

Politecnico di Torino

Master's Degree in Mechatronic Engineering

Department of Electronics and Telecommunications (DET)



Master's Degree Thesis:

Vehicle Longitudinal Velocity Estimation Using Model-Based and Data-Driven Techniques

16.03.2026

Supervisors

Prof. Giuseppe Carlo Calafiore
Pantelis Kommatas

Candidate

Dorian Macak

*To my parents and my brother,
for their endless love and belief in me.
None of this would have been possible without you.
Volim vas.*

Acknowledgements

I would like to thank Professor Giuseppe Carlo Calafiore for supervising this thesis and for his guidance and feedback during its preparation.

I would also like to thank the team at Bugatti Rimac for the opportunity to work on this project and for their support throughout its development. I am particularly grateful to Pantelis Kommatas, who supervised the project on the company side and whose regular insights and advice were invaluable to this work.

On a more personal note, I would like to thank my friends for their support and encouragement throughout these months and years, both in my studies and beyond. I am grateful to have you as friends.

Finally, I would like to thank my family for their constant support throughout these years. You have lived this journey with me through every high and low along the way. Your love and support mean more to me than I can express.

Abstract

Accurate longitudinal velocity estimation is essential for vehicle control systems such as Electronic Stability Control, Anti-lock Braking Systems, and torque vectoring. In production vehicles, this estimation typically relies on wheel speed sensors and inertial measurement units, sometimes supplemented by Global Navigation Satellite System measurements. However, these sensors present inherent limitations: integration-based drift in inertial measurements, unreliable wheel speed readings under high-slip conditions, and degraded satellite signals in obstructed environments. Sensor fusion techniques are employed to combine information from multiple sensors and mitigate their individual limitations. While model-based algorithms, particularly Kalman filtering, are widely adopted, increasing data availability and computational capabilities have motivated the exploration of data-driven techniques as alternative or complementary approaches to sensor fusion and state estimation. This thesis investigates longitudinal velocity estimation through the development and evaluation of model-based, data-driven, and hybrid estimators. The work was conducted using data from a Driver-in-the-Loop simulator based on a high-fidelity model of a high-performance production vehicle. Five estimators are presented: two Kalman filter variants, one based on a kinematic vehicle model and the other incorporating wheel dynamics, two neural network architectures — a Multilayer Perceptron and a Long Short-Term Memory network — operating directly on sensor inputs, and a hybrid estimator combining model-based and data-driven strategies. The estimators were selected from a broader set of tested configurations and evaluated alongside a reference estimator provided by the industrial partner.

The results show that the developed estimators achieve strong overall estimation performance within the considered experimental framework and reveal distinct behavioural characteristics across the estimator classes. Data-driven models achieve strong average accuracy but exhibit larger worst-case errors, whereas model-based estimators provide more predictable behaviour due to the constraints imposed by the physical model. Additional analysis shows that operating conditions and tyre slip have a strong influence on estimator performance and provides further insight into the relative strengths and limitations of model-based, data-driven, and hybrid strategies for longitudinal velocity estimation.

Contents

Acknowledgements	i
Abstract	ii
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement, Objectives and Scope	2
1.3 Thesis Structure	3
2 Theoretical Background	4
2.1 Vehicle Motion and Reference Frames	4
2.2 Wheel-Speed-Based Velocity Estimation	5
2.2.1 The Wheel: Important coordinates and the free-body diagram	5
2.2.2 Pure Rolling and the Basic Principle	6
2.2.3 Longitudinal Tyre Slip	7
2.2.4 Wheel Dynamics and Slip Modelling	8
2.2.5 Modelling Assumptions and Practical Limitations	10
2.3 Vehicle-level modelling	10
2.3.1 Kinematic Model	10
2.3.2 Dynamic Modelling	13
2.4 Sensors	14
2.4.1 Wheel-speed sensors	15
2.4.2 Inertial Measurement Unit (IMU)	15
2.4.3 GNSS, Alternative Sensing Solutions and Sensor Fusion	16
2.5 State Estimation	17
2.5.1 Model-Based Estimation	17
2.5.2 Data-Driven Estimation	18
3 Literature Review	20
3.1 Overview	20
3.2 Kinematic Model-Based Estimators	21
3.3 Dynamic Model-Based Estimators	21
3.4 Learning-Based Estimation Approaches	22
3.5 Discussion and Positioning of the Present Work	23
4 Problem Setup and Development Framework	24
4.1 Problem Formulation	24
4.2 Simulation Environment	25

4.3	Dataset Allocation and Characterisation	26
4.3.1	Dataset Allocation	26
4.3.2	Data Characterisation	26
4.4	Development Principles and Constraints	30
4.5	Measurement Corruption	31
4.6	Comparative Evaluation Framework	33
5	Model-Based Estimators	34
5.1	Chapter Overview	34
5.2	Baseline Vehicle Model	34
5.2.1	Inertial Velocity Reconstruction	35
5.2.2	Wheel-Speed-Based Velocity Reconstruction	37
5.2.3	Two-Track Planar Vehicle Representation	38
5.3	KF01 - Kinematic Kalman Filter	39
5.3.1	Estimator Summary	39
5.3.2	Estimator Signals	40
5.3.3	State Vector and Process Model	40
5.3.4	Measurement Model	42
5.3.5	Filter Tuning	43
5.3.6	Model Algorithm and Expected Behaviour	44
5.4	KF02 - Dynamic Kalman Filter	46
5.4.1	Estimator Summary	46
5.4.2	Estimator Signals	46
5.4.3	State Vector and Process Model	47
5.4.4	Measurement Model	48
5.4.5	Filter Tuning	50
5.4.6	Model Algorithm and Expected Behaviour	52
6	Data-Driven Estimators	55
6.1	Chapter Overview	55
6.2	Learning Framework	56
6.2.1	Available Signals	56
6.2.2	Training and Validation Data Split	57
6.2.3	Loss Function	57
6.2.4	Hyperparameter Selection	58
6.3	MLP Estimator	58
6.3.1	Estimator Summary	58
6.3.2	Input Representation	59
6.3.3	Architecture and Feature Set Selection	60
6.3.4	Estimator Algorithm and Expected Behaviour	61
6.4	LSTM Estimator	62

6.4.1	Estimator Summary	62
6.4.2	Input Representation	62
6.4.3	Architecture and Feature Set Selection	63
6.4.4	Estimator Algorithm and Expected Behaviour	64
7	Hybrid Estimator	66
7.1	Chapter Overview and Design Approach	66
7.2	Hybrid Residual-Corrector Estimator (HYB)	67
7.2.1	Estimator Summary	67
7.2.2	Architecture and Signal Flow	68
7.2.3	Estimator Algorithm and Expected Behaviour	69
8	Results and Evaluation	71
8.1	Global Performance Comparison	71
8.2	Behaviour Across the Dynamic Envelope	73
8.3	Slip-Based Analysis	76
8.4	Qualitative Scenario Analysis	80
8.5	Summary of Experimental Findings	82
9	Conclusion and Future Work	84
9.1	Overview of the Study	84
9.2	Main Findings	84
9.3	Limitations of the Study	85
9.4	Future Work	86
	References	88

List of Tables

4.1	Dataset allocation for estimator development and evaluation.	26
4.2	IMU corruption parameters used for signal corruption	32
5.1	Summary of KF01 – Kinematic Kalman Filter	39
5.2	Final discrete-time covariance values for KF01	44
5.3	Summary of KF02 – Dynamic Kalman Filter	46
5.4	Final parameters for KF02	51
6.1	Allocation of datasets between training and validation sets.	57
6.2	Number of samples used for training and validation within the Development Dataset.	57
6.3	Summary of the MLP estimator	59
6.4	Validation-set performance of the considered MLP configurations.	60
6.5	Summary of the LSTM estimator	62
7.1	Summary of the hybrid estimator	67
8.1	Global performance comparison on the Test Dataset	71
8.2	RMSE of each estimator within the identified dynamic regimes.	75
8.3	High-error concentration ratio (HECR) across the dynamic regimes.	75
8.4	RMSE of the estimators under high-slip and regular conditions in the test dataset.	78
8.5	High-error concentration ratio (HECR) of the estimators under high-slip and regular conditions.	78
8.6	Relative RMSE increase and HECR for braking and high-slip conditions.	80

1 Introduction

1.1 Motivation

Whether it is a conversation about a new model’s performance, a recent race, or a debate over road safety regulations, velocity inevitably enters many central car-related discussions. In everyday driving contexts, the terms “speed” and “velocity” typically refer to the vehicle’s longitudinal velocity, as displayed by the speedometer. Beyond its relevance to the driver, longitudinal velocity serves as a key performance metric during vehicle development and testing and as a fundamental input to numerous control systems in modern vehicles. Systems such as Electronic Stability Control, Anti-lock Braking Systems, and torque vectoring rely on accurate real-time velocity information to perform safety-critical and performance-oriented functions. Inaccurate information can propagate into degraded control performance or unintended interventions.

Despite its central role, longitudinal velocity cannot be directly measured by a single on-board sensor with sufficient accuracy and reliability in production vehicles. Instead, it must be estimated based on multiple imperfect sensor signals. In production vehicles, longitudinal velocity estimation typically relies on wheel speed sensors and inertial measurement units (IMUs), sometimes supplemented by Global Navigation Satellite System (GNSS) measurements. However, each of these sensors presents limitations. IMU-based velocity estimation relies on acceleration integration and is therefore subject to drift without external correction. Wheel speed measurements become unreliable under high-slip conditions, leading to significant deviations between measured wheel speed and true vehicle velocity. GNSS measurements may degrade or become unavailable in environments such as tunnels, underground structures, or dense urban areas. Alternative sensing solutions, such as Light Detection and Ranging (LiDAR) or optical ground speed sensors, exist but may suffer from occlusions (e.g. optical sensor covered in mud) and are generally too expensive for production use.

To combine information from multiple sensors and mitigate their individual limitations, sensor fusion algorithms are employed. In industry, a well-established approach is to rely on mathematical models of the vehicle or its subsystems to generate an estimate from multiple sensor signals. These methods are commonly classified as model-based estimation and, by relying on physical insight and structured system modelling, offer interpretability and predictable behaviour under known operating conditions. While model-based approaches remain widespread in production applications, increased data availability and improved computational capabilities have enabled the exploration of data-driven techniques, particularly machine learning, as alternative or complementary approaches to sensor fusion and state estimation. The systematic investigation and comparative evaluation of these approaches is therefore of significant practical interest.

1.2 Problem Statement, Objectives and Scope

This thesis investigates longitudinal velocity estimation through the development and evaluation of model-based, data-driven, and hybrid estimators, where hybrid denotes the combination of model-based and data-driven techniques.

The estimators are developed and evaluated using data generated by a Driver-in-the-Loop simulator based on a high-fidelity model of a high-performance production vehicle. The simulations are performed using a passive vehicle configuration in order to avoid interference from active control systems and to provide a consistent baseline for evaluation. The estimators are developed relying primarily on commonly available onboard sensors, such as wheel speed sensors and an inertial measurement unit, while the use of additional vehicle signals available within the reference platform is also explored. GNSS measurements are not considered in the present work, nor is variation in the tyre-road friction coefficient. In the configuration considered in this work, the simulator provides idealized sensor signals without measurement noise. To emulate realistic sensing conditions, measurement noise and signal corruption are introduced.

The datasets are provided by the industrial partner of this thesis, who additionally evaluated one of their in-house estimators on the same data to serve as a reference. In total, three datasets corresponding to three distinct simulator runs are considered. Each run follows a unique driving trajectory, collectively covering both track-oriented and general driving scenarios. The datasets are analysed in Chapter 4. To ensure a fair comparison to the reference estimator, as well as among the proposed solutions themselves, two datasets are used for training and parameter tuning, while the third dataset is used for performance evaluation.

The primary objectives of this work are the development of longitudinal velocity estimators based on model-based, data-driven, and hybrid design paradigms, and the comparative evaluation of their estimation performance within a common experimental framework. The estimators are evaluated relative to one another and alongside a reference estimator provided by the industrial partner. The main quantitative metrics considered are root-mean-square error and maximum absolute error. In addition, the behaviour and limitations of the different solutions are analysed within the available test scenario.

In total, five estimators are presented, selected from a broader set of tested configurations. Two model-based estimators were developed using Kalman filtering, one based on a kinematic vehicle model and another incorporating wheel dynamics and longitudinal slip. Two purely data-driven estimators were implemented using neural networks, specifically a Multilayer Perceptron (MLP) and a Long Short-Term Memory (LSTM) architecture, operating directly on sensor inputs. Finally, a hybrid approach combining model-based and data-driven estimation is presented.

1.3 Thesis Structure

This thesis is organised as follows.

Chapter 2 introduces the theoretical foundations required for longitudinal velocity estimation.

In particular, vehicle kinematics and dynamics are reviewed, together with the sensing principles of wheel-speed sensors and inertial measurement units. Fundamental concepts of state estimation and learning-based regression are also outlined.

Chapter 3 reviews existing approaches to vehicle velocity estimation, including kinematic and dynamic model-based methods as well as learning-based techniques. The chapter concludes by positioning the present work within the existing literature.

Chapter 4 defines the problem setup and development framework used in this work. The simulation environment, dataset allocation and characterisation, and the evaluation methodology used to compare estimators are described.

Chapter 5 presents the two model-based estimators developed in this thesis: a kinematic Kalman filter (KF01) and a dynamic Kalman filter (KF02) incorporating wheel dynamics.

Chapter 6 present the two purely data-driven estimators, based on Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM) neural network architectures.

Chapter 7 presents the hybrid estimator (HYB), which combines model-based estimation with a data-driven residual correction strategy.

Chapter 8 reports the experimental results and provides a comparative evaluation of all estimators across different operating conditions.

Chapter 9 summarises the main findings of the study, discusses its limitations, and outlines directions for future research.

2 Theoretical Background

Longitudinal velocity estimation combines vehicle dynamics and modelling, sensor modelling, and estimation theory. This chapter establishes the theoretical foundations necessary for the subsequent development of estimators. First, vehicle kinematics and dynamics are studied in the context of longitudinal velocity estimation. The relevant sensors and sensing principles are then introduced. Finally, the fundamental concepts of state estimation and machine learning-based regression are presented. These elements form the basis for the estimator architectures developed in the following chapters.

2.1 Vehicle Motion and Reference Frames

The study of longitudinal velocity and related phenomena requires the definition of the relevant notation and reference frames. The notation, reference frame definitions and conventions adopted in this section follow Meywerk [1]. Subsequent sections build upon these definitions where applicable. The following reference frames are considered in this work and illustrated in Figure 2.1:

1. inertial (world) reference frame $(O, \vec{e}_{ix}, \vec{e}_{iy}, \vec{e}_{iz})$,
2. vehicle reference frame $(S_{cm}, \vec{e}_{vx}, \vec{e}_{vy}, \vec{e}_{vz})$,
3. wheel reference frame $(S_{cmw}, \vec{e}_{wx}, \vec{e}_{wy}, \vec{e}_{wz})$
4. contact patch (tyre) reference frame $(S_{cp}, \vec{e}_{tx}, \vec{e}_{ty}, \vec{e}_{tz})$

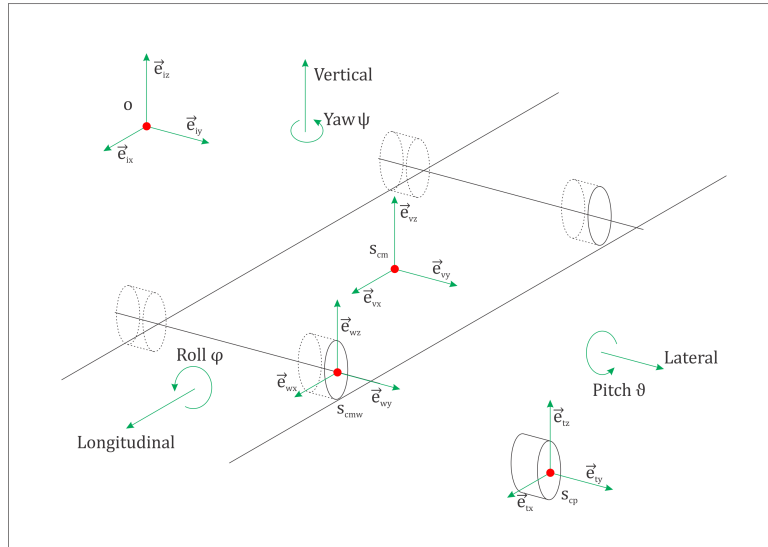


Figure 2.1: Reference frames (adapted from Meywerk [1]).

Each reference frame is defined by an origin and three mutually orthogonal unit vectors forming an orthonormal, right-handed basis. The inertial reference frame is fixed to the world (ground) and is used to express global quantities. The vehicle reference frame is fixed to the vehicle's centre of mass (CoM) and moves and rotates together with it. The three unit vectors ($\vec{e}_{vx}, \vec{e}_{vy}, \vec{e}_{vz}$) define the vehicle's longitudinal, lateral, and vertical directions, respectively. The vehicle's orientation with respect to the inertial frame is defined by three rotations: yaw (ψ) about the inertial z -axis, pitch (θ) about the intermediate y' -axis, and roll (ϕ) about the intermediate x'' -axis¹. The corresponding transformation matrix (R_{vi}) transforms vectors from the inertial to the vehicle reference frame. For velocity, this gives:

$$\mathbf{v}_v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \mathbf{R}_{vi} \dot{\mathbf{r}}_i, \quad \dot{\mathbf{r}}_i = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \quad \mathbf{R}_{vi} = \mathbf{R}_x(\phi) \mathbf{R}_y(\theta) \mathbf{R}_z(\psi). \quad (2.1)$$

where $\dot{\mathbf{r}}_i$ is the time derivative of the CoM position in the inertial frame.

The longitudinal velocity component v_x denotes the projection of the vehicle's velocity vector $\mathbf{v}$ onto the longitudinal unit vector $\vec{e}_{vx}$ of the vehicle reference frame, i.e.,

$$v_x = \mathbf{v} \cdot \vec{e}_{vx}. \quad (2.2)$$

In this project, v_x is the primary quantity of interest.

2.2 Wheel-Speed-Based Velocity Estimation

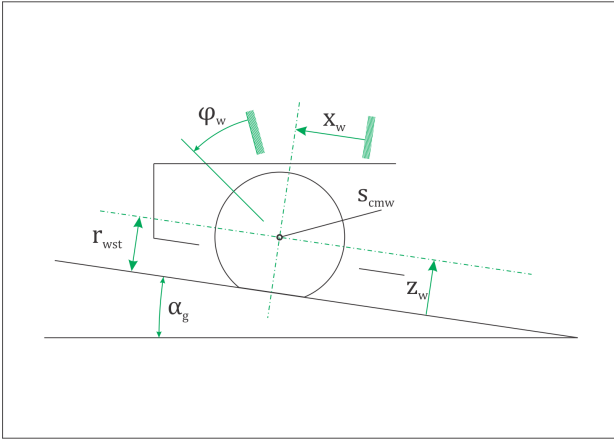
2.2.1 The Wheel: Important coordinates and the free-body diagram

Most velocity estimation methods rely, at least in part, on the angular velocity of the wheels. Utilising the reference frames defined in Section 2.1 (Vehicle Motion and Reference Frames, Fig. 2.1), important wheel coordinates are depicted in Fig. 2.2a, and a free-body diagram of the wheel is shown in Fig. 2.2b. In both figures, only the $\vec{e}_{wx} - \vec{e}_{wz}$ plane is considered.

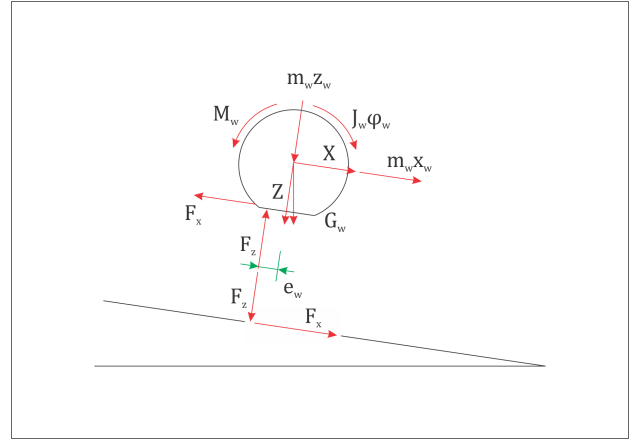
In Fig. 2.2a, the coordinates x_w and z_w describe the position of the wheel's centre of mass, measured with respect to the inertial frame. The angular displacement φ_w represents the rotation of the wheel about its lateral axis $\vec{e}_{wy}$ relative to a direction fixed in the inertial frame. The vertical distance between the wheel centre S_{cmw} and the road surface is given by the static wheel radius r_{wst} . The static radius is defined for a stationary, loaded wheel. The wheel is assumed to be on an inclined plane with an angle of inclination α_g .

In Fig. 2.2b, the quantities M_w , X , and Z act at the centre of the wheel hub. The forces X and Z represent the longitudinal and vertical reaction forces transmitted from the vehicle to the wheel, while M_w denotes the net applied torque. The centre of the hub is assumed to

¹The three angles ψ , θ , and ϕ are Euler angles following an intrinsic 3–2–1 (Z–Y'–X'') rotation sequence commonly used in vehicle dynamics. The axis y' corresponds to the lateral direction of the vehicle after the yaw rotation, and x'' to its longitudinal direction after both yaw and pitch rotations.



(a) Geometric representation



(b) Free-body diagram

Figure 2.2: Geometric representation and FBD of the wheel (adapted from Meywerk [1]).

coincide with both the centre of mass and the centre of gravity of the wheel; from this point onward, this location is simply referred to as the centre of the wheel. The wheel weight G_w acts at this point, along with the d'Alembert inertia forces $m_w \ddot{x}_w$ and $m_w \ddot{z}_w$. The angular inertia moment is expressed as $J_w \ddot{\varphi}_w$. The resultant normal and tangential forces arising from the stress distribution within the contact patch are denoted by F_z and F_x , respectively. The line of action of F_z is displaced from the wheel's vertical centreline by the eccentricity e_w .

2.2.2 Pure Rolling and the Basic Principle

To illustrate the fundamental principle of wheel-speed-based velocity estimation, a simplified case is considered. In particular, a wheel driven by torque M_w , rolling in steady state and with $F_x = 0$, is analysed. For this idealised case, the subscript $w0$ is used. Under the stated assumptions, the wheel is said to be in pure rolling and the dynamic rolling radius R_{w0} can be defined. This radius links the wheel's angular velocity Ω_{w0} and the longitudinal velocity of its centre v_{w0} .²

$$v_{w0} = R_{w0} \Omega_{w0} \quad (2.3)$$

Encoders mounted on the vehicle measure Ω_{w0} , and if R_{w0} is known, v_{w0} can be directly computed. Through simple kinematic relationships, the vehicle's longitudinal velocity v_x can then be inferred.

Although simple and intuitive, this approach presents inherent limitations. First, R_{w0} is not strictly constant. It varies with factors such as vertical load, tyre pressure, temperature, and wear. More importantly, the validity of (2.3) relies on the assumption $F_x = 0$, meaning that the tyre adheres perfectly to the road. In real driving conditions, this assumption is frequently violated. The wheels slip giving rise to longitudinal forces. In the presence of wheel slip,³

² R_{w0} differs from the static wheel radius introduced earlier and shown in Fig. 2.2a. For more on different wheel radii, see [1].

³Throughout this section (§2.2), the term *slip* refers specifically to longitudinal tyre slip, since rolling motion only along the $\vec{e}_{wx}$ direction is considered.

equation (2.3) no longer holds. In practice, R_{w0} can be estimated with sufficient accuracy such that the dominant source of error arises from tyre slip.

2.2.3 Longitudinal Tyre Slip

The longitudinal tyre slip is one of the main obstacles to accurate velocity estimation. It quantifies the relative difference between the wheel's circumferential speed and its longitudinal speed. Although an intuitive concept, there are different ways to define it, each reflecting slightly different modelling assumptions and carrying a slightly different physical meaning.

