 workspace and installed using:

```
colcon build --packages-select tester_node
```

In addition, it is highly recommended that the wheels do not touch the ground first before making a test run and checking if the steering direction satisfies the expectations. When precautions are made, one may start the node using:

```
ros2 run tester_node tester_node
```

As the last command of the inputs file is executed, the node saves the recording in `recordings` folder inside of the node with the following name schema: `rec_file_MM_DD_THH_mm_ss.csv`. For example, `rec_file_11_29_T18_35_35.csv`.

A.2. Offline Identification Package

To automate the estimation of RPAD model parameters based on recordings made with `tester_node`, `identification_toolbox` Python package was created. It preprocesses the recording data, identifies wrong data and outliers, calculates parameters of multiple longitudinal and lateral models, saves them in a *.json file and visualizes tyre models. In addition to this, it has its offline simulator, which can be used separately in other scripts to simulate RPAD behaviour.
`identification_toolbox` consists of the following folders and files:

identification_toolbox .... Package folder
- identification_toolbox .... Folder of Sub-packages
  - commands .... Sub-package for generation of input commands file
  - config ...... Basis configuration files for estimation and their Python handlers
  - controller .... Sub-package for PID and VESC controllers
  - estimation . Sub-package for parameters estimation and optimization
  - lateral .... Sub-package for tyre parameters identification routine
  - longitudinal . Sub-package for longitudinal parameters identification routine
  - postprocessing .... Sub-package for visualization
  - preprocessing .... Sub-package for recordings pre-processing
  - simulator .... Sub-package for simulator
  - validation_filters .... Sub-package for Kalman Filter and Particle Filter for special cases
  - vehicle .... Sub-package for vehicle and tyre model equations
  - __init__.py .... Package identifier file
  - logger_setup.csv .... Handler for logging
- tests .... Information about installation instructions
  - model_inputs.csv .... Input commands file
- README.txt .... General information and instructions
- run_identification.py ..... Runner for identification and optimization routines
- run_tests.py .... File for unit testing
- setup.py .... Information about installation instructions

User can install this package as any other Python package by typing in the command line or bash inside of the package folder:

```
pip install -e .
```

In this way, the user can import `identification_toolbox` Sub-packages in their code. However, this step is optional and does not impact the usage of common functionality, such as parameter identification.

A.2.1. Input Commands File Generation

As was mentioned in the previous section A.1, node `tester_node` needs *.csv-file with defined motor speed and servo position commands. This file can be created by the user or generated using one of the sub-packages of `identification_toolbox`, `commands`. The sub-package has only this specific purpose.

identification_toolbox .... Package folder
└─ identification_toolbox .... Folder of sub-packages
  └─ commands .... Sub-package for generation of input commands file
    ├─ __init__.py .... Sub-package identifier
    └─ create_test_commands.py .... Function for generation of input commands file

`commands` sub-package is capable of generating files for three kinds of manoeuvres:

- spiral maneuver for quasi-steady-state measurement (Sub-section 4.6.1) - constant speed and non-zero constant steering velocity
- straight line for longitudinal identification - constant speed
- acceleration-braking for longitudinal identification - acceleration for 3 seconds and braking to desired motor speed

The general execution call will look the following way:

```
python <path\to\toolbox>\identification_toolbox\commands\create_test_commands.py
```

However, multiple options are available for the user to generate input command files for each manoeuvre. All of them are optional and have predefined default values. Generally, the main focus is on spiral manoeuvre for tyre parameter identification, leading to the smallest set of options needed to generate a file for convenience reasons. This is done by choosing appropriately the default value for specific options. The following list will give a complete overview of possible function calls:

- sample time, `--sample <value>` (in seconds) - needed to set sample time of `tester_node` and determine correct servo position for spiral manoeuvre. By default, 0.025 seconds or 40 Hertz.
- motor speed, `--rpm <value>` (in RPM) - target motor speed for acceleration or constant velocity region. By default, 11000.0 RPM.
- motor speed for braking, `--brakerpm <value>` (in RPM) - end motor speed for acceleration - braking manoeuvre. By default, None.
- starting steering angle, `--offset <value>` (in degree) - in case of small space for spiral manoeuvre, it may be useful to start with already slightly steered front wheels. By default, 0 degrees.

- steering velocity step, `--step <value>` (in degree/second) - the rate of the change steering angle of front wheels during the spiral manoeuvre. Set to 0 to drive a straight line. By default, 1 degree/second.
- steering direction, `--spin <value>` - direction of front wheels steering for spiral manoeuvre, which defines whether the vehicle will turn clockwise or counter-clockwise. The value depends on the VESC specifications of the vehicle. 1 is for counter-clockwise and -1 for clockwise. By default, 1.

Here are some examples of common function calls made for the identifications during this research:

```
python .\identification_toolbox\commands\create_test_commands
.py --offset 4 --spin -1
python .\identification_toolbox\commands\create_test_commands
.py --step 0 --rpm 13000.0 --brakerpm 0.0
```

As a result of one of those calls, the input commands file `model_inputs.csv` is generated in the same folder where the call was made. This file can be moved then to `input_commands` folder of `tester_node` for further test manoeuvre executions.