To make this work compatible with existing literature, the definition by Pacejka [2], whose tyre modelling framework is widely used in industry, is followed. Pacejka defines the *slip radius* as the distance between the centre of a wheel rolling at steady state and its instantaneous centre of rotation (in the ground reference frame), when the wheel rolls with no longitudinal slip. Physically, a wheel rolls without slip when $F_x = 0$. This corresponds to the pure rolling condition introduced earlier and thus the slip radius equals the dynamic rolling radius. For practical reasons, however, it is often more convenient to define the slip radius as equal to the effective rolling radius, R_e . The effective rolling radius is computed for a wheel rolling at steady state with no driving moment applied, i.e. $M_w = 0$, such that under these conditions

$$R_e = \frac{v_{wx}}{\Omega_w}. \quad (2.4)$$

This is the definition adopted in this work.

With the slip radius defined, longitudinal slip is given by

$$s = \frac{R_e \Omega_w - v_{wx}}{|v_{wx}|}. \quad (2.5)$$

The absolute value preserves the physical interpretation of slip in reverse driving. To avoid numerical issues around $V_x = 0$, the following variation of the definition is adopted in numerical applications in this work:

$$s = \frac{R_e \Omega_w - v_{wx}}{\sqrt{v_{wx}^2 + \epsilon_{v_x}^2}}, \quad (2.6)$$

where ϵ_{v_x} is a small value. The equations (2.5) and (2.6) are expressed in the wheel reference frame and are valid for each individual wheel.

Using (2.5) and assuming forward motion for simplicity, the wheel's longitudinal velocity can be expressed as

$$v_{wx} = \frac{R_e \Omega_w}{1 + s}. \quad (2.7)$$

The vehicle longitudinal velocity can then be obtained from v_{wx} velocities through standard vehicle kinematics. The above expression highlights that, in the presence of longitudinal slip, the angular velocity alone is insufficient to determine the vehicle longitudinal velocity.

2.2.4 Wheel Dynamics and Slip Modelling

The relation in Eq. (2.7) indicates that accurate vehicle longitudinal velocity estimation based on wheel-speed measurements requires knowledge of slip. Consequently, a dynamic description of slip is required.

Longitudinal slip is directly related to the longitudinal tyre force F_x through tyre models.⁴ The most widely known is the Pacejka tyre model, though alternatives such as Dugoff and Fiala also exist. The selected model depends on the intended application and desired level of fidelity. Applicable for small slip values, the linear tyre model provides the simplest approximation. According to the linear tyre model:

$$F_x = C_s s, \quad (2.8)$$

where C_s denotes the longitudinal tyre stiffness, which depends on the normal load F_z .

The longitudinal tyre force can also be expressed in terms of the normal load, F_z :

$$F_x = \mu_x(s) F_z, \quad (2.9)$$

where $\mu(s)$ is the tyre longitudinal force coefficient. As the notation implies, the tyre longitudinal force coefficient depends on longitudinal slip. The relationship between the two quantities is illustrated by the μ_x - s curves in Fig. 2.3. These curves, obtained under different surface conditions, show the characteristic non-linear behaviour of the μ_x - s relationship.⁵

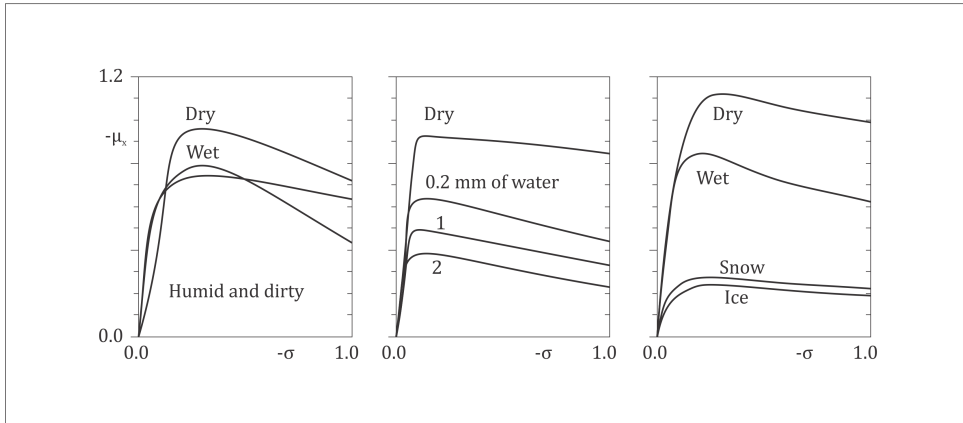


Figure 2.3: Curves of μ_x -slip under different surface conditions. Reproduced from Genta and Morello (2020, Fig. 2.35). [3]

Note: Genta and Morello adopt slightly different notation ($S \rightarrow \sigma$) and define slip such that it may take both positive and negative values, corresponding respectively to driving and braking. Since the μ_x -slip relationship is nearly symmetric, only the positive-slip branch is shown in their plots.

While the overall shape of the curves is similar, the level of the longitudinal force coefficient

⁴Tyre models are not limited to longitudinal ones but in this section we are only considering the $\vec{e}_{wx} - \vec{e}_{wz}$ plane and the wheel rolling in a straight line

⁵A typical μ -slip curve can be divided into three zones: an adhesion (linear) zone at small slip where the transmitted force increases proportionally with slip, a *transition (peak)* zone where adhesion and sliding coexist and the friction coefficient reaches its maximum, and a *sliding (saturation)* zone at large slip where most of the contact patch slides and the force decreases.

clearly varies with the road condition. Typical peak values are around $\mu_x \approx 1.0$ on dry asphalt, $\mu_x \approx 0.6$ on wet asphalt, and as low as $\mu_x \approx 0.1$ – 0.2 on snow or ice. These numbers, as well as the curves shown, should be considered indicative averages since the results depend on many factors and vary between experiments. High-performance tyres, such as those used on the reference vehicle in this work, can reach higher peak levels - up to about 1.5–1.8 [3]. This has a direct impact on the wheel dynamics. On high-grip surfaces, the wheel can transmit higher longitudinal forces before noticeable slip occurs, whereas on low-grip surfaces even moderate acceleration or braking can lead to substantial slip.

The longitudinal tyre force coefficient links the normal load F_z to the longitudinal tyre force F_x , whereas the discussed tyre models describe the relationship between F_x and longitudinal slip. Consequently, if F_x can be estimated, slip may be inferred. Another approach to estimating F_x is to analyse the free-body diagram of the wheel (Fig. 2.2). The corresponding equilibrium equations are:

$$m_w \ddot{x}_w = F_x - X - G_w \sin \alpha_g, \quad (2.10)$$

$$m_w \ddot{z}_w = F_z - Z - G_w \cos \alpha_g, \quad (2.11)$$

$$J_w \ddot{\varphi}_w = M_w - F_x r_{wst} - F_z e_w. \quad (2.12)$$

From the rotational equilibrium equation ((2.12)), the longitudinal tyre force can be expressed as

$$F_x = \frac{M_w - J_w \ddot{\varphi}_w}{r_{wst}} - \frac{e_w}{r_{wst}} F_z, \quad (2.13)$$

$$F_x = \frac{M_w - J_w \ddot{\varphi}_w}{r_{wst}} - f_r F_z, \quad f_r = \frac{e_w}{r_{wst}}. \quad (2.14)$$

The ratio f_r represents the coefficient of rolling resistance, which can be estimated empirically (as a function of velocity).⁶ In some vehicles, the applied wheel torque M_w can be estimated from drivetrain data, while J_w and r_{wst} are known parameters. The angular acceleration $\ddot{\varphi}_w$ is not directly measurable but can be inferred, for example, through discrete differentiation of measured angular velocity. Under these assumptions, F_x can, in principle, be estimated from the rotational equilibrium equation.

It should be emphasised that derivations presented so far in this section were not the only modelling path, but are illustrative of the general approach. For example, similar reasoning may be applied to the longitudinal equilibrium equation ((2.10)), where X must be estimated from vehicle-level dynamics. This requires additional modelling assumptions, for instance regarding aerodynamic forces.

In either case, once an estimate of F_x is available, slip may be inferred through the tyre force–slip relationship, enabling an improved estimate of the longitudinal vehicle velocity.

⁶An empirical formula for the rolling-resistance coefficient is $f_r(v) = \tilde{f}_{r0} + \tilde{f}_{r1} \frac{v}{v_0} + \tilde{f}_{r4} \left(\frac{v}{v_0} \right)^4$ [1].

2.2.5 Modelling Assumptions and Practical Limitations

The preceding derivations show that wheel dynamics contain information that can, in principle, be used to improve the estimate of v_x . However, several practical limitations must be acknowledged. The equilibrium equations ((2.10)–(2.12)) neglect various loss mechanisms for simplicity, whereas a more realistic wheel model would include them. Most of the losses can be summarised by an additional torque T_{loss} in the rotational equilibrium:

$$J_w \ddot{\varphi}_w = M_w - F_x r_{wst} - F_z e_w - T_{loss}. \quad (2.15)$$

Such losses arise from bearing friction, toe-angle effects,⁷ aquaplaning, and similar phenomena.

Moreover, the previous discussion relied on estimated or empirically identified quantities, such as the longitudinal tyre stiffness C_s and angular acceleration, and on simplifying approximations including the small-slip assumption. The selection of the modelling approach and the level of complexity therefore represents a design trade-off between physical fidelity and practical implementability. Despite these limitations, the derivations remain valuable, as they clarify the underlying relationships and identify variables that are informative for longitudinal velocity estimation.

A further limitation of the preceding discussion is that it considered a single wheel rolling in the $\vec{e}_{wx}$ – $\vec{e}_{wz}$ plane. In a real vehicle, the wheels rotate at different speeds, are dynamically coupled through the vehicle structure, and the vehicle motion cannot be described in one dimension only. Even if only longitudinal motion is considered, differences in wheel speeds naturally arise due to variations in tyre radii and slip. To investigate motion of the whole vehicle, the framework must be expanded to appropriate vehicle-level modelling.

2.3 Vehicle-level modelling

To describe the motion of the vehicle as a whole, a vehicle model is required. In this work, planar vehicle models are the primary focus, which means the vehicle’s centre of mass (CoM) is assumed to lie on the plane of the trajectory. Even under this strong simplification, many aspects relevant to velocity estimation can be examined, introducing lateral motion and important quantities such as the vehicle sideslip angle and the tyre slip angle. In particular, the two-track planar model is analysed, and both kinematic and dynamic formulations are considered. Subsequently, the discussion is expanded considering selected non-planar aspects.

2.3.1 Kinematic Model

The planar two-track vehicle model is shown in Fig. 2.4.

The vehicle is modelled as a rigid body in the horizontal plane. A body-fixed reference frame is attached to the centre of mass (CoM), with longitudinal and lateral axes denoted by $\vec{e}_{vx}$ and

⁷Toe angle refers to the inclination of the wheel plane relative to the vehicle’s longitudinal axis.

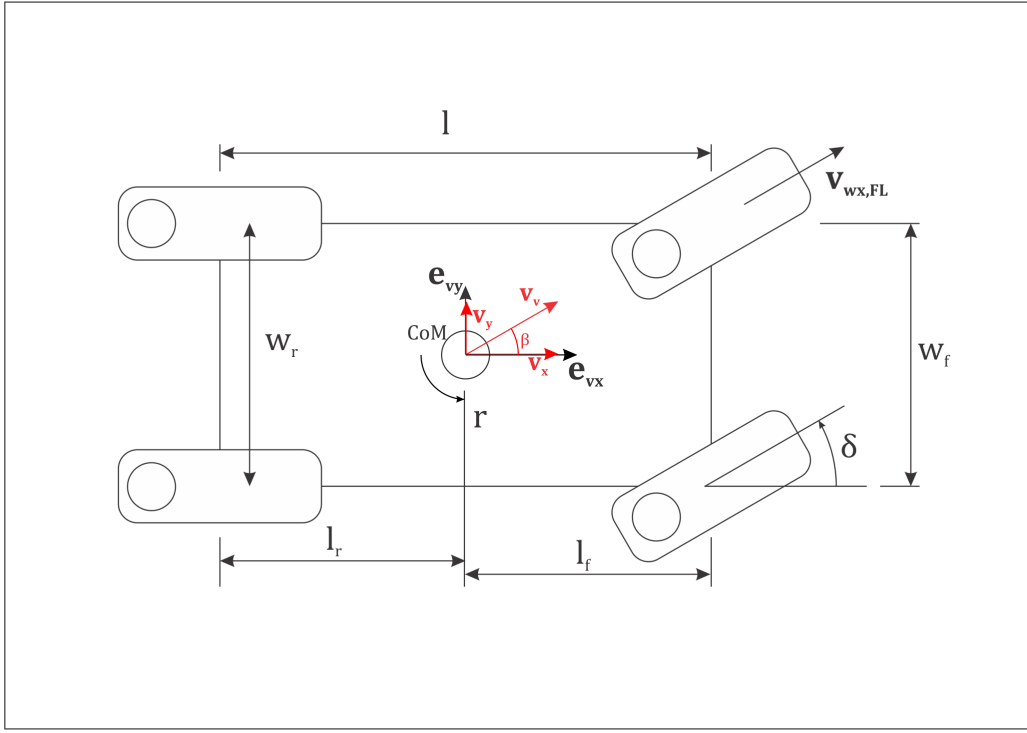


Figure 2.4: Planar two-track model: geometric representation.

$\vec{e}_{vy}$, respectively. The distances from the CoM to the front and rear axles are l_f and l_r , while the front and rear track widths are denoted by w_f and w_r . The vehicle is assumed to have front-axle steering only, with the steering angle δ assumed equal for the left and right front wheels. The longitudinal and lateral velocity components of the vehicle at the CoM are v_x and v_y , and the yaw rate is r . The vehicle sideslip angle β is defined as the angle between the vehicle velocity vector $\vec{v}_v$ and the longitudinal axis $\vec{e}_{vx}$ (Eq. (2.16)). Each wheel $i \in \{\text{FL}, \text{FR}, \text{RL}, \text{RR}\}$ has its own velocity vector $\vec{v}_{w_i}$. For each wheel, the tyre slip angle α_i is defined as the angle between the wheel velocity vector $\vec{v}_{w_i}$ and the longitudinal axis of the corresponding wheel (Eq. (2.17)).

$$\beta = \arctan\left(\frac{v_y}{v_x}\right) \quad (2.16)$$

$$\alpha_i = \arctan\left(\frac{v_{y,i}}{v_{x,i}}\right), \quad (2.17)$$

With the defined quantities and stated assumptions, the kinematics of the planar two-track model can now be analysed, focusing on the relationships relevant to longitudinal velocity estimation. Using the sideslip angle, it is possible to isolate the longitudinal and lateral components of the velocity.

$$\vec{v} = v_x \vec{e}_{vx} + v_y \vec{e}_{vy} = v_v (\cos \beta \vec{e}_{vx} + \sin \beta \vec{e}_{vy}) \quad (2.18)$$

$$v_x = v_v \cos \beta, \quad v_y = v_v \sin \beta. \quad (2.19)$$

Differentiating (2.18) and using planar yaw kinematics, $\dot{\vec{e}}_{vx} = r \vec{e}_{vy}$ and $\dot{\vec{e}}_{vy} = -r \vec{e}_{vx}$, yields

$$\vec{a} = \dot{v}_x \vec{e}_{vx} + \dot{v}_y \vec{e}_{vy} + v_x \dot{\vec{e}}_{vx} + v_y \dot{\vec{e}}_{vy} \quad (2.20)$$

$$= (\dot{v}_x - v_y r) \vec{e}_{vx} + (\dot{v}_y + v_x r) \vec{e}_{vy}. \quad (2.21)$$

The resulting body-frame acceleration components are

$$a_x = \dot{v}_x - v_y r, \quad a_y = \dot{v}_y + v_x r. \quad (2.22)$$

These relations are of particular relevance in this work, as the on-board IMU provides measurements of the body-frame specific forces, which are directly related to the vehicle accelerations, and the three-axis angular velocity.

Moreover, the angular velocities of each wheel can be expressed in terms of the CoM velocity components v_x and v_y and the contribution induced by the yaw rate r . First, for each wheel, the longitudinal wheel velocity is defined as the projection of the wheel-centre velocity onto the longitudinal axis of the corresponding wheel. It is important to stress that the projection of the wheel-centre velocity is onto the longitudinal axis of the corresponding wheel, $\vec{e}_{wx,i}$, and not onto the vehicle body longitudinal axis $\vec{e}_{vx}$. In the following, no slip conditions are assumed, i.e. $s_i = 0$ and $\alpha_i = 0$.

$$v_{wx,FL} = \left(v_x - r \frac{w_f}{2} \right) \cos \delta + (v_y + r l_f) \sin \delta \quad (2.23)$$

$$v_{wx,FR} = \left(v_x + r \frac{w_f}{2} \right) \cos \delta + (v_y + r l_f) \sin \delta \quad (2.24)$$

$$v_{wx,RL} = v_x - r \frac{w_r}{2} \quad (2.25)$$

$$v_{wx,RR} = v_x + r \frac{w_r}{2} \quad (2.26)$$

Then, the angular of each wheel is simply given by:

$$\Omega_i = \frac{v_{wx,i}}{R_{e,i}}, \quad (2.27)$$

where $R_{e,i}$ denotes the effective rolling radius of wheel i . If longitudinal slip is considered, the relationship becomes

$$\Omega_i = \frac{(1 + s_i) v_{wx,i}}{R_{e,i}}, \quad (2.28)$$

for the driving case ($s_i > 0$), where s_i denotes the longitudinal slip ratio.

2.3.2 Dynamic Modelling

To study the dynamics of the planar two-track model, its free-body diagram is shown in Fig. 2.5

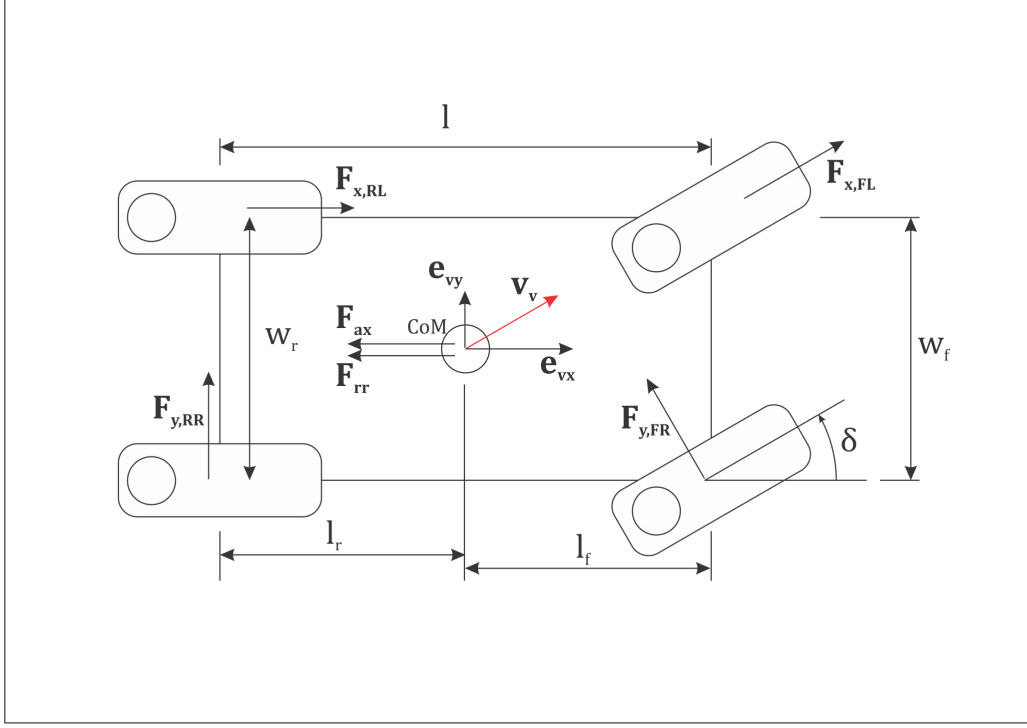


Figure 2.5: Planar two-track model: free-body diagram.

The geometric quantities and kinematic relationships introduced previously for the planar two-track vehicle model remain unchanged, but the analysis is now expanded to consider dynamics as well. The vehicle is assumed to have constant mass m and yaw moment of inertia I_z about the vertical axis through the CoM. Each wheel $i \in \{FL, FR, RL, RR\}$ generates a longitudinal tyre force $F_{x,i}$ and a lateral tyre force $F_{y,i}$ acting at the respective wheel centre. Aerodynamic drag is modelled as a lumped longitudinal force F_{ax} acting at the CoM along the negative vehicle longitudinal direction, opposing the direction of travel. Lateral aerodynamic forces are neglected, i.e. $F_{ay} = 0$. Rolling resistance effects are represented by a single equivalent longitudinal rolling resistance force F_{rr} acting at the CoM and opposing the direction of travel. Road slope effects are neglected, and the road is assumed to be flat such that no gravitational components act in the longitudinal or lateral directions.

With the stated assumptions the following equilibrium equations are derived:

$$m(\dot{v}_x - rv_y) = (F_{x,FL} + F_{x,FR}) \cos \delta - (F_{y,FL} + F_{y,FR}) \sin \delta + F_{x,RL} + F_{x,RR} - F_{ax} - F_{rr} \quad (2.29)$$

$$m(\dot{v}_y + rv_x) = (F_{y,FL} + F_{y,FR}) \cos \delta + (F_{x,FL} + F_{x,FR}) \sin \delta + F_{y,RL} + F_{y,RR} \quad (2.30)$$

$$\begin{aligned} I_z \dot{r} = & l_f [(F_{y,FL} + F_{y,FR}) \cos \delta + (F_{x,FL} + F_{x,FR}) \sin \delta] - l_r (F_{y,RL} + F_{y,RR}) \\ & + \frac{w_f}{2} [(F_{x,FR} - F_{x,FL}) \cos \delta + (F_{y,FR} - F_{y,FL}) \sin \delta] + \frac{w_r}{2} (F_{x,RR} - F_{x,RL}) \end{aligned} \quad (2.31)$$

An important aspect that is not directly visible from the equilibrium equations is the coupling of the longitudinal and lateral tyre forces through the friction limit at the tyre-road interface. The generation of lateral force reduces the available longitudinal force capacity and vice versa, commonly represented by the friction ellipse constraint. While a detailed analysis can be found in vehicle dynamics literature, the key implication in the context of this project is that, during cornering manoeuvres, a portion of the available tyre force is used to generate lateral acceleration, limiting the longitudinal force that can be transmitted for propulsion or braking. This combined-slip interaction modifies the relationship between wheel angular velocity and vehicle longitudinal velocity.

Load Transfer Effects

While the planar two-track model captures the essential kinematic and dynamic relationships at vehicle level, additional insight can be obtained by considering a non-zero CoM height h_{cm} above the road surface, which introduces load transfer effects.

Load transfer may occur between inner and outer wheels or between the front and rear axles, depending on the acceleration direction. Taking longitudinal load transfer as an example, and neglecting pitch dynamics while assuming a rigid body and constant longitudinal acceleration, the load redistribution can be approximated as

$$\Delta F_{z,\text{long}} = \frac{1}{2} \frac{m a_x h_{cm}}{l}. \quad (2.32)$$

Load transfer implies that the normal force acting on each tyre varies with the driving condition. Since the achievable longitudinal and lateral tyre forces depend on the normal load, this directly affects the effective friction limits and, consequently, the vehicle's susceptibility to slip. Furthermore, the distribution of normal loads provides indirect information about the driving mode, as predominantly lateral load transfer indicates cornering, whereas dominant longitudinal transfer corresponds to acceleration or braking.

The dynamic relationships presented above can be used to infer information about quantities relevant to longitudinal velocity estimation, such as the longitudinal tyre force F_x . As in the wheel-level analysis, the usefulness of such relations depends on the chosen trade-off between model fidelity and complexity.

More generally, the kinematic and dynamic formulations developed at both wheel and vehicle level establish a structured connection between the longitudinal velocity and other measurable quantities. At the same time, the analysis has highlighted inherent physical limitations and modelling approximations. In practice, additional challenges arise from sensor imperfections.

2.4 Sensors

This project deals with v_x estimation utilising on-board production-grade sensors. In this chapter, wheel-speed sensors and the IMU have already been mentioned due to their obvious

applicability in the context of the presented modelling framework. There are, however, other alternative or complementary sensors used, most widely employed being the Global Navigation Satellite System (GNSS). In the following, encoders, IMU, GNSS and alternative sensor solutions are discussed in the context of v_x estimation.

2.4.1 Wheel-speed sensors

Wheel-speed sensors are typically based on a toothed wheel attached to a rotating shaft. The edge of each tooth can be detected by different sensing mechanisms, but the fundamental principle is generally the same. The angular distance between successive tooth edges is determined by the wheel geometry, and by measuring the elapsed time between their detection, the angular velocity can be computed. For an ideal toothed wheel with N_{cog} equally spaced teeth, and two consecutive detection times t_k and t_{k-1} , the angular velocity can be approximated as

$$\hat{\omega}_k = \frac{2\pi}{N_{\text{cog}}(t_k - t_{k-1})}. \quad (2.33)$$

In automotive applications, wheel-speed sensors are typically magnetic encoders. According to Gustafsson [4], the main factors influencing their performance are sampling-time effects, quantization errors, and imperfections in the toothed wheel geometry.

2.4.2 Inertial Measurement Unit (IMU)

Ideal IMU measurement model

An IMU measures *specific force* and angular rates along its local axes. For an ideal sensor located at point s of the vehicle, the specific force $\mathbf{f}$ expressed in the body frame is given by

$$\mathbf{f} = \dot{\mathbf{v}}_s + \boldsymbol{\omega} \times \mathbf{v}_s - \mathbf{g}, \quad (2.34)$$

where $\mathbf{v}_s$ is the velocity of the sensor location in the body frame, $\boldsymbol{\omega} = [p, q, r]^T$ is the angular rate vector, and $\mathbf{g}$ is the gravity vector expressed in the body frame.

If the IMU is mounted at the vehicle centre of mass (CoM), then $\mathbf{v}_s = \mathbf{v}$, where $\mathbf{v} = [v_x, v_y, v_z]^T$ denotes the vehicle velocity in the body frame. Ideally, the IMU is mounted at the vehicle CoM so that the measured quantities directly correspond to the vehicle motion. In practice, this is sometimes impossible due to space constraints. The measurements can, however, be transferred to the CoM using rigid-body kinematic relations.

Assuming planar vehicle motion ($v_z = 0$), the longitudinal component of (2.34) becomes

$$f_x = \dot{v}_x - r v_y + g \sin \theta, \quad (2.35)$$

where $r = \dot{\psi}$ is the yaw rate and θ the pitch angle.

Accelerometers measure specific force rather than linear acceleration, so gravity compensation based on the estimated vehicle orientation is required to recover the linear acceleration

components.

Estimating v_x with the IMU

Since the IMU provides measurements related to linear acceleration and angular velocity, it can be used for longitudinal velocity estimation. Following the planar kinematic relationship derived in Eq. (2.22), the longitudinal velocity can be obtained as

$$v_x(t) = v_x(t_0) + \int_{t_0}^t \left(a_x(\tau) + v_y(\tau)r(\tau) \right) d\tau. \quad (2.36)$$

Non-idealities

Although the IMU provides highly informative signals for v_x estimation, inertial integration leads to error accumulation over time. Biases and noise components are integrated together with the signal, causing drift in the velocity estimate. Therefore, inertial velocity estimation is reliable only over short time intervals or when periodically corrected using additional measurements.

In practice, IMU measurements are affected by bias, scale-factor errors, and stochastic processes such as bias instability and random walk. IMU noise characteristics reported in manufacturer datasheets are typically obtained under controlled laboratory conditions. In practical applications, however, the effective measurement noise may be higher due to environmental and integration effects. Factors such as structural vibrations, temperature variations, electronic interference, and mounting imperfections can degrade the observed performance of inertial sensors relative to nominal specifications. For this reason, inertial sensor noise parameters are often characterized experimentally rather than relying solely on datasheet values [5, 6].

2.4.3 GNSS, Alternative Sensing Solutions and Sensor Fusion

Global Navigation Satellite Systems (GNSS) provide absolute position and velocity information globally by processing signals transmitted from satellite constellations. Multiple approaches for estimating vehicle velocity using GNSS signals exist, including those based on the temporal evolution of position measurements and those measuring Doppler shift of the received signals. GNSS measurements are affected by signal attenuation and interruption in environments such as tunnels, underground structures, or dense urban areas. Another common issue is measurement latency. Despite these limitations, when available, GNSS can serve as an additional source of information for longitudinal velocity estimation and as a reference for correcting other velocity estimates.

In addition to wheel-speed sensors, IMUs and GNSS, other technologies such as Light Detection and Ranging (LiDAR) or optical ground speed sensors can be used for vehicle velocity estimation. These systems are not commonly found in standard production vehicles and are not available on the reference vehicle considered in this work.

As discussed in this chapter, multiple sensing solutions provide information relevant to longitudinal velocity estimation, each with its own characteristics and limitations arising from both sensor imperfections and the underlying physical relationships. Sensor fusion techniques are employed to combine information from multiple sensors in order to mitigate their individual limitations and obtain a more accurate estimate. State estimation techniques provide a mathematical framework for performing such fusion.

2.5 State Estimation

2.5.1 Model-Based Estimation

Longitudinal velocity estimation can be formulated as a state estimation problem, in which the internal vehicle states evolve according to dynamic equations and are observed through imperfect sensor measurements. Such systems are commonly represented in state-space representation.

In discrete-time, the system can be expressed as

$$x_{k+1} = f(x_k, u_k) + w_k, \quad (2.37)$$

$$y_k = h(x_k, u_k) + v_k. \quad (2.38)$$

where x_k denotes the state vector, u_k the system inputs, y_k the measurement vector, and w_k and v_k represent process and measurement noise, respectively. The Eq. (2.37) and Eq. (2.38) are referred to as the process (motion) model and the measurement model, respectively.

State observers aim to exploit the knowledge of y_k in order to estimate x_k . Real-world systems are corrupted by modelling errors and sensor noise, so estimation performance depends on how such uncertainties are handled. Kalman filters provide a statistically grounded framework for state estimation. The Kalman filter is a recursive method that uses a system model to predict state evolution and incorporates sensor measurements to update this prediction.

For a discrete-time, linear time-varying system, the general expressions in Eqs. (2.37)–(2.38) reduce to

$$x_{k+1} = A_k x_k + B_k u_k + w_k, \quad (2.39)$$

$$y_k = H_k x_k + v_k, \quad (2.40)$$

where A_k , B_k , and H_k denote the system matrices, and w_k and v_k represent zero-mean process and measurement noise with covariance matrices Q_k and R_k , respectively.

For the system in Eqs. (2.39)–(2.40), the discrete-time Kalman filter can be applied. The corresponding state estimate is obtained through a recursive prediction and update procedure.

Prediction

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}, \quad (2.41)$$

$$P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^\top + Q_{k-1}, \quad (2.42)$$

Update

$$K_k = P_{k|k-1}H_k^\top \left(H_k P_{k|k-1} H_k^\top + R_k \right)^{-1}, \quad (2.43)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left(y_k - H_k \hat{x}_{k|k-1} \right), \quad (2.44)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}. \quad (2.45)$$

Here, $\hat{x}_{k|k-1}$ and $\hat{x}_{k|k}$ denote the predicted and updated state estimates, respectively, $P_{k|k-1}$ and $P_{k|k}$ the corresponding estimation error covariance matrices, K_k the Kalman gain, and $y_k - H_k \hat{x}_{k|k-1}$ the innovation.

The discrete-time Kalman filter presented above is derived for linear systems. However, since many vehicle motion models are nonlinear, the Extended Kalman Filter (EKF) is commonly employed. The EKF linearizes the functions $f(x_k, u_k)$ and $h(x_k, u_k)$ around the state estimate at each time step and then works with the obtained linearized system.

2.5.2 Data-Driven Estimation

In this work, data-driven estimation refers to approaches that do not rely explicitly on a physics-based model of vehicle motion. Instead, the estimation is performed directly from the data, using available signals or features derived from them. Given that the objective of this project is to estimate a numerical quantity arising from a complex dynamical system, and that representative historical data are available, machine learning (ML) is well suited to the problem. In particular, supervised learning techniques can be used to tackle velocity estimation as a regression task.

Within the ML framework, neural networks are widely adopted due to their expressive approximation capabilities and architectural versatility. Neural networks are parametric models whose complexity can be adjusted through their architecture, allowing the estimator to be designed according to computational constraints. Different network structures, such as feedforward networks and recurrent neural networks and (e.g., LSTMs), enable the modelling of both static and temporal relationships in data.

Multilayer Perceptron (MLP)

Multilayer Perceptrons (MLPs) are feedforward neural networks. The schematic representation of an MLP with two hidden layers (the two middle layers) is shown in Fig. 2.6.

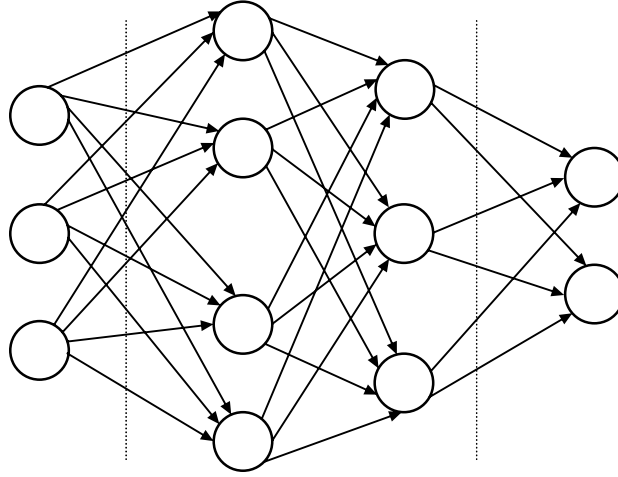


Figure 2.6: Schematic representation of the MLP architecture.

The leftmost layer represents the input layer, while the rightmost layer corresponds to the output layer. The layers are fully connected, meaning that each neuron in a given layer is connected to all neurons in the subsequent layer through trainable weights and biases, followed by a nonlinear activation function that enables the network to represent nonlinear mappings between inputs and outputs.

Depending on the application, the complexity of the network can be selected, as well as the activation functions. Being feedforward networks, MLPs can learn instantaneous nonlinear mappings.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks belong to the class of recurrent neural networks. With recurrent connections, internal memory states and gating mechanisms, LSTMs are designed to model temporal dependencies in sequential data. This makes them particularly suitable for applications where the target quantity does not only depend on current inputs but their temporal evolution as well. As for the MLPs, the size and complexity of the network is tailored to the intended application.

3 Literature Review

3.1 Overview

Vehicle longitudinal velocity estimation belongs to a wider field of vehicle state estimation, which is widely studied in both industry and academia. A technical survey by Jin et al. [7] analyses vehicle dynamic state estimation from the perspectives of vehicle models, estimations techniques, sensor configurations, and advanced estimation methodologies. A more recent review by Wang et al. [8], focused on autonomous ground vehicles, classifies state estimation according to estimation objects, vehicle characteristics and estimation algorithms. Within this broader landscape, the present work focuses specifically on longitudinal velocity estimation in a narrowed domain. The high-level conceptual framework considered in this thesis is illustrated in Fig. 3.1 below.

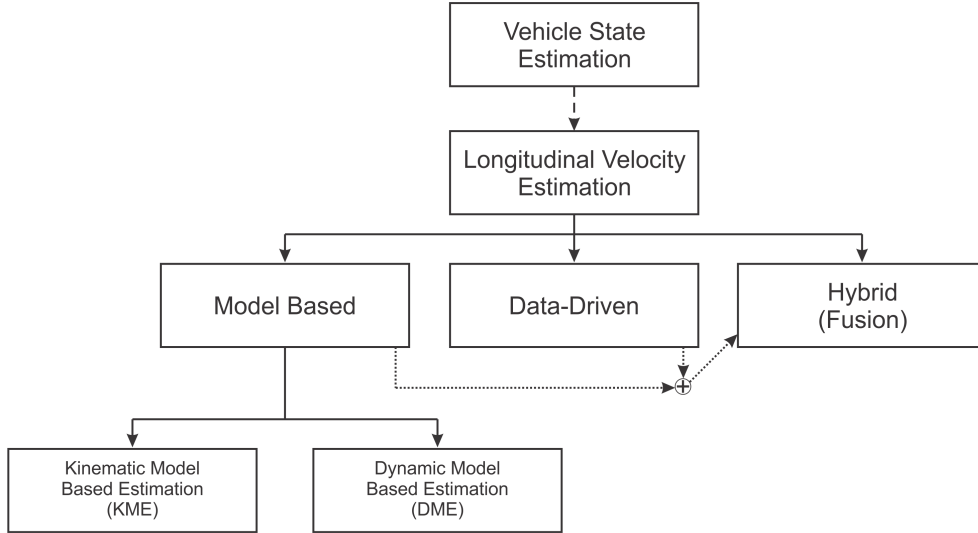


Figure 3.1: Schematic representation of the state estimation domain considered in this work.

In terms of estimation algorithms, a clear distinction is made between model-based, data-driven and hybrid (fusion) approaches, with model-based estimators further classified into those derived from kinematic vehicle models and those based on dynamic models [7, 8]. Amongst model-based estimators, the usual choice is between deterministic observers and Kalman filters. While nonlinear observers are sometimes preferred due to lower computational complexity [9], Kalman filters are quoted as most widely adopted [8].

3.2 Kinematic Model-Based Estimators

Since they do not depend on nonlinear vehicle dynamics, kinematic-model-based estimators (KMEs) tend to be simpler when compared to dynamic-model-based estimators (DMEs). As a result, they are generally easier to implement, less computationally demanding, and less susceptible to performance degradation caused by modelling uncertainties. [REF]. However, KMEs are known to generally experience significant performance degradation under high-slip driving conditions, in which case they commonly rely on GNSS signals for correction [8, 10]. A key limitation of this approach is that GNSS measurements are affected by signal attenuation and interruption in complex environments. While mitigating this issue is an active area of research [11, 12], GNSS measurements are not considered in this work. Even so, the previously mentioned advantages keep KMEs relevant in the field and of great interest to this project. Seo and Kang [13] propose a kinematic model based vehicle velocity estimator, using IMU, steering angle and wheel speed sensor inputs, demonstrating performance advantages when compared to a standard DME. The solution proposed in [13] relies on two vehicle models (the bicycle model and the two-track model), two Kalman filters (a standard and a linear one), and a gating mechanism to switch between different kinematic constraints. While the overall estimation algorithm is complex, the reported performance demonstrates the potential of KMEs.

3.3 Dynamic Model-Based Estimators

As discussed in Chapter 2, the advantage of considering dynamics is the possibility to infer quantities relevant to longitudinal velocity estimation, such as tyre slip. Dynamic models, however, rely on parameters and variables that are often uncertain or not directly measurable, such as vehicle inertia and tyre characteristics. A trade-off is made between model fidelity and complexity. DMEs rely on the coupling of vehicle and tyre models since tyre-road interaction governs the forces driving vehicle motion. Vehicle models range from the simple longitudinal model to high-fidelity multi-degree-of-freedom formulations, while tyre models span from linear tyre models to complex nonlinear formulations such as the Pacejka model [7].

In [14], a nonlinear observer (NLO) based on a 3-DOF vehicle model and the Dugoff tyre model is presented. The work reports overall improved performance compared to an NLO based on a lower-fidelity vehicle model and an EKF, but it also highlights increased estimation errors during sudden steering manoeuvres. A more recent work by Hashemi et al. [10] argues that tyre models have not been widely adopted in velocity estimators due to uncertain road friction and varying tyre parameters. They propose an estimator that utilises a simplified LuGre tyre model with a reduced set of tyre parameters and achieves robust velocity estimation even in the presence of model uncertainties.

While the present work does not consider variations in the tyre-road friction coefficient, other parameter uncertainties, as well as model complexity, remain relevant design considerations. Although EKFs are widely used, they can suffer from high computational complexity due

to linearisation and matrix operations at each time step, particularly when applied to high-fidelity vehicle models. The work by Guo et al. [15] proposes an implementation scheme to improve the computational efficiency of EKF for velocity estimation. The authors develop an EKF based on a 7-DOF vehicle model, combining vehicle and wheel dynamics and obtaining a 19-dimensional state vector. In addition, [15] employs a second-order random walk model for tyre forces in order to avoid explicit tyre parameter identification and reduce estimator complexity. Although ECU implementation feasibility is not explicitly evaluated in the present thesis, computational complexity and practical implementability are considered during the estimator design process.

Overall, the literature demonstrates that accurate velocity estimation can be achieved using a wide range of modelling approaches for DMEs. Nevertheless, estimator complexity, computational requirements, and robustness to parameter uncertainties remain important practical considerations.

3.4 Learning-Based Estimation Approaches

Contrary to model-based approaches, data-driven estimation techniques do not require an explicit vehicle model. While there is a broad range of data-driven techniques, this work focuses on machine learning (ML), and in particular on neural networks (NNs), which are among the most widely adopted learning-based tools in vehicle state estimation [7]. Owing to their versatility and ability to map highly complex and nonlinear relationships, NNs have been applied to various estimation tasks, including tyre force and vehicle sideslip angle estimation [16, 17]. In addition, ML approaches can learn to compensate for measurement noise and environmental disturbances [8], which is particularly relevant given that longitudinal velocity estimation relies on inherently noisy sensors such as the IMU. Most importantly, neural networks have also been explored directly for longitudinal velocity estimation [18]. In the work by Napolitano Dell’Annunziata et al. [18], neural networks were employed to estimate longitudinal velocity for a high-performance vehicle using data collected from multiple tracks, making the application comparable to the present study. Both feedforward and recurrent architectures were trained using signal data available from onboard sensors. The recurrent neural networks were constructed by combining two feedforward neural networks, where the second network received the same inputs as the first, augmented with the longitudinal velocity estimated at the previous time step. While the authors report good overall estimation performance, they observe no significant improvement when employing the recurrent architectures. While the estimators developed in [18] were entirely based on ML, neural networks can also be combined with other estimation strategies. In [19], a hybrid method combining a convolutional neural network and a Kalman filter is proposed for improved estimation of the true vehicle acceleration and angular velocity. This can subsequently be used to enhance velocity estimation.

Despite the accelerating spread of ML techniques, it is important to stress that the performance of ML-based architectures is highly dependent on data availability and quality. Further-

more, unless online learning strategies are employed, neural networks may struggle to maintain performance under changing system conditions and would require retraining. Nonetheless, the growing body of literature and the increasing diversity of ML techniques suggest both significant potential and sustained interest in their application to vehicle state estimation.

3.5 Discussion and Positioning of the Present Work

The literature demonstrates a wide range of approaches to vehicle longitudinal velocity estimation. Broadly, these approaches can be divided into model-based and data-driven methods, with model-based techniques further classified into kinematic and dynamic formulations. Hybrid approaches combine elements of model-based and data-driven estimation. Successful velocity estimators have been reported in each of these categories. However, important limitations remain. While kinematic models tend to be simpler and less dependent on uncertain parameters than dynamic model-based estimators, they are generally unable to capture system behaviour under highly dynamic conditions, often resulting in degraded performance. The potential for higher model fidelity offered by dynamic model-based estimators comes with the risk of increased model complexity. When combined with algorithms such as the extended Kalman filter, this complexity can lead to high computational requirements. Neural networks are versatile and capable of mapping highly nonlinear relationships. Their effectiveness, however, depends strongly on data availability and quality.

Given the diversity of approaches and their respective advantages and limitations, a systematic comparison of these paradigms under consistent conditions is both relevant and necessary.

4 Problem Setup and Development Framework

4.1 Problem Formulation

Longitudinal Velocity as the Estimation Target

The primary quantity of interest in this work is the vehicle longitudinal velocity, v_x . As defined in (2.2), it corresponds to the projection of the centre-of-mass velocity onto the longitudinal axis of the vehicle reference frame (Fig. 2.1). It is the target variable to be estimated.

All quantities are considered in discrete time. The sampling instants are defined as

$$t_k = kT_s, \quad k \in \mathbb{N}, \quad (4.1)$$

where T_s is the sampling period. The true value of longitudinal velocity at time step k is referred to as the ground-truth longitudinal velocity or simply true longitudinal velocity and is denoted by $v_x(k)$.

The objective of the estimation problem is to compute an estimate $\hat{v}_x(k)$ that approximates $v_x(k)$ at each time step.

General Estimator Structure

At each time step, a set of signals is provided by the vehicle platform. Let $\mathbf{y}_k \in \mathbb{R}^m$ denote the vector of signals available to the estimator at time step k . This vector may include sensor signals as well as additional signals available to the estimator.

The components of $\mathbf{y}_k$ are generally modelled as stochastic processes, although some signals are treated as deterministic and known. Where appropriate, additive noise models are introduced during estimator development (see Section X).

The information set available to the estimator at time k is defined as

$$\mathcal{I}_k = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_k\}. \quad (4.2)$$

The estimation problem can therefore be expressed as a mapping

$$\hat{v}_x(k) = \mathcal{F}(\mathcal{I}_k), \quad (4.3)$$

where $\mathcal{F}(\cdot)$ denotes the estimation algorithm. The mapping (4.3) implies that only information available up to and including time step k may be used to compute $\hat{v}_x(k)$. No future information is permitted to comply with the envisioned real-time application of the estimator.

Different estimation approaches considered in this thesis correspond to different realisations of the mapping $\mathcal{F}$, while operating within the same discrete-time and information framework

defined above.

4.2 Simulation Environment

The reference vehicle and the data used in this work were provided by an industrial partner. Due to confidentiality reasons, the identity of the partner and the specific vehicle model are not disclosed.

The datasets were generated using a Driver-in-the-Loop (DiL) simulator based on a high-fidelity model of a high-performance production vehicle. The simulator enables controlled excitation of the vehicle model while providing direct access to internal states and platform signals. The vehicle model was operated in a passive configuration, meaning that active control systems were disabled. As a result, the vehicle response was governed solely by driver inputs and intrinsic vehicle dynamics, without control interventions. The simulation environment was configured with a constant value of the tyre-road friction coefficient.

Apart from the signals, key reference vehicle parameters were provided, including the wheel-base, total mass, front-rear weight distribution, front track, rear track and static loaded radii. The collected signals include standard on-board sensor measurements (e.g., yaw rate, longitudinal acceleration) as well as additional vehicle-level signals made available by the reference platform.

Under the simulator configuration considered in this work, all logged signals are idealised and noise-free, sampled at a fixed rate of $T_s = 0.01$ s, and perfectly synchronised. Accordingly, the simulated v_x signal is considered the ground truth longitudinal velocity and serves as the reference signal for the evaluation of all estimators developed in this thesis.

Three independent simulation runs were considered. Two runs correspond to high-speed circuit driving scenarios, while the third represents more general driving conditions. Together, these datasets provide diverse driving regimes, including those identified in previous chapters as critical for longitudinal velocity estimation. A detailed characterisation of the dataset properties is provided in the following sections.

Industrial Reference Estimator

Additionally, the industrial partner provided estimation data from an in-house estimator (hereafter referred to as the VDC estimator) which was used as a benchmark. The outputs of this estimator were generated using the same simulation data employed in this study, allowing consistent comparison under identical input conditions.

The reference estimator operates using the following measured inputs:

- Longitudinal and lateral accelerations: a_x, a_y
- Yaw rate: r
- Wheel angular velocities: $\Omega_i, \quad i \in \{\text{FL}, \text{FR}, \text{RL}, \text{RR}\}$

- Steering angle (steering wheel angle or front wheel angle): δ

4.3 Dataset Allocation and Characterisation

4.3.1 Dataset Allocation

To ensure methodological consistency and fairness in the development and evaluation of the different estimators, the available data were appropriately partitioned and allocated.

Three independent datasets were considered:

- GeneralDriving (GD)
- NBR_fastpace (NBR)
- NTC_reverse (NTC)

GD and NBR were used for estimator development, including model tuning and machine learning training and validation, whereas NTC was held out exclusively for final performance assessment. GD and NBR therefore form the development dataset (DDS), while NTC serves as the test dataset (TDS) throughout this work. This allocation prevents overfitting and avoids information leakage between development and test stages.

Dataset	Role
GeneralDriving (GD)	Development
NBR_fastpace (NBR)	Development
NTC_reverse (NTC)	Test

Table 4.1: Dataset allocation for estimator development and evaluation.

4.3.2 Data Characterisation

To support the development process and to contextualise the performance of the different estimators, the provided data were characterised. This provides insight into the overall driving conditions, the operating dynamic domain, and the presence of critical scenarios and manoeuvres relevant to longitudinal velocity estimation. The results of this characterisation were used both to inform design decisions and to interpret estimator behaviour in dynamically demanding operating regions. In the following, the most relevant characteristics of the datasets are presented. The analysis is not intended to be exhaustive, but rather to provide a structured overview of the dataset properties that are most relevant to longitudinal velocity estimation.

Due to confidentiality constraints, selected signals are normalised by the maximum value observed across all datasets and expressed as a percentage of this reference.