A.2.2. Parameters Identification

A recordings file made with `tester_node` A.1 is required for further process. If the user has alternative ways to make measurements and save them in separate *.csv-file, the column names should at least match those generated with `tester_node`. The recording file should be in the same folder from which the function call will be made.
Another key component is `VESC.yaml` file of the vehicle. This file should be copied to `path-to-toolbox/identification_toolbox/config`. This step ensures that the current vehicle configuration will be considered when setting motor and servo limitations and specific variable ratios.
The main file for automated parameters identification is `run_identification.py`, located in the root folder of `identification_toolbox`. As done by generating input commands file, the user has multiple options through the function call. Basic function call

```
python <path-to-toolbox>\run_identification.py
```

results in the identification of tyre parameters for linear, reduced (3 parameters) Pacejka model and complete (4 parameters) Pacejka model and of linear (3 parameters), physical (4 parameters) and polynomial (6 parameters) longitudinal models. However, the user can also start identifying only for tyre or longitudinal parameters and optimize already identified ones. In the latter case, one should ensure that parameters are saved in *.json-file with the same format as in the already available file ./identification_toolbox/config/estimated_parameters.json. Full description of possible options:

- type of identification -t <type name> - available choices are *lateral*, *longitudinal*, *optimize* and *all*. By default, *all*.
- path to recording file -p <path-to-file> - if not specified, *.csv-file from current folder will be assumed as one.
- visualize results -v <boolean> - *True* to save figures of comparison between model and measurements. By default, *False*.
- name of the file to save -f <file-name> - by default, estimated_parameters.json.
- configuration of simulator -s <path-to-config> - path to configuration file of simulator. An example can be found in path-to-toolbox/identification_toolbox/config. simulator_config.json file in this folder is a default option.
- file with estimated parameters -b <path-to-file> - file is used as basis for parameter optimization. By default, estimated_parameters.json in ./config is used as basis.

Here are some examples of calls made for identification during this research:

```
python <path-to-toolbox>\run_identification.py -t lateral -v
True
python <path-to-toolbox>\run_identification.py -t optimize -b
g/h/i.json
```

If -f option were not specified, the parameters would be saved in estimated_parameters.json file in the folder the call was made from, as well as a logging file of identification process and figures, if -v True option was entered.

A.2.3. Simulator

To use the simulator, it is recommended to install the package first A.2. Then, it can be imported as a Python class:

```
from identification_toolbox.simulator import Simulator
```

`Simulator` object can be initialized without any parameters. However, there are four different optional arguments which can personalize the simulation. Those arguments are:

- `dt` - sample time of the simulation. By default, 0.025s.
- `x0` - initial states values. By default, a dimensional zero array of 7 elements (Description of 7 states is later in this subsection).
- `param_file` - path to *.json-file of identified parameters. By default, a file from `config` sub-package of the toolbox will be used.
- `config_file` - path to *.json-file of simulator configuration. By default, a file from `config` sub-package of the toolbox will be used.

In general, there are two possibilities for running the simulator - open loop and closed loop simulation. For the open loop simulation, the inputs are the Nx2 array of duty cycle and steering angle values (not servo position) and, optionally, the Nx1 time array, in case sample time varies during the measurement. Here is an example of how it can be called:

```
sim_obj = Simulator(dt=sample_time, x0=initial_state,
            param_file=some_path1, config_file=some_path2)
x, xdot = sim_obj.open_loop_simulation(nx2_inputs,
            t_array=nx1_time)
```

The closed loop simulation is designed to test the speed controller or simulate the VESC controller. The reference output is then the desired velocity vector. To call closed-loop simulation, one has to specify the Nx1 array of steering angle values and the Nx1 array of desired velocity array. Considering initialization of `Simulator` object `sim_obj` from previous example, function call can made with:

```
x, x_dot = sim_obj.closed_loop_simulation(nx1_steering,
                                          nx1_v_reference)
```

The results are the Nx7 state array and its Nx7 derivative array. Seven states are x,y-position, yaw angle, x,y-velocity, yaw rate and vehicle slip angle. The results can be visualized against IMU measurements using static method
`visualize_simulation` of `Simulator` class. Two input arguments of this function are:

- tuple of states array and its derivative array estimated during the simulation
- tuple of time, longitudinal and lateral accelerations and yaw rate arrays of IMU (consider the order of variables).

List of Abbreviations

RPAD Research Platform for Autonomous Driving

ROS Robot Operation System

DC Direct Current

IMU Inertial Measurement Unit

LiPo Lithium Polymer

SDK Software Development Kit

RPM Revolutions Per Minute

RMSE Root Mean Square Error

DoF Degrees of Freedom

MPC Model Predictive Control

CoG Center of Gravity

LUT Look-Up Table

VESC Vedder Electronic Speed Controller

References

[1] F. Foundation, “F1TENTH.” https://f1tenth.org/, 2023. Accessed: December 17, 2023.