In Fig. 4.1, the longitudinal velocity time series across GD, NBR and NTC are shown. Time-series representations, such as those shown in Fig. 4.1, were generated for all relevant measured and derived signals during the development process. These plots provide an intuitive

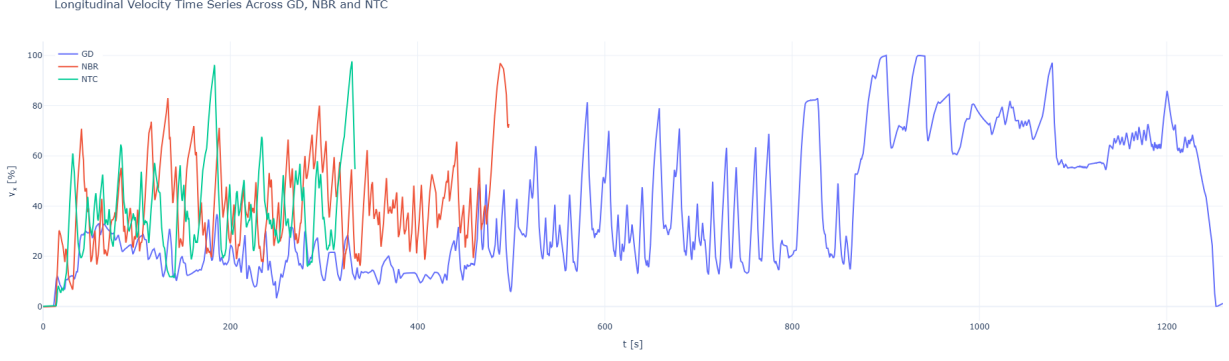


Figure 4.1: Normalised longitudinal velocity time series for the GD, NBR and NTC datasets.

means of inspecting signal behaviour over time and allow specific events or irregularities to be examined in detail when required.

The similarities and differences between the development and test datasets are particularly relevant, as they indicate the extent to which the estimators were required to generalise beyond the conditions observed during development. Figs. 4.2 and 4.3 show a comparison of the empirical v_x distributions on DDS and TDS.

Fig. 4.2 shows normalised histograms for DDS and TDS, with the bin width set to 1%. From this figure, it is evident that the DDS covers the full v_x range of the TDS. However, the distributions differ noticeably: the proportion of samples in the medium-to-high range of the empirical distribution is significantly larger for the test dataset.

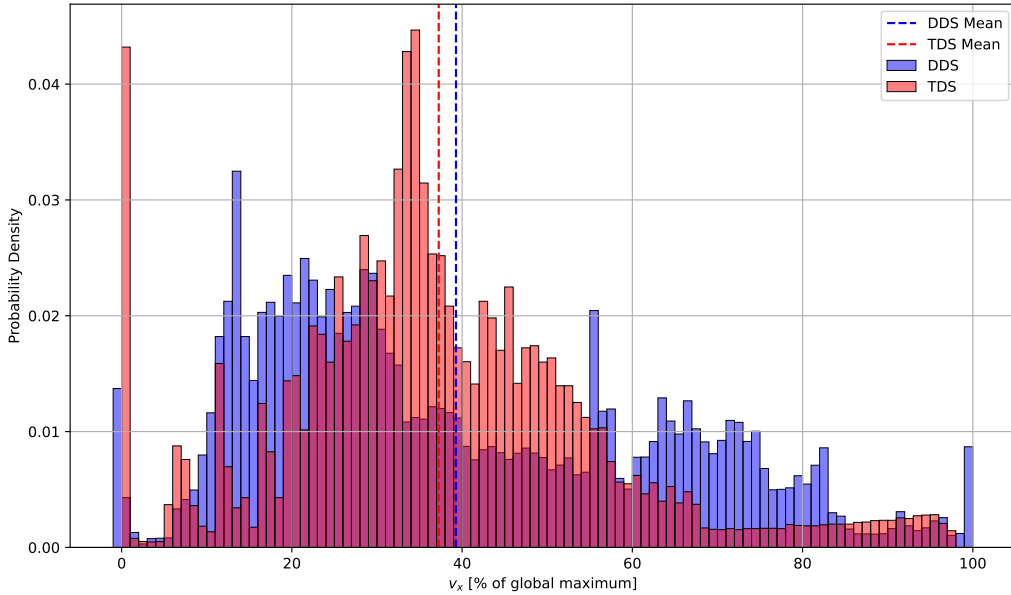


Figure 4.2: Empirical probability density of the normalised longitudinal velocity for DDS and TDS.

Fig. 4.3 shows the empirical cumulative distribution functions (ECDFs) of the normalised longitudinal velocity, together with the percentage value of the Wasserstein-1 distance computed for the two datasets.

The Wasserstein-1 distance (also referred to as the Earth Mover’s Distance, EMD) quantifies

the minimum “work” required to transform one distribution into another. In one dimension, it is defined as

$$W_1(F_1, F_2) = \int_{-\infty}^{\infty} |F_1(x) - F_2(x)| dx, \quad (4.4)$$

where F_1 and F_2 denote the cumulative distribution functions of the two datasets. It therefore corresponds to the area between the two ECDFs and provides a scalar measure of distributional discrepancy between v_x on DDS and TDS.

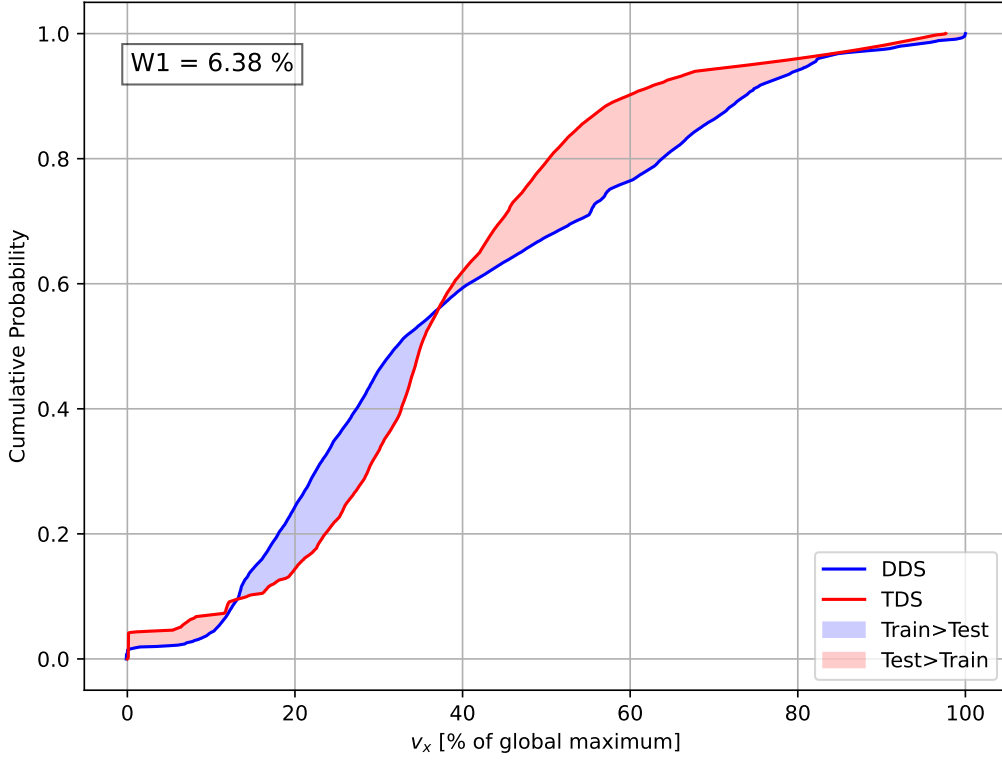


Figure 4.3: ECDFs of the normalised longitudinal velocity for DDS and TDS. The shaded area corresponds to the absolute difference between the distributions, and the resulting Wasserstein-1 distance (W1), expressed in percentage points of the global maximum velocity, is indicated in the figure.

To further analyse the differences in driving dynamics between the two datasets, so-called GG diagrams were considered. These plots represent the joint distribution of longitudinal and lateral accelerations (a_x and a_y) over the driving sessions. Large positive or negative longitudinal accelerations correspond to strong acceleration and braking events, respectively, while high magnitudes of lateral acceleration are associated with cornering manoeuvres. Regions where both $|a_x|$ and $|a_y|$ are simultaneously high indicate combined dynamic loading conditions, such as trail braking.

As discussed in Chapter 2, high-acceleration operating regions are particularly relevant to this project. Under these conditions, tyre forces approach the friction limits, increasing the likelihood of slip, which was identified as one of the main challenges to accurate velocity estimation. As a result, the GG diagrams provide insight into the extent to which the estimators were exposed to dynamically demanding operating regimes during development and testing.

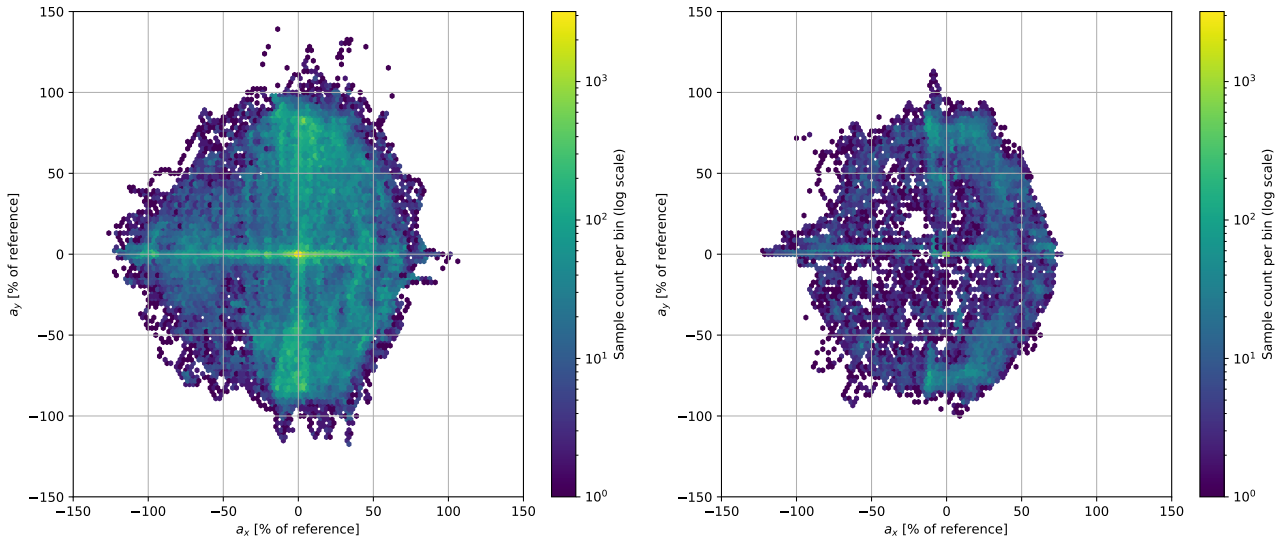


Figure 4.4: GG diagrams for DDS and TDS, shown as hexagonal density maps of the normalised longitudinal and lateral accelerations. The colour scale (logarithmic) represents the number of samples per bin and is shared between the two plots to enable direct visual comparison of the occupied dynamic envelope.

Fig. 4.4 shows that the runs constituting the DDS cover a larger dynamic envelope than the TDS. The extreme values of both longitudinal and lateral acceleration are greater in magnitude (for both positive and negative directions) in the DDS. Moreover, the DDS GG diagram exhibits a more uniformly populated acceleration plane, whereas the TDS plot displays regions with comparatively sparse occupancy, indicating that certain parts of the potential dynamic envelope were not covered during the test runs.

However, some points in the TDS lie outside the range observed in the DDS under combined acceleration conditions, such as those visible in the upper-left region of the TDS GG diagram. This is significant, as it means that the estimators are evaluated in operating conditions that were not encountered during development (tuning and training/validation), and therefore must handle scenarios beyond the dynamic domain on which they were developed.

Another diagnostic tool employed in this work were positional plots, such as the example shown in Fig. 4.5. By visualising signals along the driven trajectory, these plots provide spatial context to the time-series analysis and allow signal behaviour to be interpreted from a driving perspective.

Beyond general consistency checks (e.g., verifying that strong negative longitudinal acceleration coincides with braking zones), positional plots proved particularly useful for diagnosing unexpected estimator behaviour and irregular signal patterns.

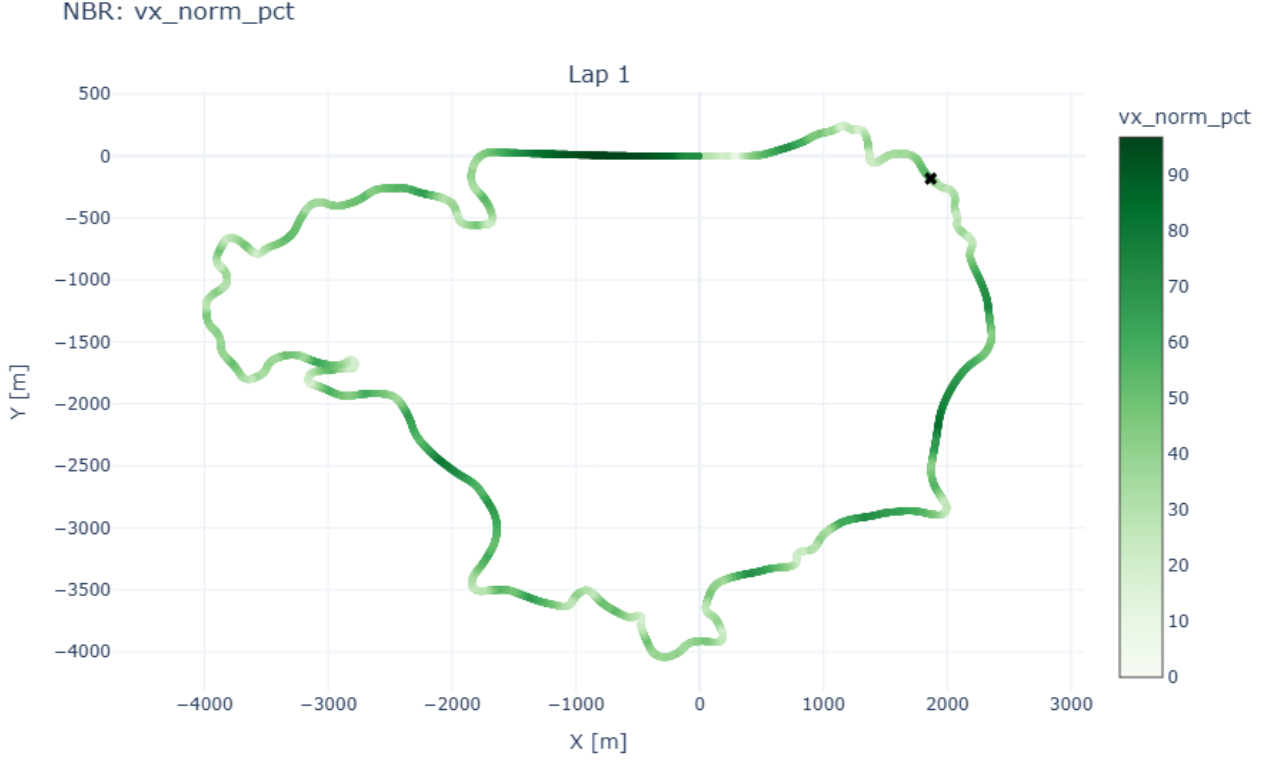


Figure 4.5: Positional plot of the normalised longitudinal velocity for the NBR run. The driven trajectory is shown in the X - Y plane and colour-coded according to v_x , allowing the spatial distribution of vehicle speed to be interpreted in the context of the track layout.

4.4 Development Principles and Constraints

The reference vehicle platform provides a wide range of vehicle platform signals that could in principle be used for velocity estimation. However, not all available signals are equally useful for this purpose. Some signals may have only a weak relationship with the velocity estimation problem, while others may be difficult to measure reliably in real vehicles or may be available only as estimated quantities. For this reason, the development of the estimators in this work was guided by a set of practical principles governing both estimator design and signal selection.

Signal selection was primarily informed by the expected relevance of each signal to the velocity estimation problem and by considerations regarding signal quality in real-world vehicle platforms. While the simulator environment provides access to high-quality signals, it was recognised that certain quantities may not be measured or estimated with the same accuracy in real vehicles. Consequently, preference was given to signals that are commonly available on modern vehicle platforms and that are expected to provide informative measurements for velocity estimation under realistic conditions.

The role of signals in the estimator design process differs between model-based and data-driven approaches. In model-based estimators, the choice of signals is constrained by the structure of the underlying physical model. More detailed models can enable the use of ad-

ditional signals which are informative for longitudinal velocity. However, as discussed in the theoretical development presented in earlier chapters, increasing model complexity introduces additional modelling assumptions and reliance on signals that may be uncertain or difficult to obtain in practice. Estimator design therefore involves a balance between model complexity, signal availability and modelling uncertainty.

Machine learning approaches offer greater flexibility in this respect. Since these methods do not rely on an explicit physical model of the system, the selection of input signals is not constrained by model structure. Instead, signals were selected based on a combination of physical intuition and exploratory data analysis aimed at identifying informative signals for velocity estimation. This included examining relationships between candidate signals and the true longitudinal velocity as well as estimation performance across different signal combinations.

More generally, a practical development strategy was adopted throughout this work. Simple estimator designs were implemented first, with additional complexity introduced when necessary. This approach reflects the exploratory nature of the present thesis, which was not constrained by a predefined estimation framework. Rather than optimising a single estimator within a fixed structure, the objective was to explore different estimation paradigms and progressively develop representative implementations of each. As a result, estimator complexity was introduced when justified by observed performance limitations.

Finally, practical considerations also influenced the overall development framework. In particular, the estimators rely exclusively on on-board vehicle platform signals, while global navigation satellite system (GNSS) measurements were not considered. As discussed in previous chapters, GNSS-based velocity estimation may suffer from latency, signal attenuation and temporary unavailability in complex environments. However, within the considered simulator-based environment all signals are perfectly synchronised and position measurements are always available with perfect accuracy. It was therefore of greater interest to consider the case where GNSS is not available. Additionally, in order to maintain a realistic problem setting, selected measurements were corrupted to emulate sensor imperfections.

Together, these principles and constraints provided a consistent framework guiding the development and evaluation of the estimators presented in the following sections.

4.5 Measurement Corruption

The signals provided by the simulator are noise-free and perfectly synchronised, resulting in unrealistically high measurement accuracy when compared with real vehicle platforms. While no sensor is perfectly accurate in practice, Chapters 2 and 3 highlight that velocity estimation approaches relying on inertial measurements are particularly susceptible to noise-induced performance degradation. This sensitivity arises primarily because such methods rely on the integration of acceleration signals, causing measurement errors to accumulate over time. In addition, automotive-grade inertial measurement units (IMUs) typically exhibit non-negligible noise and bias effects that can significantly influence estimation performance. During the early

stages of estimator development it was also observed that, when using clean inertial signals, even a very simple vehicle model could achieve unrealistically strong performance in certain scenarios without requiring corrective measurements (see Chapter 5). This behaviour did not align with observations commonly reported in the literature. For these reasons, signal corruption was introduced for the IMU signals in order to obtain a more realistic problem setting.

The aim of the corruption process was to introduce sensor imperfections in a realistic manner inspired by the behaviour of real inertial sensors. The objective was not to reproduce the characteristics of any specific IMU model, which would in any case be difficult to achieve using only simulation data and IMU technical datasheets. In practice, the behaviour of a sensor in a vehicle depends not only on its intrinsic noise characteristics but also on installation effects such as mounting conditions, misalignment between sensor and vehicle reference frames, and the influence of the surrounding vehicle environment. Nevertheless, available sensor technical specifications were used to guide the selection of representative noise parameters. Corruption was therefore applied to all IMU signals used in this work, namely the longitudinal acceleration a_x , lateral acceleration a_y , and yaw rate measurements.

The corruption of IMU signals was achieved utilising the MATLAB imuSensor model which computes an inertial measurement unit reading based on an inertial input signal. The model has tunable parameters which correspond to typical specifications reported in IMU datasheets.

In this work, noise density, bias instability and constant bias were considered. The constant bias was added based on the signal offset over temperature specification since it was reasonable to assume this affect would be present in the reference vehicle. In particular, Noise levels were guided by specifications from STMicroelectronics ASM330LHH and Bosch MMP3 sensors. The nominal datasheet noise parameters were conservatively inflated to account for real-world operating conditions discussed in Chapter 2.4.2. The parameters used in the signal corruption are summarised in Table 4.2.

Table 4.2: IMU corruption parameters used for signal corruption

Parameter	Value
Sampling frequency f_s	1.00×10^2 Hz
Gyro noise density	2.10×10^{-3} rad/s/ $\sqrt{\text{Hz}}$
Accel noise density	1.96×10^{-2} m/s ² / $\sqrt{\text{Hz}}$
Gyro bias instability	4.36×10^{-5} rad/s
Accel bias instability	1.20×10^{-3} m/s ²
Gyro constant bias	1.05×10^{-3} rad/s
Accel constant bias	2.94×10^{-2} m/s ²

The resulting corrupted signals provide a more realistic representation of inertial signals that would be available on a real vehicle platform and form the basis for the estimator development presented in the following chapters.

4.6 Comparative Evaluation Framework

The performance of the developed estimators is assessed by comparing the estimated longitudinal velocity with the reference velocity available in the dataset. Let $v_x(k)$ denote the reference longitudinal velocity at time step k , and $\hat{v}_x(k)$ the corresponding estimated value. The estimation error is therefore defined as

$$e(k) = \hat{v}_x(k) - v_x(k). \quad (4.5)$$

Two primary performance metrics are used for quantitative evaluation: the root-mean-square error (RMSE) and the maximum absolute error. For a dataset containing N samples, the RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{k=1}^N e(k)^2}, \quad (4.6)$$

while the maximum absolute error is given by

$$e_{\max} = \max_{k \in \{1, \dots, N\}} |e(k)|. \quad (4.7)$$

These metrics provide complementary information, with RMSE reflecting the overall estimation accuracy and the maximum absolute error capturing the worst-case deviation. The developed estimators are compared both among themselves and against the industrial reference estimator.

In addition to these aggregate metrics, estimator performance is further analysed across different operating conditions. Particular attention is given to identifying situations in which estimation performance degrades and to examining whether different approaches exhibit distinct strengths or weaknesses under specific driving conditions. To this end, the analysis considers estimator behaviour across different dynamic operating regimes and in relation to tyre slip.

5 Model-Based Estimators

5.1 Chapter Overview

This chapter presents the model-based estimators developed in this work. Kalman filtering was selected as the estimation framework since it is reported as the most widely adopted approach for vehicle state estimation in the literature (see Chapter 3). Moreover, Kalman filter algorithms assume a stochastic system formulation, which is well aligned with the present application, where modelling uncertainties and sensor noise must be considered. Both estimators developed in this chapter are therefore based on Kalman filtering.

The first estimator, **KF01**, is a Kalman filter based on a kinematic planar two-track vehicle model. The second estimator, **KF02**, extends this formulation by incorporating wheel dynamics and directly including longitudinal slip in the system model. These estimators are therefore also referred to as the kinematic Kalman filter (KF01) and the dynamic Kalman filter (KF02).

Importantly, these estimators did not emerge as independent designs but resulted from an iterative development process. Following the development principle described in Chapter 4, simple models are explored first and subsequent increases in modelling complexity are motivated by the observed behaviour and performance of the simpler formulations. The modelling progression that led to the final estimator configurations is illustrated in Fig. 5.1. Both estimators were developed exclusively on the development dataset (DDS) defined in Chapter 4.

The remainder of the chapter is organised as follows. Section 5.2 motivates and introduces the vehicle motion model used as the basis for the estimators. Sections 5.3 and 5.4 then present KF01 and KF02 in a structured and parallel manner.