[2] A. L. Gratzer, S. Thormann, A. Schirrer, and S. Jakubek, “String stable and collision-safe model predictive platoon control,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 19358–19373, 10 2022.

[3] A. L. Gratzer, M. M. Broger, A. Schirrer, and S. Jakubek, “Flatness-based mixed-integer obstacle avoidance mpc for collision-safe automated urban driving,” in *9th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1844–1849, IEEE, 7 2023.

[4] A. L. Gratzer, M. M. Broger, A. Schirrer, and S. Jakubek, “Two-Layer MPC Architecture for Efficient Mixed- Integer-Informed Obstacle Avoidance in Real-Time,” *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[5] A. L. Gratzer, A. Schmiedhofer, A. Schirrer, and S. Jakubek, “Agile Mixed-Integer-based Lane-Change MPC for Collision-Free and Efficient Automated Driving,” *IEEE Transactions on Intelligent Vehicles (under review)*, 2024.

[6] F. Foundation, “Building the F1TENTH Car.” https://f1tenth.readthedocs.io/en/foxy_test/getting_started/build_car/index.html#building-the-f1tenth-car, 2023. Accessed: December 17, 2023.

[7] TRAXXAS, “Slash 4X4 Platinum Edition.” https://traxxas.com/products/models/electric/6804Rslash4x4platinum, 2023. Accessed: December 17, 2023.

[8] NVIDIA, “NVIDIA Jetson Xavier.” https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-series/, 2023. Accessed: December 17, 2023.

[9] T. B. LTD, “VESC 6 MkVI PCB.” https://trampaboards.com/vesc-6-mkv--the-next-generation--benjamin-vedder-electronic-speed-controller-p-27517.html, 2023. Accessed: December 17, 2023.

[10] LP-RESEARCH, “LPMS-AL3 Series: 9-Axis Inertial Measurement Unit.” https://www.lp-research.com/imu/lpms-al3/, 2023. Accessed: December 17, 2023.

[11] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot operating system 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[12] M. Schorn, *Diagnose und Sensor-Fehlertoleranz aktiver Fahrwerke*, pp. 27–46. Wiesbaden: Vieweg, 2006.

[13] M. Althoff, M. Koschi, and S. Manzinger, "Commonroad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, p. 719–726, IEEE Press, 2017.

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

[15] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, p. 628–647, July 2014.

[16] M. Reimpell, J.; Burckhardt, *Fahrwerktechnik: Radschlupfregelsysteme*. Würzburg: Vogel Fachbuchverlag, 1993.

[17] H. B. Pacejka, "Chapter 4 - semi-empirical tire models," in *Tire and Vehicle Dynamics (Third Edition)* (H. B. Pacejka, ed.), pp. 149–209, Oxford: Butterworth-Heinemann, third edition ed., 2012.

[18] H. B. Pacejka, "Chapter 1 - tire characteristics and vehicle handling and stability," in *Tire and Vehicle Dynamics (Third Edition)* (H. B. Pacejka, ed.), pp. 1–58, Oxford: Butterworth-Heinemann, third edition ed., 2012.

[19] J. Roberts and T. D. Roberts, "Use of the butterworth low-pass filter for oceanographic data," *Journal of Geophysical Research: Oceans*, vol. 83, no. C11, pp. 5510–5514, 1978.

[20] H. A. Soule and M. P. Miller, "The experimental determination of the moments of inertia of airplanes," NACA Technical Report 467, National Advisory Committee for Aeronautics, January 1934. Accession Number: 93R20831.

[21] M. A. Branch, T. F. Coleman, and Y. Li, "A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems," *SIAM Journal on Scientific Computing*, vol. 21, no. 1, pp. 1–23, 1999.

[22] R. Y. H. . J. C. G. Christoph Voser, "Analysis and control of high sideslip manoeuvres, vehicle system dynamics," *International Journal of Vehicle Mechanics and Mobility*, vol. 48, pp. 317–336, 2010.

[23] G. B. Greg Welch, *An Introduction to the Kalman Filter*. Department of Computer Science University of North Carolina at Chapel Hill Chapel Hill, NC 27599-3175, 1997.

[24] F. Govaers and F. Govaers, *Introduction and implementations of the kalman filter*. London, England: IntechOpen, 2019.

[25] F. G. Toro, D. E. D. Fuentes, D. Lu, U. Becker, H. Manz, and B. Cai, "Particle filter technique for position estimation in gnss-based localisation systems," in *2015 International Association of Institutes of Navigation World Congress (IAIN)*, pp. 1–8, 2015.

[26] J. Becker, N. Imholz, L. Schwarzenbach, E. Ghignone, N. Baumann, and M. Magno, "Model- and acceleration-based pursuit controller for high-performance autonomous racing," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5276–5283, 2023.