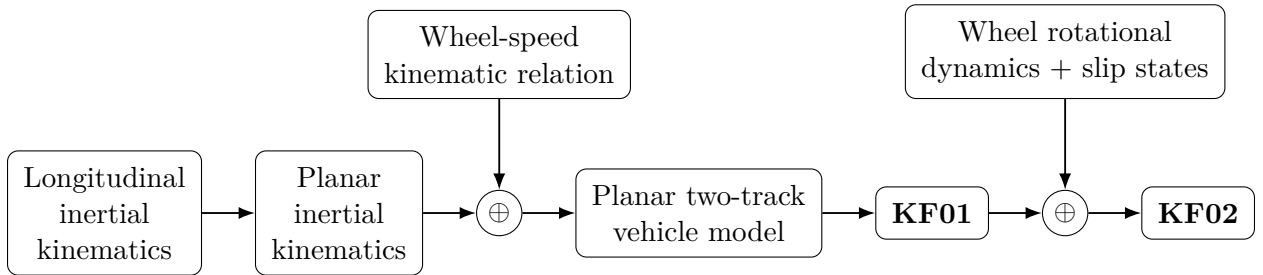


Figure 5.1: Schematic overview of the modelling progression leading to the final Kalman filter estimator structures.

5.2 Baseline Vehicle Model

The formulation of a Kalman filter depends directly on the underlying system model. Consequently, the first step in the development of the model-based estimators was the selection of an appropriate vehicle model. The objective of this stage was therefore to investigate the

capability of different modelling assumptions to capture the aspects of vehicle motion most relevant to longitudinal velocity estimation.

Since the available simulator data provide perfectly noise-free signals, the initial modelling experiments were conducted using these clean signals in order to isolate modelling effects from measurement noise. This allowed the limitations of different modelling approaches to be analysed before introducing sensor imperfections.

As discussed in Chapter 2, longitudinal velocity estimation in production vehicles relies primarily on two sources of information: inertial measurements provided by the IMU and wheel-speed measurements obtained from the wheel encoders. Each of these sensing modalities provides partial information about the vehicle's motion and exhibits characteristic limitations when considered in isolation. The modelling development therefore begins by examining how longitudinal velocity can be reconstructed from these two information sources separately before combining them within a unified vehicle representation. In this section, the modelling steps that lead to the baseline vehicle model used for the development of KF01 and KF02 are described.

5.2.1 Inertial Velocity Reconstruction

From the perspective of inertial sensing, the simplest approach to longitudinal velocity estimation is obtained by considering motion only along the longitudinal axis of the vehicle. Under this assumption, the longitudinal kinematic relation is written as

$$a_x = \dot{v}_x. \quad (5.1)$$

This suggests estimating the longitudinal velocity by integrating the measured longitudinal acceleration over time,

$$v_x(t) = v_x(t_0) + \int_{t_0}^t a_x(\tau) d\tau, \quad (5.2)$$

where t_0 denotes the initial time and $v_x(t_0)$ the corresponding initial condition. The true initial longitudinal velocity was set as the initial condition. Fig. 5.2 shows the resulting longitudinal velocity estimate for the NBR run, obtained by integrating the measured a_x signal and comparing it to the ground-truth longitudinal velocity.

Figure 5.2 shows a significant drift in the reconstructed velocity estimate. This behaviour arises from the accumulation of modelling errors over time as the acceleration signal is integrated.

The formulation is therefore extended from purely longitudinal motion to planar vehicle kinematics, which incorporate the measured yaw rate and introduce coupling between the longitudinal and lateral velocity components. According to planar inertial kinematics (Chapter 2), the time derivatives of the velocity components in the vehicle-fixed reference frame are given by

$$\dot{v}_x = a_x + v_y r, \quad \dot{v}_y = a_y - v_x r, \quad (5.3)$$

where r denotes the yaw rate and v_x, v_y are the longitudinal and lateral velocity components

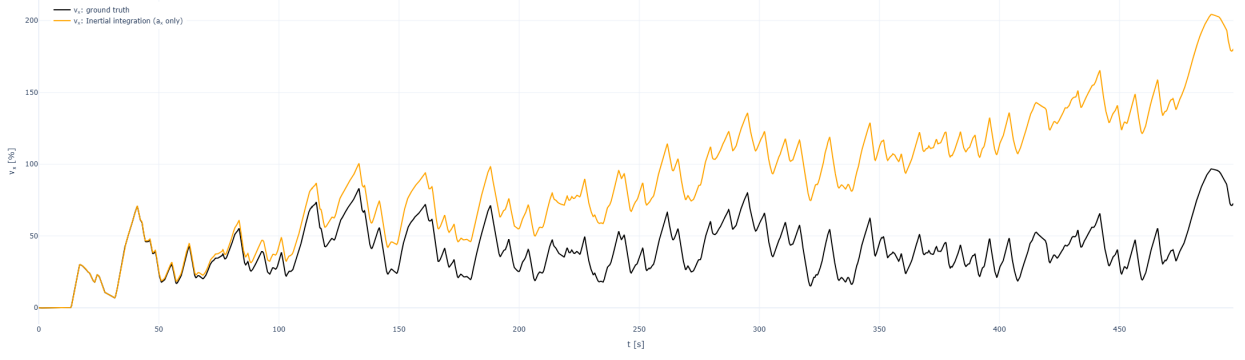


Figure 5.2: Normalised longitudinal velocity for the NBR run. The velocity estimate is obtained by integrating the measured longitudinal acceleration according to (5.2).

at the vehicle centre of mass.

Integrating these relations over time yields the velocity reconstruction equations:

$$v_x(t) = v_x(t_0) + \int_{t_0}^t (a_x(\tau) + v_y(\tau)r(\tau)) d\tau, \quad (5.4)$$

$$v_y(t) = v_y(t_0) + \int_{t_0}^t (a_y(\tau) - v_x(\tau)r(\tau)) d\tau. \quad (5.5)$$

For $v_x(t_0)$, the true initial longitudinal velocity was used. Since no ground truth data for v_y was available, the initial lateral velocity was set to $v_y(t_0) = 0$. As all simulator runs start essentially from standstill, this assumption is considered reasonable and is supported by the obtained performance.

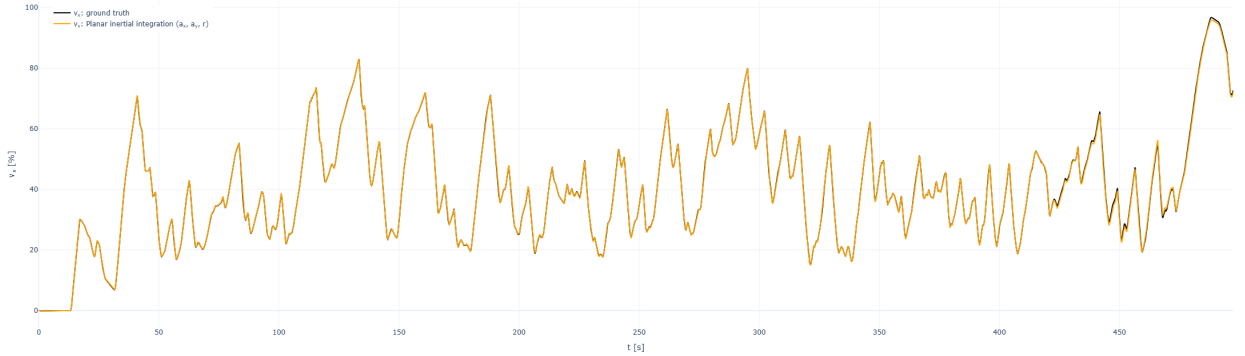


Figure 5.3: Longitudinal velocity reconstruction on the NBR run using planar inertial kinematics obtained by integrating (5.4)–(5.5). The reconstructed velocity is compared to the ground truth and the industrial reference estimator.

Fig. 5.3 shows the resulting time series for the NBR run. Compared with the longitudinal-only integration approach, the inclusion of yaw-rate coupling yields a substantial improvement in the estimated longitudinal velocity. On the NBR run, the achieved accuracy even exceeds that of the industrial reference estimator. While the design and tuning strategy of the industrial reference estimator are not known, it is reasonable to expect that, as an industrial product,

it was developed to operate with real sensor signals subject to noise and imperfections. The observed performance difference should therefore be interpreted with this consideration in mind. Nevertheless, the observation provides a strong indication that planar inertial kinematics offer a sufficiently accurate baseline representation for the use of inertial measurements within the estimator development. Importantly, this improvement is achieved with only a modest increase in modelling complexity, which makes the planar inertial formulation a natural baseline for the estimator designs considered in this work.

5.2.2 Wheel-Speed-Based Velocity Reconstruction

Wheel angular velocity measurements provide information about the vehicle longitudinal velocity through the relation between wheel rotational speed and translational motion. For a single wheel under pure rolling conditions, the longitudinal velocity at the wheel centre can be expressed as

$$v_{wx} = R_e \Omega, \quad (5.6)$$

where R_e denotes the effective rolling radius and Ω the wheel angular velocity. A detailed discussion of this relation and of wheel-speed-based velocity estimation is provided in Chapter 2.2. In practice, however, this relationship can be affected by longitudinal slip, particularly during strong acceleration or braking events.

In order to reduce estimator complexity, some approaches do not use all wheel-speed measurements directly, but instead rely on reduced wheel subsets or derived wheel-speed combinations, such as averaging the speeds of multiple wheels or considering only the wheels on a single axle. For example, Napolitano Dell’Annunziata et al. [18] evaluate velocity estimation strategies using all four wheels, only the rear (driven) wheels, or only the front (non-driven) wheels. Their results show that the best performance is obtained when all wheels are used, while relying solely on the driven wheels yields the poorest accuracy. However, their approach is not directly applicable in the present work, since the vehicle considered here is all-wheel-drive.

Empirical observations from the development experiments performed in this work further support the use of all four wheel-speed measurements. In many situations where slip occurs at one wheel, the remaining wheels still provide useful information about the vehicle motion, whereas wheel-speed-based velocity estimation deteriorates significantly when all four wheels experience substantial slip.

To relate the four wheel angular velocities to the vehicle motion, the geometric relationship between the wheel locations and the vehicle centre of mass must be taken into account. Together with the inertial formulation discussed in the previous subsection, this motivates the introduction of the planar two-track vehicle representation described next. This representation forms the baseline vehicle model used in the development of the estimators KF01 and KF02.

5.2.3 Two-Track Planar Vehicle Representation

The planar two-track vehicle model adopted in this work is illustrated in Fig. 5.4. In this representation, the vehicle is treated as a rigid body with its centre of mass constrained to the plane of the trajectory. The geometric parameters l_f , l_r , w_f , and w_r denote the distances from the centre of mass to the front and rear axles, and the front and rear track widths, respectively. These geometric parameters were provided by the industrial partner.

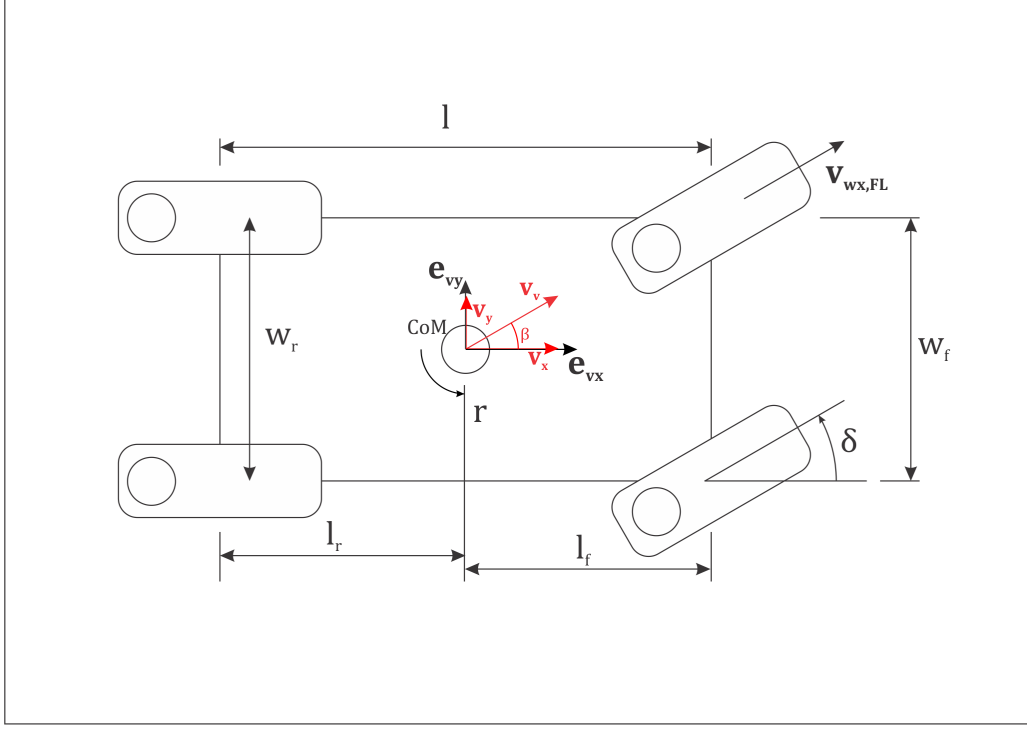


Figure 5.4: Planar two-track vehicle model used as the baseline vehicle model in this work..

This baseline two-track representation is consistent with the inertial kinematic relations introduced in Section 5.2.1 and the wheel-speed relations discussed in Section 5.2.2. Furthermore, it allows the wheel angular velocities to be kinematically related to the motion of the vehicle centre of mass through the velocities at the individual wheel locations.

For the front wheels, accounting for the steering angle δ , the longitudinal velocity components at the wheel centres can be expressed as

$$v_{wx,FL} = \left(v_x - r \frac{w_f}{2} \right) \cos \delta + (v_y + r l_f) \sin \delta, \quad (5.7)$$

$$v_{wx,FR} = \left(v_x + r \frac{w_f}{2} \right) \cos \delta + (v_y + r l_f) \sin \delta. \quad (5.8)$$

For the rear wheels, which are aligned with the vehicle longitudinal axis, the relations reduce to

$$v_{wx,RL} = v_x - r \frac{w_r}{2}, \quad (5.9)$$

$$v_{wx,RR} = v_x + r \frac{w_r}{2}. \quad (5.10)$$

The wheel angular velocities are then obtained through the rolling relation

$$\Omega_i = \frac{v_{wx,i}}{R_{e,i}}, \quad i \in \{FL, FR, RL, RR\}. \quad (5.11)$$

An important assumption underlying these relations is that all wheels satisfy the pure rolling condition at all times.

Together with the inertial kinematic relations

$$\dot{v}_x = a_x + v_y r, \quad (5.12)$$

$$\dot{v}_y = a_y - v_x r. \quad (5.13)$$

these expressions define the mathematical representation of the baseline vehicle model used in this work. This vehicle model is used directly in the formulation of the estimator KF01 and later expanded for the development of KF02.

5.3 KF01 - Kinematic Kalman Filter

5.3.1 Estimator Summary

The KF01 estimator is based on the planar two-track vehicle representation introduced in Section 5.2. Within this framework, inertial measurements and wheel-speed signals provide complementary information about the vehicle velocity. To combine these signals within a Kalman filtering framework, the vehicle kinematics are expressed in state-space form. The resulting estimator formulation is presented below in terms of the signals employed by the estimator, the chosen state representation, and the corresponding process and measurement models.

Table 5.1 summarises the main characteristics of the final estimator configuration, including notes on its expected performance. The underlying design choices are discussed in detail in the following subsections.

Table 5.1: Summary of KF01 – Kinematic Kalman Filter

Item	Description
Estimator type	Model-based
Estimation algorithm	Linear Kalman Filter
Physical model	Planar kinematic two-track vehicle model
Estimated states	v_x, v_y
Estimator inputs	IMU signals (a_x, a_y, r), wheel speeds (Ω_i), steering angle (δ)
Key assumptions	Pure rolling condition, planar vehicle motion
Expected strengths	Robust kinematic sensor fusion, low computational complexity
Expected limitations	Performance degradation under significant wheel slip
$i \in \{FL, FR, RL, RR\}$	

5.3.2 Estimator Signals

The KF01 estimator employs the following signals: longitudinal acceleration a_x , lateral acceleration a_y , yaw rate r , the angular velocities of the four wheels ($\Omega_{FL}, \Omega_{FR}, \Omega_{RL}, \Omega_{RR}$), and the steering wheel angle δ_{SWA} .

The signals a_x , a_y , and r correspond to measurements provided by the inertial measurement unit (IMU). In order to approximate realistic sensing conditions, these signals are corrupted using the measurement noise model described in Chapter 4.

Wheel-speed signals correspond to measurements provided by the wheel-speed encoders and are used directly in the present work without additional corruption.

The steering wheel angle δ_{SWA} is converted to the road wheel steering angle δ through a constant steering ratio i_s according to

$$\delta = \frac{\delta_{\text{SWA}}}{i_s}. \quad (5.14)$$

In practice, the steering ratio of a vehicle is not necessarily constant and may vary depending on the steering system characteristics. Since detailed information about the steering ratio of the reference vehicle was not available, a constant ratio is assumed in this work.

5.3.3 State Vector and Process Model

State Vector

The KF01 estimator represents the planar vehicle velocity through the state vector

$$\mathbf{x} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}.$$

Process Model

The process model of the KF01 estimator is based directly on the inertial kinematic relations given in Eqs. (5.12) - (5.13). For completeness, these relations are restated here:

$$\dot{v}_x = a_x + v_y r, \quad (5.15)$$

$$\dot{v}_y = a_y - v_x r. \quad (5.16)$$

Introducing the state vector defined earlier, the continuous-time process model can be written compactly as

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{w}(t), \quad (5.17)$$

with

$$\mathbf{x}(t) = \begin{bmatrix} v_x(t) \\ v_y(t) \end{bmatrix}, \quad \mathbf{u}(t) = \begin{bmatrix} a_x(t) \\ a_y(t) \end{bmatrix}, \quad (5.18)$$

and

$$\mathbf{A}(t) = \begin{bmatrix} 0 & r(t) \\ -r(t) & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (5.19)$$

In the representation above, the measured longitudinal and lateral accelerations a_x and a_y enter the process model as exogenous inputs through the input vector $\mathbf{u}(t)$, while the yaw rate $r(t)$ appears in the system matrix $\mathbf{A}(t)$ as a time-varying parameter. As a result, the system is linear but time-varying. The term $\mathbf{w}(t)$ represents process noise. In Kalman filtering theory, this term is typically modelled as zero-mean white Gaussian noise. In practical applications, it serves to account for modelling inaccuracies and uncertainty in the signals driving the process model. Its covariance matrix is denoted by $\mathbf{Q}$ and is selected during the filter tuning process.

Discretisation

Since the available signals and the estimator implementation are discrete-time, the continuous-time state-space model must be discretised. The corresponding discrete-time process model is written as

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k. \quad (5.20)$$

Two discretisation approaches were considered in this work. The first is a first-order forward Euler discretisation, which yields

$$\mathbf{A}_k^{\text{Euler}} = \mathbf{I} + T_s \mathbf{A}(t_k) = \begin{bmatrix} 1 & T_s r_k \\ -T_s r_k & 1 \end{bmatrix}, \quad \mathbf{B}_k^{\text{Euler}} = T_s \mathbf{B} = T_s \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (5.21)$$

The second approach is the exact zero-order hold (ZOH) discretisation. Defining

$$\theta_k = r_k T_s, \quad (5.22)$$

the resulting discrete-time model is

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k, \quad (5.23)$$

with

$$\mathbf{A}_k = \exp(\mathbf{A}(t_k) T_s) = \begin{bmatrix} \cos \theta_k & \sin \theta_k \\ -\sin \theta_k & \cos \theta_k \end{bmatrix}, \quad (5.24)$$

and

$$\mathbf{B}_k = \int_0^{T_s} \exp(\mathbf{A}(t_k)\tau) d\tau = \frac{1}{r_k} \begin{bmatrix} \sin \theta_k & 1 - \cos \theta_k \\ \cos \theta_k - 1 & \sin \theta_k \end{bmatrix}. \quad (5.25)$$

For numerical robustness, the discrete-time input matrix can equivalently be written in the half-angle form

$$\mathbf{B}_k = \frac{1}{r_k} \begin{bmatrix} \sin \theta_k & 2 \sin^2\left(\frac{\theta_k}{2}\right) \\ -2 \sin^2\left(\frac{\theta_k}{2}\right) & \sin \theta_k \end{bmatrix}, \quad (5.26)$$

which avoids the subtraction of nearly equal numbers when r_k approaches zero. In the limit $r_k \rightarrow 0$, the exact ZOH matrices reduce to

$$\lim_{r_k \rightarrow 0} \mathbf{A}_k = \mathbf{I}, \quad \lim_{r_k \rightarrow 0} \mathbf{B}_k = T_s \mathbf{I}. \quad (5.27)$$

Both discretisation approaches produced nearly identical estimator behaviour for the sampling time used in this work. Nevertheless, the exact ZOH discretisation was adopted in the final implementation, as it provides a numerically consistent discrete-time representation of the continuous-time process model.

5.3.4 Measurement Model

The measurement model relates the wheel-speed measurements to the estimator states. In discrete time it is written as

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{d}_k + \mathbf{v}_k, \quad (5.28)$$

where $\mathbf{y}_k$ denotes the measurement vector, $\mathbf{H}_k$ the measurement matrix, $\mathbf{d}_k$ a known offset term, and $\mathbf{v}_k$ the measurement noise.

The measurement vector consists of the four wheel angular velocities,

$$\mathbf{y}_k = \begin{bmatrix} \Omega_{FL,k}^m \\ \Omega_{FR,k}^m \\ \Omega_{RL,k}^m \\ \Omega_{RR,k}^m \end{bmatrix}. \quad (5.29)$$

The relations between the wheel angular velocities and the vehicle velocities follow directly from the baseline vehicle model introduced earlier in this chapter. In particular, the wheel-centre longitudinal velocities are given by Eqs. 5.7–5.10, while the wheel angular velocities follow from the rolling relation in Eq. 5.11. Substituting these relations yields a measurement model that is linear in the estimator states.

The resulting measurement matrix is

$$\mathbf{H}_k = \begin{bmatrix} \frac{\cos \delta_k}{R_{e,FL}} & \frac{\sin \delta_k}{R_{e,FL}} \\ \frac{\cos \delta_k}{R_{e,FR}} & \frac{\sin \delta_k}{R_{e,FR}} \\ \frac{1}{R_{e,RL}} & 0 \\ \frac{1}{R_{e,RR}} & 0 \end{bmatrix}, \quad (5.30)$$

while the offset term collecting the known yaw-rate contributions is

$$\mathbf{d}_k = \begin{bmatrix} \frac{-r_k \frac{w_f}{2} \cos \delta_k + r_k l_f \sin \delta_k}{R_{e,FL}} \\ \frac{r_k \frac{w_f}{2} \cos \delta_k + r_k l_f \sin \delta_k}{R_{e,FR}} \\ \frac{-r_k \frac{w_r}{2}}{R_{e,RL}} \\ \frac{r_k \frac{w_r}{2}}{R_{e,RR}} \end{bmatrix}. \quad (5.31)$$

In this formulation the steering angle δ_k and yaw rate r_k enter the measurement equations as known time-varying signals, while the effective rolling radii $R_{e,i}$ and vehicle geometric parameters are treated as constant model parameters and were provided by the industrial partner. It should be noted that the lateral velocity component becomes weakly observable when the steering angle approaches zero. The term $\mathbf{v}_k$ represents measurement noise. In Kalman filtering theory this term is typically modelled as zero-mean white Gaussian noise with covariance matrix $\mathbf{R}$. In the present formulation, where the wheel-speed measurements provided by the simulator are noise-free, this term primarily accounts for modelling inaccuracies associated with the simplified rolling relations used in the measurement model.

5.3.5 Filter Tuning

The covariance matrices were tuned iteratively, with longitudinal velocity RMSE used as the primary performance metric. Innovation and error consistency indicators were also examined as secondary checks during tuning.

Both the process and measurement covariance matrices were assumed diagonal. The process noise covariance was parameterised as

$$\mathbf{Q} = \begin{bmatrix} q_{v_x} & 0 \\ 0 & q_{v_y} \end{bmatrix},$$

while the measurement noise covariance was defined as

$$\mathbf{R} = r_\Omega \mathbf{I},$$

where r_Ω represents the variance associated with the wheel-speed measurements.

The longitudinal process noise variance q_{v_x} was constructed from two contributions. The first accounts for mismatch between the simplified process model and the true vehicle dynamics. To estimate this term, measurement updates were temporarily disabled and the state was propagated using the process model alone. Denoting the resulting prediction error as $e_k =$

$\hat{v}_{x,k} - v_{x,k}^{\text{true}}$, the sample variance of the per-step error increment $\Delta e_k = e_k - e_{k-1}$ was used as an estimate of the model-mismatch contribution to q_{v_x} . This quantity was computed for each run and combined using a sample-count weighted average.

A second contribution accounts for the propagation of accelerometer measurement noise through the velocity prediction step. This effect was approximated using

$$q_{v_x}^{\text{noise}} = T_s^2 \sigma_a^2,$$

where T_s denotes the sampling time and σ_a^2 the accelerometer noise variance. The final value of q_{v_x} was obtained by combining the two contributions.

Since ground truth for the lateral velocity component was not available, the corresponding process noise variance q_{v_y} was set equal to q_{v_x} .

After fixing the process noise covariance, the measurement noise variance r_Ω was tuned manually. The final values of the process and measurement noise variance entries are reported in Table 5.2.

Table 5.2: Final discrete-time covariance values for KF01

Parameter	Value
q_{v_x}	1.555×10^{-3}
q_{v_y}	1.555×10^{-3}
r_Ω	5.865×10^3

5.3.6 Model Algorithm and Expected Behaviour

Given the process and measurement models introduced in the previous subsections, the estimator is implemented using the standard discrete-time Kalman filter recursion. The filter operates sequentially at each sampling instant and was designed with real-time application in mind.

Prediction

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}_k \hat{\mathbf{x}}_{k|k} + \mathbf{B}_k \mathbf{u}_k, \quad (5.32)$$

$$\mathbf{P}_{k+1|k} = \mathbf{A}_k \mathbf{P}_{k|k} \mathbf{A}_k^\top + \mathbf{Q}. \quad (5.33)$$

Update

$$\hat{\mathbf{y}}_{k|k-1} = \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} + \mathbf{d}_k, \quad (5.34)$$

$$\boldsymbol{\nu}_k = \mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}, \quad (5.35)$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}, \quad (5.36)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \mathbf{S}_k^{-1}, \quad (5.37)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \boldsymbol{\nu}_k, \quad (5.38)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}. \quad (5.39)$$

Algorithm 1: KF01 Estimator Algorithm

Input: IMU signals a_x, a_y, r , steering angle δ , wheel speeds Ω_i

Output: Estimated longitudinal velocity $\hat{v}_x$

Initialise prior state estimate $\hat{\mathbf{x}}_{0|-1}$ and covariance $\mathbf{P}_{0|-1}$;

for each sampling instant k do

 Acquire sensor measurements;

 Construct measurement model matrices $\mathbf{H}_k$ and $\mathbf{d}_k$;

Update step;

 Compute innovation;

 Compute innovation covariance;

 Compute Kalman gain;

 Update state estimate $\hat{\mathbf{x}}_{k|k}$ and covariance $\mathbf{P}_{k|k}$;

Output longitudinal velocity estimate $\hat{v}_x$;

Prediction step;

 Construct process matrices $\mathbf{A}_k$ and $\mathbf{B}_k$;

 Predict next prior $\hat{\mathbf{x}}_{k+1|k}$ and covariance $\mathbf{P}_{k+1|k}$;

Expected Behaviour

KF01 is based on a simple kinematic model and standard onboard signals, which makes it straightforward to implement and relatively insensitive to uncertain vehicle parameters. It is therefore expected to provide reliable velocity estimation under conditions where the pure-rolling assumption is approximately satisfied. However, because wheel slip is not explicitly considered in the model, the estimator is expected to degrade during strongly accelerating or braking manoeuvres and in other situations where significant tyre slip occurs. In addition, the lateral velocity component is only weakly observable for small steering angles, which limits the richness of the state reconstruction during near-straight driving. These expectations follow directly from the modelling assumptions adopted for KF01 and are consistent with the theoretical considerations and findings reported in the literature discussed in Chapter 3.

5.4 KF02 - Dynamic Kalman Filter

5.4.1 Estimator Summary

KF01 relies on a kinematic vehicle representation which assumes pure rolling of the wheels; therefore, performance degradation may occur under high-slip conditions. To address this limitation, the baseline vehicle model is extended to incorporate information related to wheel slip. In particular, KF02 introduces wheel rotational dynamics and treats the longitudinal slip ratios of the individual wheels as additional estimator states. This allows the filter to account for deviations from the pure rolling condition while maintaining the overall modelling framework established for KF01.

Table 5.3 summarises the main characteristics of the final estimator configuration, including notes on its expected performance. The underlying design choices are discussed in detail in the following subsections.

Table 5.3: Summary of KF02 – Dynamic Kalman Filter

Item	Description
Estimator type	Model-based
Estimation algorithm	Extended Kalman Filter (EKF)
Physical model	Planar two-track vehicle model with wheel dynamics and slip states
Estimated states	v_x, v_y, ω_i, s_i
Estimator inputs	$a_x, a_y, r, \Omega_i, \delta, T_{d,i}, T_{b,i}, F_{z,i}$
Key assumptions	Planar vehicle motion, tyre force model, stochastic slip dynamics
Expected strengths	Improved robustness under wheel slip conditions
Expected limitations	Increased model complexity and parameter dependence
$i \in \{FL, FR, RL, RR\}$	

5.4.2 Estimator Signals

KF02 employs the following signals: longitudinal acceleration a_x , lateral acceleration a_y , yaw rate r , wheel angular velocities Ω_i , steering angle δ , drivetrain torques $T_{d,i}$, brake torques $T_{b,i}$, and wheel normal loads $F_{z,i}$. The first group of signals ($a_x, a_y, r, \Omega_i, \delta$) is shared with KF01 and was introduced in the previous section.

Drive and brake torques ($T_{d,i}, T_{b,i}$) are available on the simulation platform and on the reference vehicle. However, unlike the signals listed above, these quantities are not standard measurements commonly employed in vehicle state estimation and are typically obtained through drivetrain and braking system models.

The wheel normal loads ($F_{z,i}$) are likewise available on the simulation platform and on the reference vehicle; however, on real vehicles they are generally obtained through estimation based on simplified load transfer models.

While these signals are employed in KF02 to obtain information related to wheel slip, it

should be noted that they may not be available with high accuracy on production vehicles. The formulation adopted here therefore represents a favourable scenario for exploiting wheel-dynamics information within the estimator.

5.4.3 State Vector and Process Model

State Vector

The state vector of KF02 is obtained by augmenting the KF01 state with wheel angular velocities and longitudinal slip ratios. It is defined as

$$\mathbf{x} = \begin{bmatrix} v_x & v_y & \omega_{FL} & \omega_{FR} & \omega_{RL} & \omega_{RR} & s_{FL} & s_{FR} & s_{RL} & s_{RR} \end{bmatrix}^\top. \quad (5.40)$$

Process Model

The process model of KF02 extends that of KF01. The kinematic relations governing v_x and v_y are identical to those introduced for KF01. The additional states are described through wheel rotational dynamics and a stochastic model for slip evolution.

For each wheel $i \in \{FL, FR, RL, RR\}$, the rotational dynamics are written as

$$I_{w,i}\dot{\omega}_i = T_{d,i} - T_{b,i} - R_{e,i}F_{x,i}, \quad (5.41)$$

where $I_{w,i}$ denotes the wheel rotational inertia, $R_{e,i}$ the effective rolling radius, and $F_{x,i}$ the longitudinal tyre force. The wheel rotational inertias $I_{w,i}$ were estimated using a hollow cylinder approximation and the available wheel mass and geometry.

In order to retain a simple process model while capturing the main nonlinear characteristics of longitudinal tyre behaviour, the tyre force is described using the relation

$$F_{x,i} = \mu F_{z,i} \tanh\left(\frac{C_{xN}}{\mu} s_i\right), \quad (5.42)$$

where μ denotes the road friction coefficient provided by the simulator, $F_{z,i}$ the wheel normal load, s_i the longitudinal slip ratio, and C_{xN} a normalised longitudinal stiffness parameter. The value of C_{xN} was selected empirically as described in Section 5.4.5. This tyre force formulation preserves an approximately linear relationship between slip and tyre force for small $|s_i|$, with a slope proportional to the normal load, while introducing saturation at higher slip magnitudes.

The longitudinal slip ratios are modelled as stochastic states. Following modelling choices commonly adopted for uncertain tyre-related quantities in the literature (e.g. Guo et al. [15]), their dynamics are described using a random-walk model

$$\dot{s}_i = w_{s,i}, \quad w_{s,i} \sim \mathcal{N}(0, \sigma_s), \quad (5.43)$$

where $w_{s,i}$ is a zero-mean Gaussian white-noise process representing uncertainty in the slip dynamics. This avoids imposing a deterministic slip dynamics model while allowing the estimator

to reconcile discrepancies between wheel-speed measurements and vehicle kinematics.

Let $\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ denote the continuous-time state derivative assembled from the kinematic relations, wheel rotational dynamics, and slip-state model. Since the estimator is implemented in discrete time and the resulting process model is nonlinear, a first-order forward Euler discretisation is adopted:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + T_s \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k. \quad (5.44)$$

Unlike KF01, the process model of KF02 is nonlinear with respect to the state vector. A standard linear Kalman filter formulation is therefore not directly applicable. In this work the estimator is implemented as an Extended Kalman Filter (EKF), which was selected as a widely adopted and computationally efficient nonlinear extension of the Kalman filter.

The EKF operates by linearising the nonlinear process model locally around the current state estimate. The covariance propagation therefore relies on the Jacobian of the discretised process model,

$$\mathbf{F}_k = \left. \frac{\partial}{\partial \mathbf{x}} (\mathbf{x} + T_s \mathbf{f}(\mathbf{x}, \mathbf{u}_k)) \right|_{\hat{\mathbf{x}}_{k|k}}. \quad (5.45)$$

With the adopted state ordering, the Jacobian matrix can be written in block form as

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{F}_{vv,k} & \mathbf{0}_{2 \times 4} & \mathbf{0}_{2 \times 4} \\ \mathbf{0}_{4 \times 2} & \mathbf{I}_4 & \mathbf{F}_{\omega s,k} \\ \mathbf{0}_{4 \times 2} & \mathbf{0}_{4 \times 4} & \mathbf{I}_4 \end{bmatrix}, \quad (5.46)$$

where

$$\mathbf{F}_{vv,k} = \begin{bmatrix} 1 & T_s r_k \\ -T_s r_k & 1 \end{bmatrix}, \quad (5.47)$$

and

$$\mathbf{F}_{\omega s,k} = \text{diag} \left(-T_s \frac{R_i}{I_{w,i}} \frac{\partial F_{x,i}}{\partial s_i} \right). \quad (5.48)$$

Using the adopted tyre-force model, the derivative appearing in the Jacobian is

$$\frac{\partial F_{x,i}}{\partial s_i} = C_{xN} F_{z,i} \text{sech}^2 \left(\frac{C_{xN}}{\mu} s_i \right). \quad (5.49)$$

This Jacobian is used in the EKF covariance prediction step, while the linearisation of the measurement model is introduced in the following subsection.

5.4.4 Measurement Model

The measurement model of KF02 builds upon the wheel-speed relations introduced for KF01. However, since the longitudinal slip ratios are now included explicitly in the state vector, the

pure rolling condition is no longer imposed. Instead, wheel angular velocity, vehicle kinematics and slip states are related through nonlinear consistency relations.

For each wheel $i \in \{FL, FR, RL, RR\}$, the direct wheel-speed measurement is written as

$$\Omega_i^m = \Omega_i + v_{\Omega,i}, \quad (5.50)$$

where Ω_i^m denotes the measured wheel angular velocity, Ω_i the corresponding wheel angular velocity state, and $v_{\Omega,i}$ the associated measurement noise.

Let $v_{wx,i}$ denote the longitudinal velocity of the i -th wheel centre projected onto the longitudinal axis of the corresponding wheel, as shown in Eqs. (5.7)–(5.10) earlier in the chapter. The slip-aware rolling relation is written as

$$R_{e,i}\Omega_i = v_{wx,i} + s_i\tilde{v}_{wx,i}, \quad (5.51)$$

which holds for $\tilde{v}_{wx,i} > 0$. Here $R_{e,i}$ denotes the effective rolling radius and s_i is the longitudinal slip ratio. The quantity $\tilde{v}_{wx,i}$ represents a regularised longitudinal wheel velocity defined as:

$$\tilde{v}_{wx,i} = \sqrt{v_{wx,i}^2 + \varepsilon_v^2}. \quad (5.52)$$

The parameter $\varepsilon_v > 0$ is introduced to avoid numerical ill-conditioning of the slip definition at very small longitudinal wheel velocities. The chosen value was selected empirically to ensure numerically stable behaviour across the available datasets while preserving the standard slip interpretation for $|v_{wx,i}| \gg \varepsilon_v$.

Using the relation above, the nonlinear consistency function for each wheel is written as

$$g_i(\mathbf{x}) = \Omega_i - \frac{v_{wx,i} + s_i\tilde{v}_{wx,i}}{R_{e,i}}. \quad (5.53)$$

The complete measurement model can therefore be written in the nonlinear state-space form

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \quad (5.54)$$

with measurement vector

$$\mathbf{y}_k = \left[\Omega_{FL}^m \quad \Omega_{FR}^m \quad \Omega_{RL}^m \quad \Omega_{RR}^m \quad 0 \quad 0 \quad 0 \quad 0 \right]^\top, \quad (5.55)$$

and nonlinear measurement function

$$\mathbf{h}(\mathbf{x}_k) = \left[\Omega_{FL} \quad \Omega_{FR} \quad \Omega_{RL} \quad \Omega_{RR} \quad g_{FL}(\mathbf{x}_k) \quad g_{FR}(\mathbf{x}_k) \quad g_{RL}(\mathbf{x}_k) \quad g_{RR}(\mathbf{x}_k) \right]^\top. \quad (5.56)$$

Thus, the first four equations correspond to the direct wheel-speed measurements, while the remaining four enforce consistency between wheel angular velocity, vehicle kinematics and slip.

Because the measurement model is nonlinear, the Extended Kalman Filter requires linearisation around the predicted state estimate. The corresponding measurement Jacobian is

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k|k-1}}. \quad (5.57)$$

With the adopted state ordering, the Jacobian matrix can be written in block form as

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{0}_{4 \times 2} & \mathbf{I}_4 & \mathbf{0}_{4 \times 4} \\ \mathbf{H}_{cv,k} & \mathbf{I}_4 & \mathbf{H}_{cs,k} \end{bmatrix}, \quad (5.58)$$

where

$$\mathbf{H}_{cs,k} = \text{diag} \left(-\frac{\tilde{v}_{wx,i}}{R_{e,i}} \right), \quad (5.59)$$

and

$$\mathbf{H}_{cv,k} = \begin{bmatrix} \frac{\partial g_{FL}}{\partial v_x} & \frac{\partial g_{FL}}{\partial v_y} \\ \frac{\partial g_{FR}}{\partial v_x} & \frac{\partial g_{FR}}{\partial v_y} \\ \frac{\partial g_{RL}}{\partial v_x} & \frac{\partial g_{RL}}{\partial v_y} \\ \frac{\partial g_{RR}}{\partial v_x} & \frac{\partial g_{RR}}{\partial v_y} \end{bmatrix}. \quad (5.60)$$

In Eq. (5.60), the partial derivatives are given by the following expressions:

$$\frac{\partial g_i}{\partial v_x} = -\frac{1}{R_{e,i}} \left(1 + s_i \frac{v_{wx,i}}{\tilde{v}_{wx,i}} \right) \frac{\partial v_{wx,i}}{\partial v_x}, \quad (5.61)$$

$$\frac{\partial g_i}{\partial v_y} = -\frac{1}{R_{e,i}} \left(1 + s_i \frac{v_{wy,i}}{\tilde{v}_{wy,i}} \right) \frac{\partial v_{wy,i}}{\partial v_y}. \quad (5.62)$$

The term $\mathbf{v}_k$ in Eq. (5.54) represents measurement noise, with covariance matrix $\mathbf{R}$. In the present formulation, where the wheel-speed measurements provided by the simulator are noise-free, this term primarily accounts for modelling inaccuracies associated with the simplified rolling relations used in the measurement model.

5.4.5 Filter Tuning

The filter tuning of KF02 follows the same general approach adopted for KF01. In particular, the process noise variances associated with the vehicle velocity states were selected using the same reasoning described in Section 5.3.5 and were therefore kept unchanged.

The process and measurement noise covariance matrices were assumed diagonal. In accordance with the ordering of the estimator states and measurements introduced earlier, their structure is given by

$$\mathbf{Q} = \text{diag} \left(q_{v_x}, q_{v_y}, q_{\Omega}, q_{\Omega}, q_{\Omega}, q_{\Omega}, q_s, q_s, q_s, q_s \right), \quad (5.63)$$

Table 5.4: Final parameters for KF02

Parameter	Value
C_{xN}	45
q_{v_x}	1.555×10^{-3}
q_{v_y}	1.555×10^{-3}
q_Ω	25
q_s	2.87×10^{-5}
r_Ω	1×10^{-4}
r_{c0}	9

$$R = \text{diag}(r_\Omega, r_\Omega, r_\Omega, r_\Omega, r_{c0}, r_{c0}, r_{c0}, r_{c0}). \quad (5.64)$$

The slip process noise variance q_s was derived from the variance of the per-timestep increment of a longitudinal slip proxy signal computed on the development dataset, consistent with the random-walk slip model adopted in the estimator.

The remaining covariance parameters, including the wheel angular velocity process noise variance q_Ω , the wheel-speed measurement noise variance r_Ω , and the consistency function noise variance r_{c0} , were selected through iterative manual tuning on the development dataset. The primary tuning criterion was the validation RMSE of the longitudinal velocity estimate, evaluated on the two development circuits described in Chapter 4. When conflicting trends were observed between the two circuits, preference was given to the NBR dataset, which, being a fast pace run on a race circuit, more closely resembles the operating conditions of the test circuit.

In addition to estimation accuracy, qualitative diagnostics were also monitored during the tuning process. In particular, the residuals of the slip consistency constraint defined in Eq. (5.53) and the resulting slip-state estimates were inspected to ensure that the filter behaviour remained physically plausible.

In addition to the covariance parameters, the normalised longitudinal tyre stiffness C_{xN} , introduced in Eq. (5.42), also had to be determined empirically. This parameter was selected using the same tuning procedure described above. Even with a high assumed variance of the wheel-speed states, C_{xN} had a significant impact on the observed estimation performance.

While the RMSE behaviour was the primary criterion which determined the selected configuration, the estimated slip states provide additional qualitative support for this choice. Fig. 5.5 compares the rear-wheel slip ratios estimated by the filter with slip values computed from the clean simulator signals using Eq. (2.6). Since no measurements of lateral velocity v_y were available in the dataset, only the rear-wheel slips are shown. The comparison is presented for the NBR dataset, which was favoured during the tuning process when divergent trends were observed between the development circuits. The estimated slip states capture the main trends of the reference slip signals, indicating that the selected parameter configuration produces physically reasonable slip dynamics.

The final tuned parameters for KF02 are listed in Table 5.4.

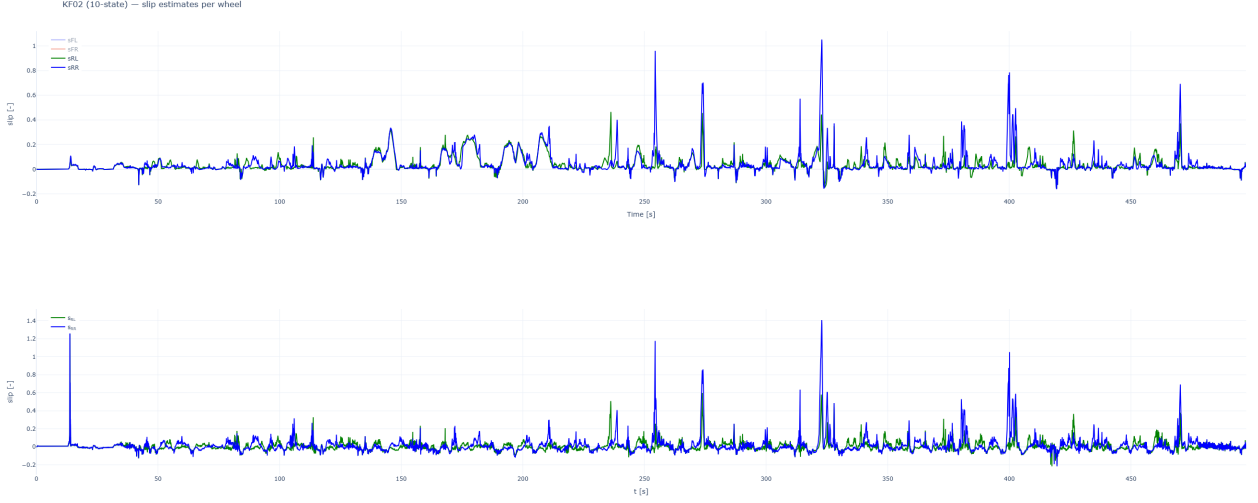


Figure 5.5: Longitudinal slip ratios used for qualitative validation of KF02. KF02 slip state estimates (top) and reference slip computed from clean simulator signals using Eq. (2.6) (bottom).

5.4.6 Model Algorithm and Expected Behaviour

Given the nonlinear process and measurement models introduced in the previous subsections, the estimator is implemented using the standard discrete-time Extended Kalman Filter (EKF) recursion. As in KF01, the filter operates sequentially at each sampling instant and is formulated with real-time implementation in mind.

The EKF follows the same prediction and update structure as the linear Kalman filter presented for KF01. However, since the process and measurement models are nonlinear, the system is linearised at each time step around the current state estimate. This linearisation is performed using the Jacobian matrices derived in the previous subsections.

Prediction

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k), \quad (5.65)$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^\top + \mathbf{Q}, \quad (5.66)$$

where $\mathbf{f}(\cdot)$ denotes the nonlinear process model and $\mathbf{F}_k$ is the Jacobian of the process model evaluated at the current state estimate.

Update

$$\hat{\mathbf{y}}_{k|k-1} = \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}), \quad (5.67)$$

$$\boldsymbol{\nu}_k = \mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}, \quad (5.68)$$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}, \quad (5.69)$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \mathbf{S}_k^{-1}, \quad (5.70)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \boldsymbol{\nu}_k, \quad (5.71)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}. \quad (5.72)$$

Algorithm 2: KF02 Estimator Algorithm

Input: IMU signals a_x, a_y, r , steering angle δ , wheel speeds Ω_i , drive/brake torques T_i , normal loads $F_{z,i}$

Output: Estimated longitudinal velocity $\hat{v}_x$

Initialise prior state estimate $\hat{\mathbf{x}}_{0|-1}$ and covariance $\mathbf{P}_{0|-1}$;

for each sampling instant k **do**

 Acquire sensor measurements;

 Construct nonlinear measurement model and evaluate Jacobian $\mathbf{H}_k$;

Update step;

 Compute innovation;

 Compute innovation covariance;

 Compute Kalman gain;

 Update state estimate $\hat{\mathbf{x}}_{k|k}$ and covariance $\mathbf{P}_{k|k}$;

Output longitudinal velocity estimate $\hat{v}_x$;

Prediction step;

 Propagate nonlinear process model $\mathbf{f}(\cdot)$;

 Evaluate process model Jacobian $\mathbf{F}_k$;

 Predict next prior $\hat{\mathbf{x}}_{k+1|k}$ and covariance $\mathbf{P}_{k+1|k}$;

Expected Behaviour

KF02 extends the kinematic formulation of KF01 by introducing wheel rotational dynamics and longitudinal slip states. By explicitly modelling slip-related effects, the estimator is expected to provide improved velocity estimation during strongly accelerating or braking manoeuvres where the pure-rolling assumption becomes violated.

However, this improvement comes at the cost of increased model complexity and a larger number of parameters. The estimator relies on additional modelling assumptions, including simplified tyre-force relations and approximate drivetrain information. As a result, the performance of KF02 may become more sensitive to modelling inaccuracies and parameter tuning.

In operating conditions where the pure-rolling assumption is approximately satisfied, the additional model complexity may therefore not necessarily lead to improved estimation performance.

These expectations follow directly from the modelling assumptions adopted for KF02 and will be examined experimentally in the evaluation presented in Chapter 8.

6 Data-Driven Estimators

6.1 Chapter Overview

This chapter presents the data-driven and hybrid estimators developed in this work. In contrast to the model-based approaches introduced in Chapter 5, these estimators learn the relationship between available vehicle signals and the longitudinal velocity directly from data using neural network models.

Two neural network architectures are investigated. The first estimator is based on a multi-layer perceptron (MLP), which learns an instantaneous mapping between the input signals and the longitudinal velocity. The second estimator employs a long short-term memory (LSTM) network, which additionally incorporates temporal information through sequential inputs.

The estimators presented in this chapter were selected from a range of candidate configurations explored during the development process. While the two estimators rely on different network architectures, a consistent learning framework was defined in order to isolate the most relevant design aspects within the context of this study. The overall development process leading to the final estimator structures is illustrated in Fig. 6.1.

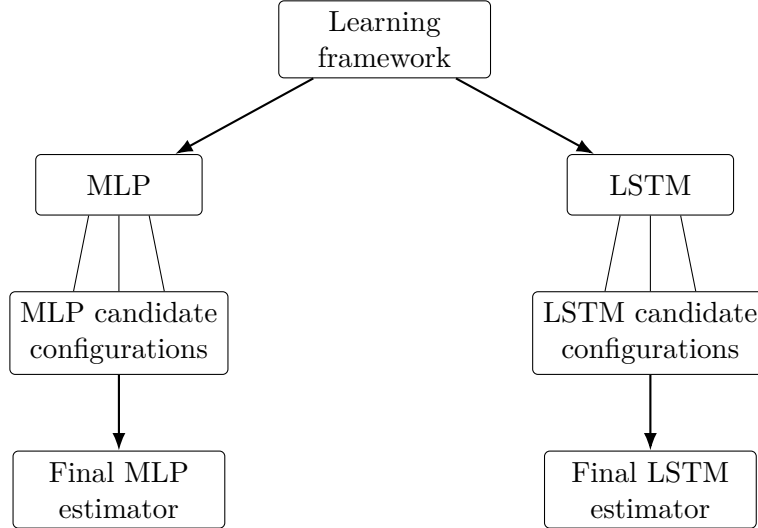


Figure 6.1: Schematic overview of the development process leading to the final data-driven estimators.

The expected strengths and limitations reported for each estimator are derived from theoretical considerations and from the design choices made during development. These statements describe the anticipated behaviour of the estimators and serve to guide the interpretation of the results. The experimental analysis presented in Chapter 8 focuses on aspects of estimator performance that are of direct interest to the present study, rather than attempting to explicitly verify every expected property listed for the estimators.

6.2 Learning Framework

This section introduces the learning framework used for the development of the data-driven estimators. The design choices described here are common to both neural estimators and are defined in advance in order to ensure a consistent basis for comparison.

6.2.1 Available Signals

While the selection of input signals for the model-based estimators (MBEs) is inherently linked to the underlying system modelling and therefore largely determined by the structure of the vehicle model, data-driven estimators (DDEs) do not face this constraint.

During early exploratory development, a range of input combinations was investigated. However, in order to provide a consistent comparison framework, the estimators presented in this work were ultimately developed using the same sensing signals available to the MBEs. This choice allows the analysis to focus on the representational capabilities of model-based and learning-based architectures when provided with the same information.

Accordingly, two signal sets were considered, corresponding to the signals used by the KF01 and KF02 estimators respectively:

- **Feature set A (FSA):** $a_x, ; a_y, ; r, ; \omega_i, ; \delta_{SWA}$
- **Feature set B (FSB):** $a_x, ; a_y, ; r, ; \omega_i, ; \delta_{SWA}, ; F_{z,i}, ; T_i$

where $i \in \{FL, FR, RL, RR\}$ denotes the wheel index and T_i represents the net torque acting on each wheel.

It is important to note that although Feature Set B lists the net wheel torque T_i as an estimator signal, this quantity is not directly provided by the vehicle platform. Instead, the available signals are the drive torque $T_{d,i}$ and the braking torque $T_{b,i}$ applied at each wheel. The net torque T_i is therefore computed as the algebraic sum of these two signals at each time step. This choice reduces the dimensionality of Feature Set B from 20 to 16 inputs, limiting the potential influence of input dimensionality on the comparison between feature sets. Moreover, the net wheel torque represents a physically meaningful quantity governing wheel rotational dynamics.

With that said, it should be emphasised that Feature Set B, and thus the estimators utilising it, effectively incorporate information originating from 20 signals rather than 16.

All signals were standardised prior to training using z-score standardisation. This operation normalises each feature by removing its mean and scaling it by its standard deviation, according to

$$z_j = \frac{x_j - \mu_j}{\sigma_j} \quad (6.1)$$

where x_j denotes the j -th estimator signal, μ_j and σ_j represent the mean and standard deviation of that signal computed over the training set, and z_j is the resulting scaled feature.

This step is particularly important in the present work since the signals originate from quantities expressed in different physical units and numerical ranges. Standardisation therefore prevents signals with larger numerical magnitudes from dominating the optimisation process and ensures that all inputs contribute comparably during training. It is important to note that the parameters of the standard scaler were fitted exclusively on the training data. The same transformation was then applied to the validation set in order to prevent information leakage between the two datasets.

6.2.2 Training and Validation Data Split

Since the development of all estimators in this work was restricted to the Development Dataset (DDS), defined in Section 4.3, the training and validation of the data-driven estimators (DDEs) was also confined to this domain. Consequently, a further data split had to be defined within the DDS.

The training and validation sets were defined as follows:

- **GD dataset:** the entire dataset was included in the training set.
- **NBR dataset:** the first 70% of the run was used for training, while the final 30% of the time series was used as the validation set.

Since the LSTM architecture incorporates temporal information, it was important to preserve the chronological ordering of the data when defining this split. Using a contiguous split of the NBR run ensures that the validation set represents a temporally consistent segment of the driving sequence rather than randomly sampled points.

The resulting allocation of the datasets is summarised in Table 6.1, while Table 6.2 reports the corresponding sample counts and their relative proportions within the DDS.

Table 6.1: Allocation of datasets between training and validation sets.

Dataset	Training Set	Validation Set
GD	Entire dataset	—
NBR	First 70% of run	Final 30% of run

Table 6.2: Number of samples used for training and validation within the Development Dataset.

Set	Number of Samples	Share of DDS
Training Set	160 812	91.5%
Validation Set	14 921	8.5%

6.2.3 Loss Function

The mean squared error (MSE) between the predicted and ground truth longitudinal velocity was used as the training loss function, consistent with its widespread adoption in regression

tasks in supervised learning. MSE is a smooth and differentiable function whose gradient varies continuously with the prediction error, enabling stable optimisation during training. Furthermore, the squared penalty weights large errors more heavily than small ones, which is appropriate in this context where large estimation errors may occur during challenging driving scenarios and are considered particularly costly.

The loss function used in this work is defined as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{k=1}^N (\hat{v}_{x,k} - v_{x,k})^2, \quad (6.2)$$

where $\hat{v}_{x,k}$ denotes the predicted longitudinal velocity at sample k , $v_{x,k}$ is the corresponding ground truth value, and N represents the number of samples used in the loss computation.

6.2.4 Hyperparameter Selection

During early exploratory development, sweeps were performed over a relatively large hyperparameter space. For many of the investigated parameters, the observed variation in estimator performance was limited. These parameters were therefore fixed in the final experiments in order to provide a consistent comparison environment and to focus the analysis on the architectural and feature-design choices that are of primary interest in this work.

In particular, the hyperparameters shared by both data-driven estimators were set as follows:

- Learning rate: 10^{-3}
- Batch size: 256
- Number of training epochs: 200
- Early stopping patience: 20 epochs without improvement in validation MSE
- Minimum learning rate ($\eta_{\min}$): 5×10^{-5}

Furthermore, both neural architectures employed the Adam optimisation algorithm together with a cosine annealing learning-rate scheduler. These choices correspond to configurations that showed stable convergence behaviour across early experimental runs for both the MLP and LSTM estimators.

6.3 MLP Estimator

6.3.1 Estimator Summary

The first data-driven estimator investigated in this work is based on a multi-layer perceptron (MLP). Table 6.3 summarises the main characteristics of the final estimator configuration, including notes on its expected performance. The underlying design choices are discussed in detail in the following subsections.

Table 6.3: Summary of the MLP estimator

Item	Description
Estimator type	Data-driven
Estimation algorithm	Multi-layer perceptron (MLP) regression
Physical model	None (data-driven mapping)
Estimated states	v_x
Estimator inputs	$a_x, a_y, r, \Omega_i, \delta, T_{d,i}, T_{b,i}, F_{z,i}$
Key assumptions	Instantaneous mapping between input features and v_x ; no temporal dependence assumed.
Expected strengths	Ability to capture nonlinear relationships between input signals and v_x across a range of operating conditions.
Expected limitations	Performance degradation under rapid transients due to absence of temporal context.
$i \in \{FL, FR, RL, RR\}$	

Given the fixed hyperparameter configuration described in Section 6.2.4, the development of the MLP estimator focused primarily on the choice of network architecture and input feature set. While other aspects of the estimator could be further investigated, these two design variables are of particular interest in the context of the present comparative study, as they directly influence the representational capacity of the model and the extent to which additional sensing information can be exploited.

Accordingly, the following sections present a comparison of two MLP architectures evaluated on the two estimator signal sets (FSA and FSB) introduced in Section 6.2.1. Based on this comparison, a single architecture and feature set are selected to represent the final MLP estimator considered in the remainder of this work.

6.3.2 Input Representation

The MLP estimator operates on instantaneous vectors of estimator signals. At each sampling instant, the network receives a vector containing the current measurements and produces a direct estimate of the longitudinal velocity v_x . Consequently, no explicit temporal context is provided to the model and the estimator relies solely on instantaneous relationships between the input signals and the target variable.

Two feature sets were considered, corresponding to the estimator signal sets introduced in Section 6.2.1.

Feature Set A (FSA):

$$[a_x, a_y, r, \Omega_i, \delta_{\text{SWA}}], \quad i \in \{FL, FR, RL, RR\}$$

Feature Set B (FSB):

$$[a_x, a_y, r, \Omega_i, \delta_{\text{SWA}}, F_{z,i}, T_i], \quad i \in \{FL, FR, RL, RR\}$$

where T_i denotes the net wheel torque applied at wheel i , computed as $T_i = T_{d,i} + T_{b,i}$, as discussed in Section 6.2.1.

6.3.3 Architecture and Feature Set Selection

Two MLP architectures are evaluated on Feature Set A (FSA) and Feature Set B (FSB) in order to determine the final MLP estimator configuration.

In a production deployment, the complexity of the network architecture would typically be constrained by the computational resources available on the target electronic control unit (ECU). Since hardware deployment targets were outside the scope of the present study, preliminary experiments were conducted across a range of architectures in order to identify suitable model sizes. The final architectures used for comparison were selected based on the consistency of their performance across both feature sets and their alignment with architectures commonly used in comparable vehicle state estimation studies in the literature [20, 18].

The MLP estimators considered in this work consist of a sequence of fully connected layers interleaved with nonlinear activation functions (see Section 2.5.2). The two candidate architectures are defined by their hidden-layer dimensions and are referred to as:

- **Small architecture:** [128, 64]
- **Large architecture:** [256, 128, 64]

In both cases, rectified linear unit (ReLU) activation functions are used after each hidden layer.

Within the defined learning framework, the performance of these two architectures was evaluated on both Feature Set A and Feature Set B. The resulting comparison is summarised in Table 6.4, which reports the estimation performance on the validation set obtained for each architecture–feature set combination.

Model selection was primarily based on the root mean squared error (RMSE) on the validation set, which is consistent with the mean squared error loss function used during training. Network complexity was also considered, with preference given to simpler architectures when comparable RMSE values were obtained. The maximum absolute error (MaxAE) was examined as a secondary metric to assess worst-case estimation errors.

Table 6.4: Validation-set performance of the considered MLP configurations.

Metric	Small + FSA	Small + FSB	Large + FSA	Large + FSB
RMSE	1.096	0.890	1.182	0.867
MaxAE	16.753	11.794	17.344	13.558

Following the results presented in Table 6.4, the small architecture operating on Feature Set B is selected as the final MLP configuration. While this configuration does not achieve the lowest RMSE, the difference in performance was considered too small to justify the increased

complexity of the larger architecture. Furthermore, the selected configuration yields the lowest MaxAE, which further supports this choice.

6.3.4 Estimator Algorithm and Expected Behaviour

The selected MLP estimator operates according to Algorithm 3. At each sampling instant, the available estimator signals are assembled into an input feature vector, standardised using the scaling parameters fitted during training, and passed through the trained neural network to obtain the longitudinal velocity estimate.

Algorithm 3: MLP Estimator Algorithm

Input: Input signals $a_x, a_y, r, \Omega_i, \delta_{\text{SWA}}, F_{z,i}, T_{d,i}, T_{b,i}$

Output: Estimated longitudinal velocity $\hat{v}_x$

Load trained network parameters and signal-scaling parameters;

for *each sampling instant* k **do**

Acquire input signals;

Compute net wheel torque $T_i = T_{d,i} + T_{b,i}$;

Construct input feature vector;

Standardise input signals using the fitted scaling parameters;

Perform forward pass through trained MLP;

Output longitudinal velocity estimate $\hat{v}_x$;

Expected Behaviour

The MLP estimator is expected to perform well when the relationship between the available estimator signals and v_x can be captured through an instantaneous nonlinear mapping. In particular, it is expected to benefit from the ability of neural networks to approximate complex nonlinear dependencies that are difficult to represent explicitly in simple model-based formulations.

At the same time, the absence of explicit temporal context is expected to limit robustness in situations where the recent evolution of the signals carries important information, such as during rapid transients. Therefore, while the MLP may achieve accurate estimation under a broad range of conditions, some degradation may be expected relative to recurrent architectures in scenarios where temporal dynamics are especially informative.

A further limitation arises from the data-driven nature of the estimator. Since the MLP is trained exclusively on the Development Dataset (DDS), performance may degrade when operating conditions deviate significantly from the training data distribution.

6.4 LSTM Estimator

6.4.1 Estimator Summary

The second data-driven estimator investigated in this work is based on a Long Short-Term Memory (LSTM) neural network. Table 6.5 summarises the main characteristics of the final estimator configuration, including the estimator type, employed feature set, and key assumptions, together with notes on the expected behaviour of the estimator.

Table 6.5: Summary of the LSTM estimator

Item	Description
Estimator type	Data-driven
Estimation algorithm	Long Short-Term Memory (LSTM) neural network regression
Physical model	None (data-driven mapping)
Estimated states	v_x
Estimator inputs	$a_x, a_y, r, \Omega_i, \delta, T_{d,i}, T_{b,i}, F_{z,i}$
Key assumptions	Temporal sequences of estimator signals contain sufficient information to improve v_x estimation. Relationships learned from the development dataset remain representative of the operating conditions encountered during estimation.
Expected strengths	Improved v_x estimation during transient driving conditions.
Expected limitations	Estimation performance may degrade when operating conditions differ significantly from those encountered during training.
$i \in \{FL, FR, RL, RR\}$	

In contrast to the MLP estimator presented in Section 6.3, the LSTM processes short temporal sequences of estimator signals rather than instantaneous input vectors. This allows the estimator to exploit temporal relationships between successive observations when inferring the longitudinal velocity v_x .

The design choices leading to the final estimator configuration are discussed in detail in the following subsections.

6.4.2 Input Representation

The LSTM estimator uses the same estimator signals defined for the MLP estimator in Section 6.3.2, and therefore relies on the same feature sets introduced in Section 6.2.1. The same considerations regarding signal availability, physical interpretation, and preprocessing therefore apply.

In contrast to the MLP, which operates on instantaneous input vectors, the LSTM processes short temporal sequences of these signals. At each sampling instant, the estimator receives a sequence of recent measurements and produces an estimate of the longitudinal velocity v_x .

This formulation allows the network to exploit temporal relationships between successive observations, thereby providing the estimator with a limited temporal context when inferring the vehicle velocity.

6.4.3 Architecture and Feature Set Selection

The adopted architecture consists of a single LSTM layer with 128 hidden units, followed by a fully connected linear layer mapping the final hidden representation to the scalar output $\hat{v}_x$.

The selected architecture is consistent with recurrent network configurations commonly adopted in vehicle state estimation studies. In particular, similar single-layer recurrent architectures with 128 hidden units have been reported in works such as Kong et al. [20] and Xiao et al. [11]. During the exploratory development phase, this configuration was also found to exhibit stable convergence behaviour while maintaining a moderate model size.

A stateless windowed formulation was adopted, in which the hidden state is initialised at the beginning of each input sequence. This avoids dependencies between consecutive sequences during training and deployment and limits the temporal context available to the model to the chosen sequence length, reducing the risk of learning long-term patterns specific to the training runs that may not generalise to unseen data.

With the network architecture fixed, the remaining design choice concerned the sequence length. The sequence length, denoted by W , defines the number of past timesteps provided to the LSTM at each prediction. To select an appropriate value, three sequence lengths were considered:

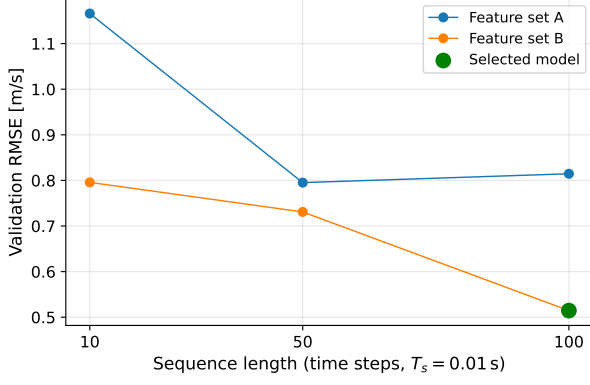
- **Sequence length 10:** 0.1 s of temporal context
- **Sequence length 50:** 0.5 s of temporal context
- **Sequence length 100:** 1.0 s of temporal context

These values correspond to the temporal duration of the input sequences at the adopted sampling rate of 100 Hz.

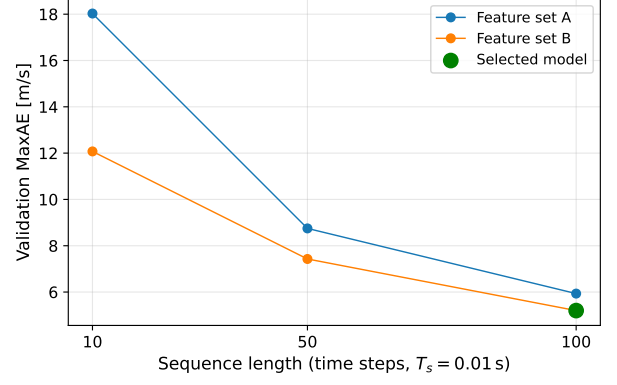
Within the defined learning framework, the performance of the candidate sequence lengths was evaluated using both Feature Set A (FSA) and Feature Set B (FSB). The resulting comparison is summarised in Fig. 6.2, which reports the validation-set estimation performance obtained for each sequence-length–feature-set combination.

Model selection was primarily based on the root mean squared error (RMSE). The computational implications of longer sequence lengths were also considered, with preference given to shorter sequences when comparable RMSE values were obtained. The maximum absolute error (MaxAE) was examined as a secondary metric to assess worst-case estimation errors.

The results show that the FSB configurations outperform the corresponding FSA configurations in all considered cases. Furthermore, increasing the available temporal context consistently improves performance within the explored selection space, suggesting that additional temporal information is beneficial for v_x estimation.



(a) Validation RMSE for different sequence lengths.



(b) Maximum absolute error (MaxAE) for different sequence lengths.

Figure 6.2: Validation performance of the LSTM estimator as a function of sequence length for the two feature sets.

The only exception is the FSA configuration with a sequence length of 50 timesteps, which marginally outperforms the FSA configuration with 100 timesteps in RMSE. However, it is worth noting that while all other runs converged within approximately 14–45 epochs, the FSA/50 configuration required 133 epochs to complete the training loop. This suggests that the observed result may reflect a more favourable optimisation trajectory rather than a genuine advantage of the shorter sequence length.

Based on the obtained results, the configuration using Feature Set B with a sequence length of 100 is selected as the final LSTM-based estimator configuration.

6.4.4 Estimator Algorithm and Expected Behaviour

Since the LSTM requires a complete input sequence of length W before producing the first prediction, the input stream is padded with $W - 1$ initial feature vectors. The signals a_x , a_y , r , Ω_i , δ_{SWA} , $T_{d,i}$, and $T_{b,i}$ are initialised to zero. The normal loads $F_{z,i}$ are initialised using the static vehicle mass and the nominal front–rear weight distribution, assuming equal left–right load distribution. This initialisation is consistent with the experimental runs used in the dataset, where the vehicle begins each run from rest.

Algorithm 4: LSTM Estimator Algorithm

Input: Input signals $a_x, a_y, r, \Omega_i, \delta_{\text{SWA}}, F_{z,i}, T_{d,i}, T_{b,i}$

Output: Estimated longitudinal velocity $\hat{v}_x$

Load trained network parameters and signal-scaling parameters;

Initialise input buffer with $W - 1$ padded feature vectors;

for *each sampling instant* k **do**

 Acquire input signals;

 Compute net wheel torque $T_i = T_{d,i} + T_{b,i}$;

 Construct current feature vector;

 Update buffer: append current feature vector and discard oldest entry, maintaining a sliding window of W timesteps;

 Standardise input window using the fitted scaling parameters;

 Perform forward pass through trained LSTM and FC output layer;

Output longitudinal velocity estimate $\hat{v}_{x,k}$;

Expected Behaviour

The LSTM estimator is expected to outperform the MLP estimator since each prediction is based on a sequence of past observations rather than on instantaneous input vectors. As a result, the estimator inherently exploits a larger amount of information when producing the velocity estimate v_x . Furthermore, this information is provided in a temporally ordered form, which is expected to be particularly beneficial during fast transient driving conditions, where the temporal structure of the signals may help stabilise the velocity estimate and reduce short-term fluctuations.

As with other purely data-driven approaches, the estimator may exhibit degraded performance when operating conditions differ significantly from those encountered during training.

7 Hybrid Estimator

7.1 Chapter Overview and Design Approach

Hybrid estimation approaches aim to combine the capabilities of the model-based and data-driven estimators discussed throughout this work. Model-based methods provide physically interpretable structures derived from vehicle dynamics, while machine learning models offer flexible nonlinear mapping capabilities that may capture relationships not explicitly represented in simplified physical models. The hybrid approach investigated in this work therefore explores whether these complementary strengths can be combined to improve longitudinal velocity estimation.

To investigate this possibility, a residual-corrector architecture is adopted. In this configuration a model-based estimator is used as a baseline estimator, and its output is corrected at each time step by a neural network. The general structure of this architecture is illustrated in Fig. 7.1. The baseline estimator produces an initial estimate of the longitudinal velocity, while the neural network predicts a correction term which is added to this estimate to obtain the final hybrid estimate.

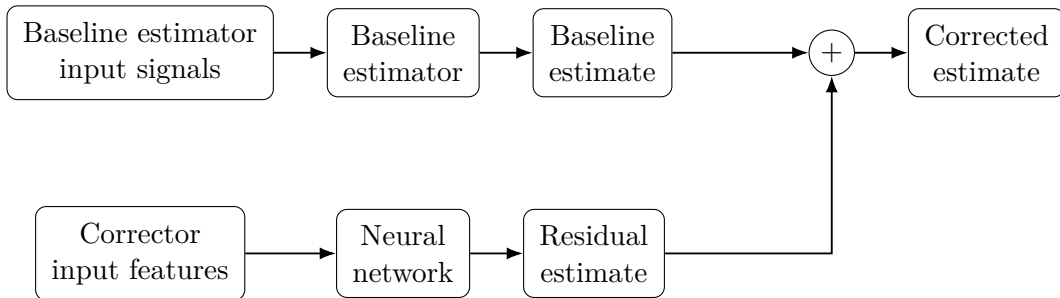


Figure 7.1: General residual-corrector architecture used in hybrid estimation.

In this configuration the neural network attempts to learn residual dynamics that are not captured by the baseline estimator. These residuals arise from unmodelled dynamics and algorithmic assumptions. By learning a mapping between the available signals and the resulting estimation error, the network aims to compensate for these effects while preserving the physically grounded structure of the baseline estimator.

Being the simplest model-based estimator developed in this work, the kinematic Kalman filter KF01 (see Section 5.3) is selected as the baseline estimator. This selection is consistent with the development principle adopted throughout the thesis of starting from simple models before introducing additional complexity. Furthermore, KF01 relies exclusively on signals that are directly available from standard vehicle sensors.

In principle, the neural network corrector could improve the baseline estimator either by exploiting the same information available to the baseline estimator or by incorporating additional

signals. In this work the hybrid estimator is deliberately restricted to the same information pool already available to KF01. This design choice isolates the contribution of improved information utilisation rather than additional sensing, reduces the dependence on additional sensor signals, and ensures that any potential performance gains cannot be attributed to additional measurements. Based on these principles, the specific design of the hybrid estimator is described in the following sections.

7.2 Hybrid Residual-Corrector Estimator (HYB)

7.2.1 Estimator Summary

The hybrid estimator combines the kinematic Kalman filter KF01 with a neural-network-based residual corrector. As discussed in Chapter 6, both multilayer perceptron (MLP) and long short-term memory (LSTM) architectures were investigated for data-driven estimation. Based on their behaviour on the development dataset, the LSTM architecture demonstrated more favourable performance and is therefore selected as the neural component of the hybrid estimator. Table 6.5 summarises the main characteristics of the final estimator configuration.

Table 7.1: Summary of the hybrid estimator

Item	Description
Estimator type	Hybrid (model-based + data-driven)
Estimation algorithm	Linear Kalman filter (KF01) combined with an LSTM neural network residual corrector
Physical model	Planar kinematic two-track vehicle model
Estimated states	v_x, v_y
Estimator inputs	$a_x, a_y, r, \Omega_i, \delta$
Key assumptions	KF01 estimate provides a reliable baseline for longitudinal velocity estimation KF01 residual signals contain dynamics that can be captured by the LSTM Relationships learned on the development dataset remain representative for the test dataset
Expected strengths	Improved accuracy when KF01 modelling assumptions break down (e.g. high slip, strong transients) KF01 provides stable baseline behaviour while the LSTM corrects systematic estimation errors
Expected limitations	LSTM correction may degrade when operating conditions diverge from the training conditions Additive correction may introduce oscillatory behaviour since no inherent smoothness constraint is imposed Correction limited to information available in KF01 signals — scenarios requiring additional sensing may not benefit
$i \in \{FL, FR, RL, RR\}$	

To maintain consistency with the data-driven estimator development, the same learning

framework from Chapter 6 is adopted. In particular, the fixed hyperparameters that were kept constant during the MLP and LSTM development (see Section 6.2.4) are retained for the hybrid estimator as well. However, while the hybrid architecture directly employs KF01 as the baseline estimator, the LSTM corrector used in the hybrid estimator differs from the standalone LSTM estimator. The neural network is adapted for its role as a residual correction module. The following sections discuss the specific design choices and present the final architecture.

7.2.2 Architecture and Signal Flow

The final hybrid estimator (HYB) follows the general structure shown in Fig. 7.1, combining a model-based baseline estimator with a neural-network-based correction module. For HYB, the kinematic Kalman filter KF01 is used as the baseline estimator, while the neural component is implemented using an LSTM network.

As discussed previously, the information pool available to the hybrid estimator is restricted to the sensing signals used by KF01. Consequently, the system-level input signals to the estimator are

$$a_x, a_y, r, \Omega_i, \delta,$$

where $i \in \{FL, FR, RL, RR\}$.

However, these signals are not provided directly to the LSTM residual corrector. Instead, the neural network operates on a set of derived signals constructed from the KF01 estimation process and the available inertial measurements. The feature set used by the LSTM corrector is defined as

$$\hat{v}_x^{KF01}, a_x, a_y, r, \nu_i^{KF01},$$

where $\hat{v}_x^{KF01}$ denotes the longitudinal velocity estimate produced by KF01 and ν_i^{KF01} represents the wheel-speed innovation signals of the Kalman filter, defined as

$$\nu_i^{KF01} = \Omega_i - \hat{\Omega}_i^{KF01}.$$

The innovation signals quantify the discrepancy between the measured and predicted wheel speeds and therefore provide a direct indication of situations in which the KF01 model assumptions become violated. In addition, the inertial signals a_x , a_y , and r provide information about the current vehicle dynamics and manoeuvre intensity, while the KF01 estimate $\hat{v}_x^{KF01}$ characterises the operating point of the estimator.

With this signal design, the expectation is that the LSTM corrector can recognise situations in which the KF01 estimator begins to deviate from the true vehicle behaviour and, using the available dynamic context and operating-point information, infer an appropriate correction to the baseline estimate.

To process the described signal set, the LSTM corrector employs a simple recurrent architecture composed of a single LSTM layer with 128 hidden units followed by a fully connected

(FC) output layer. This architecture matches the final standalone LSTM estimator developed in Chapter 6.

However, the temporal context differs from the standalone estimator. While the final standalone LSTM model uses a sequence length of 100 samples, the HYB estimator uses a sequence length of 10 samples, corresponding to the shortest sequence length evaluated in the final standalone LSTM model selection.

Since good validation convergence was observed during training, this configuration was retained for the final HYB architecture. The final LSTM corrector (LSTM_c) predicts a residual correction $\Delta v_x^{\text{LSTM}_c}$, which is added to the KF01 estimate to produce the final velocity estimate. The resulting hybrid estimator is summarised algorithmically in the following subsection.

7.2.3 Estimator Algorithm and Expected Behaviour

Given the architecture described in the previous subsection, the hybrid estimator operates sequentially at each sampling instant. The baseline estimate is first obtained from the KF01 kinematic Kalman filter, after which the LSTM corrector predicts a residual correction that is added to the KF01 estimate to produce the final velocity estimate.

Similarly to the standalone LSTM estimator described in Chapter 6, the LSTM corrector requires a complete input sequence before producing the first prediction. Therefore the input stream is padded during the initial samples. In the hybrid configuration the sequence length is $W = 10$, resulting in a significantly shorter warm-up period compared to the standalone LSTM estimator. All signals are initialised to zero during padding, which is consistent with the dataset where each run begins from rest.

Algorithm 5: Hybrid Estimator (HYB) Algorithm

Input: Sensor signals $a_x, a_y, r, \Omega_i, \delta$

Output: Estimated longitudinal velocity $\hat{v}_x^{\text{HYB}}$

Load trained LSTM_c network parameters and signal-scaling parameters;

Initialise KF01 state $\hat{\mathbf{x}}_{0|-1}$ and covariance $\mathbf{P}_{0|-1}$;

Initialise input buffer with $W - 1$ padded feature vectors;

for each sampling instant k do

 Acquire sensor signals;

 Run KF01 prediction and update steps;

 Compute KF01 estimate $\hat{v}_x^{\text{KF01}}(k)$ and wheel-speed innovations $\nu_i^{\text{KF01}}(k)$;

 Construct current feature vector $\mathbf{f}(k) = [\hat{v}_x^{\text{KF01}}, a_x, a_y, r, \nu_i^{\text{KF01}}]$;

 Update buffer: append current feature vector and discard oldest entry, maintaining a sliding window of W timesteps;

 Standardise input window using the fitted scaling parameters;

 Perform forward pass through trained LSTM_c and FC output layer;

 Compute residual correction $\Delta v_x^{\text{LSTM}_c}(k)$;

Output longitudinal velocity estimate $\hat{v}_x^{\text{HYB}}(k) = \hat{v}_x^{\text{KF01}}(k) + \Delta v_x^{\text{LSTM}_c}(k)$;

From a conceptual standpoint, the hybrid estimator aims to combine the robustness of the KF01 baseline with the nonlinear mapping capabilities of the LSTM network. The expectation is that the neural corrector can identify situations in which the modelling assumptions of KF01 become violated, such as during highly transient or high-slip manoeuvres, and provide an appropriate residual correction. At the same time, the presence of the KF01 baseline ensures that the estimator retains stable behaviour even when the neural correction is small or ineffective.

The performance of the proposed estimator is evaluated in the following chapter and compared with the other estimators developed in this work.

8 Results and Evaluation

In this chapter, the performance of the longitudinal velocity estimators developed in this work is evaluated. The estimators considered in this work belong to three main classes: model-based estimators (KF01, KF02), data-driven estimators (MLP, LSTM), and a hybrid estimator (HYB).

The architectures and development processes of these estimators are described in detail in Chapters 5–7. Each estimator provides an estimate of the longitudinal vehicle velocity v_x , which is compared against the ground-truth velocity obtained from the Driver-in-the-Loop simulator.

The evaluation is performed on the Test Dataset (TDS), corresponding to the NTC simulation run introduced in Chapter 4. Estimation accuracy is quantified using the root-mean-square error (RMSE) and the maximum absolute estimation error (MaxAE). For comparison, results produced by the industrial reference estimator (VDC) on the same simulation run are also included.

In addition to global performance metrics, estimator behaviour is analysed across different operating conditions of the vehicle, characterised by the dynamic envelope observed in the TDS. Further analysis investigates estimator behaviour in the presence of wheel slip, which was previously identified as a major obstacle to accurate longitudinal velocity estimation. Selected scenarios are then examined to highlight the limitations and behavioural differences between the proposed estimators and between the broader estimator classes.

8.1 Global Performance Comparison

The values of the two evaluation metrics for each estimator are reported in Table 8.1. Due to confidentiality reasons, selected signals are normalised with respect to the maximum vehicle speed observed across all datasets. Consequently, the MaxAE metric is expressed as a percentage of this reference value and is computed by normalising the maximum absolute velocity error by the global maximum vehicle speed.

Table 8.1: Global performance comparison on the Test Dataset

Estimator Class	Estimator	RMSE [m/s]	MaxAE [%]
Reference	VDC	1.4032	4.64
Model-based	KF01	0.8805	3.47
	KF02	0.9703	4.46
Data-driven	MLP	0.8644	12.37
	LSTM	0.6101	9.03
Hybrid	HYB	0.4871	3.43

As shown in Table 8.1, all investigated estimators achieve good overall estimation performance on the Test Dataset. In particular, all proposed estimators achieve a lower RMSE than the industrial reference estimator (VDC). However, this comparison should be interpreted with some caution. While the design and tuning strategy of the industrial reference estimator are not known, it is reasonable to expect that, as an industrial product, it was developed to operate with real sensor signals subject to noise and imperfections. The results therefore mainly indicate that the proposed estimators provide a suitable level of performance within the present experimental setting, rather than constituting a definitive performance advantage over the industrial solution.

In terms of RMSE, both data-driven estimators outperform the two model-based estimators, while the hybrid estimator achieves the lowest overall error. When compared with the best estimator of each class, HYB reduces the RMSE by 44.7% relative to the best model-based estimator and by 20.2% relative to the best data-driven estimator.

For the MaxAE metric, HYB again achieves the best performance. However, in this case the model-based estimators clearly outperform the purely data-driven approaches. Using HYB as a reference, it achieves a 1.2% improvement in MaxAE relative to the best model-based estimator and a 62.0% improvement relative to the best data-driven estimator.

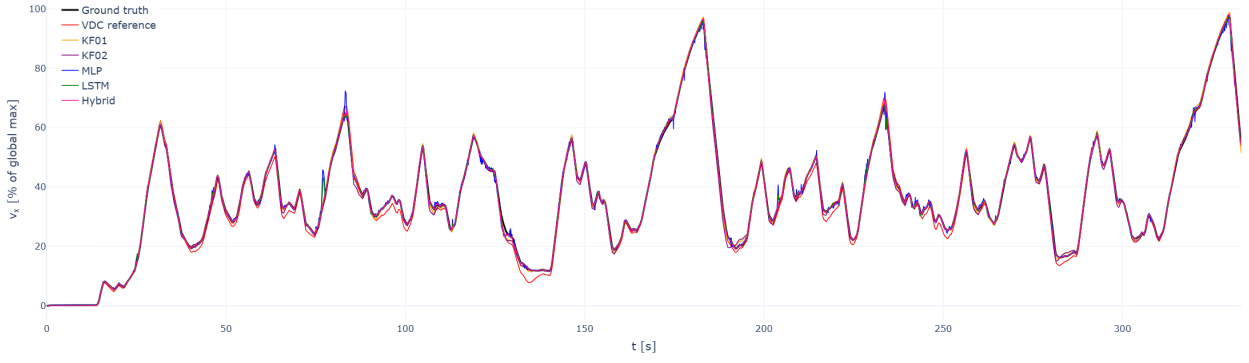
Within the present experimental setup, the hybrid estimator achieves the best performance among the considered approaches. At the same time, a clear difference in behaviour can be observed between the two other estimator classes. The data-driven estimators achieve lower RMSE values, indicating better average accuracy across the dataset, whereas the model-based estimators exhibit substantially lower maximum errors, suggesting more conservative behaviour in extreme conditions.

A closer look at the estimators within each class also reveals some differences. Within the model-based class, KF01 achieves lower errors than KF02 for both RMSE and MaxAE. This suggests that the additional modelling complexity and the expanded set of input signals used by KF02 do not translate into improved estimation performance under the present conditions. Within the data-driven class, LSTM achieves lower errors than MLP for both metrics. While this result cannot be attributed solely to the use of temporal sequences, it is consistent with the expected benefit of incorporating temporal context in the estimation process.

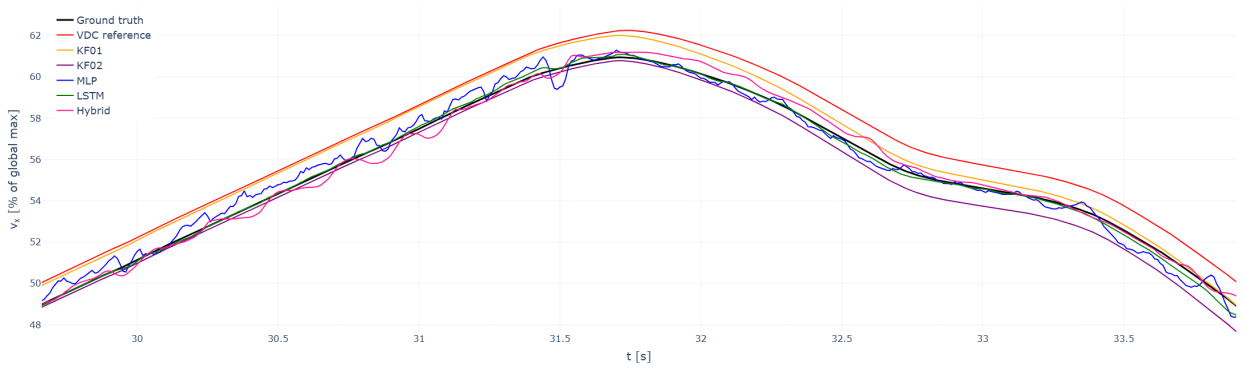
Time series

Figure 8.1 shows the time series of the longitudinal velocity estimated by all developed estimators together with the reference estimator and the ground-truth v_x signal. Figure 8.1a presents the full time series over the entire Test Dataset, while Figure 8.1b shows a zoomed-in segment in which the individual estimator trajectories can be more clearly distinguished.

From the figure it can be observed that all estimators broadly track the ground-truth velocity throughout the dataset. At the same time, some qualitative differences in behaviour are visible. In particular, the MLP and HYB estimates exhibit noticeable high-frequency fluctuations compared to the smoother signals produced by the other estimators. This behaviour



(a) Full time series over the Test Dataset.



(b) Zoomed-in segment showing the individual estimator trajectories.

Figure 8.1: Time-series comparison of the longitudinal velocity estimates produced by all developed estimators together with the reference estimator and the ground-truth signal.

can be related to the structure of these estimators, in which neural networks map noisy input signals directly to the velocity estimate while relying on little or no temporal context.

Within the time span shown in the figure, the data-driven and hybrid estimators generally follow the ground-truth signal more closely than the model-based estimators. These qualitative observations complement the quantitative results reported in Table 8.1 and provide an initial visual comparison of estimator behaviour over the Test Dataset.

8.2 Behaviour Across the Dynamic Envelope

While the preceding analysis evaluated estimator performance using global metrics and by describing the general behaviour observed in the estimate time series, estimation errors may not be uniformly distributed across the vehicle's operating domain. Certain driving conditions, such as heavy braking or combined longitudinal and lateral loading, are expected to be particularly challenging for longitudinal velocity estimation, primarily due to the increased likelihood of tyre slip. In Chapter 8 the driving data were characterised using GG diagrams to describe the dynamic envelope of the vehicle. To investigate whether estimator errors concentrate in specific regions of this operating domain, the Test Dataset was segmented according to the vehicle's

dynamic regime and analysed accordingly.

Dynamic Regime Classification

Four regimes were considered: braking, traction, high lateral acceleration, and combined loading. In addition, a fifth category referred to as regular driving was defined to represent all remaining samples in the test dataset that do not fall into the other four regimes. The classification was based on the ground-truth longitudinal and lateral accelerations obtained from the simulator.

To remove high-frequency oscillations, the acceleration signals were first smoothed using a moving-average filter. Timesteps were then classified into the dynamic regimes based on percentile thresholds of the smoothed signals: braking corresponds to the top 10% of negative a_x , traction to the top 10% of positive a_x , and high lateral acceleration to the top 10% of $|a_y|$. The combined loading regime was defined when the resultant acceleration exceeded the 90th percentile while both $|a_x|$ and $|a_y|$ were above their respective median values. Segments shorter than 0.2s were excluded to avoid fragmentation caused by short threshold crossings. Samples not meeting any of the above criteria were classified as regular driving. Where criteria overlapped, combined loading took precedence, followed by longitudinal regimes and finally the high lateral acceleration. With the described configuration, 64 segments belonging to the four considered regimes were identified in the test dataset. The resulting segmentation is illustrated in Fig. 8.2.

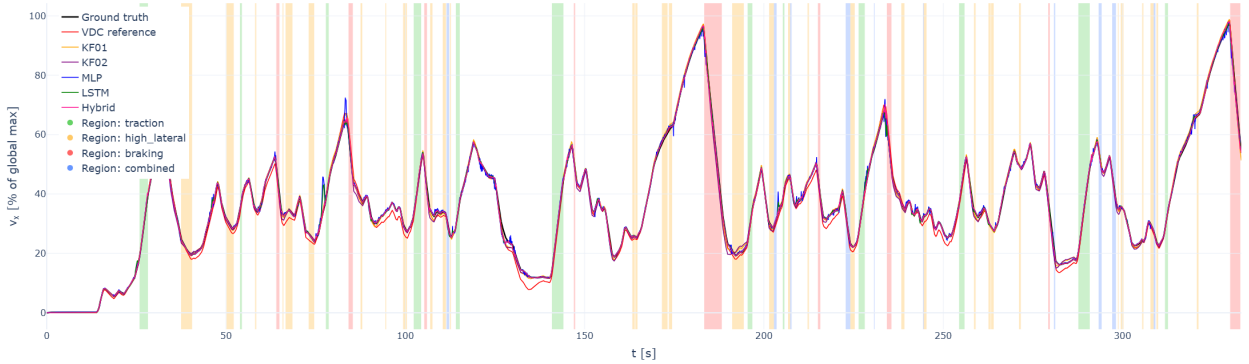


Figure 8.2: v_x time series for the TDS with the identified dynamic regimes.

Regime Performance Metrics

Two metrics were evaluated for each estimator within every dynamic regime. First, the RMSE was computed using only the samples belonging to each category. This indicates whether estimation accuracy degrades during particular driving conditions. Second, a high-error concentration ratio (HECR) was computed to determine whether large estimation errors are over-represented in a given regime relative to the dataset as a whole. For each estimator, the top

10% largest absolute errors were identified and the probability of observing such high-error samples within each category was compared to the global probability. The HECR is defined as

$$\text{HECR} = \frac{P(\text{high error} \mid \text{category})}{P(\text{high error} \mid \text{overall})}.$$

Values larger than one indicate that large estimation errors are over-represented in the corresponding regime relative to the dataset average, whereas values smaller than one indicate that the regime is less error-prone than average.

Estimator Performance Across Dynamic Regimes

The results for all estimators in terms of the defined dynamic regimes are shown in Tables 8.2 and 8.3.

Table 8.2: RMSE of each estimator within the identified dynamic regimes.

Estimator	Braking	Traction	Combined	High lateral	Regular	Overall
VDC ref	1.517	1.032	1.322	1.435	1.419	1.403
KF01	1.658	0.714	0.624	0.805	0.848	0.881
KF02	2.217	0.310	1.185	0.985	0.887	0.970
MLP	1.357	0.643	0.760	0.339	0.891	0.864
LSTM	0.789	0.333	0.553	0.223	0.646	0.610
HYB	1.329	0.455	0.492	0.200	0.430	0.487
Samples	1342	1923	642	3061	26345	33313

Table 8.3: High-error concentration ratio (HECR) across the dynamic regimes.

Estimator	Braking	Traction	Combined	High lateral	Regular
VDC ref	1.82	0.00	0.26	0.89	1.06
KF01	4.17	0.42	0.89	1.45	0.83
KF02	4.10	0.00	2.21	1.37	0.84
MLP	3.84	1.10	2.09	0.23	0.91
LSTM	3.09	0.64	2.02	0.52	0.95
HYB	6.45	1.59	2.34	0.89	0.75

RMSE results show that braking events have the greatest negative impact on the estimation accuracy. Across all six estimators, five proposed by this work and the reference estimator, there is an increase in the RMSE for braking events compared to the overall performance. HECR results show the same pattern, with high errors being more likely in the braking regime than the overall test run for all estimators. For HYB high errors are approximately 6.5 times more likely during braking than across the test dataset as a whole. This is consistent with the expectations since braking is associated with significant tyre slip.

Conversely, all estimators exhibit improved estimation accuracy and a significantly lower high error probability during traction events. While high traction demands are also associated

with tyre slip, this occurs if and when the tyres actually lose traction. This translates to lower longitudinal accelerations compared to what is achievable in the traction condition. Therefore, the traction regime as defined in the present work isolates events during which tyres are generally less likely to slip and thus the estimation task is made easier.

Although combined longitudinal and lateral loading is also considered a challenging operating condition, since the available tyre traction is shared between longitudinal and lateral directions, no strong patterns are observed in the combined regime in terms of RMSE. Four estimators show improved RMSE, while two exhibit degraded performance compared to the overall results. Out of the developed estimators, the split is three to two, with no clear distinction between the model classes. However, for four out of the five developed estimators, high errors are approximately 2 times more likely during the combined regime than overall. It is worth pointing out, however, that with the definition of the regime and the selected segmentation thresholds, only 1.9% of the total samples in the test dataset belong to the combined regime.

Under high lateral loading, the data-driven estimators and HYB exhibit a notable reduction in RMSE and high-error probability compared to their overall performance, while the model-based estimators show only marginal changes. High lateral acceleration regime is characterised by low longitudinal acceleration (57% lower than in the regular regime, with small standard deviation of 0.08 g compared to 0.26 g for the regular regime) indicating conditions with limited opportunity for longitudinal tyre slip, either caused by braking or high traction demands. Even though this suggests favourable conditions in terms of likelihood of tyre slip, purely model-based estimators remain subject to modelling simplifications such as the fixed steering ratio approximation or the assumption of no lateral tyre slip. Estimators incorporating neural network, on the other hand, may be better suited to exploit the reliable measurements.

Finally, the performance in the regular regime is comparable to the overall conditions both in terms of RMSE and HECR. Since the regular regime accounts for 79.1% of all test samples, HECR suggests that many high error events are still found in dynamic conditions not included in the other four analysed regimes.

8.3 Slip-Based Analysis

As most of the results from the dynamic-based analysis were discussed in terms of tyre slip, the following discussion investigates longitudinal slip directly as a key potential mechanism underlying the observed estimation errors. The adopted definition of slip in the present work is shown in Eq. (2.5). However, this definition relies on the longitudinal velocity of each wheel in the wheel reference frame $v_{wx,i}$, which for the front wheels depends on the vehicle lateral velocity v_y . Since no ground truth for v_y is available in the datasets, an alternative slip proxy is considered for the following analysis.

Slip Proxy Definition

The longitudinal slip proxy used in the following discussion is defined as

$$s_i = \frac{R_i \Omega_i - v_x}{v_x} \quad (8.1)$$

where R_i denotes the nominal effective rolling radius of wheel i , Ω_i the corresponding wheel angular velocity, and v_x the ground-truth longitudinal vehicle velocity. Positive values of s_i indicate traction slip (wheel spinning faster than the vehicle), while negative values correspond to braking slip (wheel rotating slower than the vehicle).

Slip-Based Classification

Since both positive and negative slip degrade the observability of v_x through the wheel-speed measurements, only the absolute value of the slip proxy is considered in the following analysis. The objective is to isolate operating conditions characterised by highly elevated slip from the remaining operating conditions of the test run, which are referred to as regular conditions.

To capture situations where wheel-speed information becomes unreliable, the analysis considers the maximum absolute slip across the four wheels at each time instant,

$$s_{\max} = \max(|s_{FL}|, |s_{FR}|, |s_{RL}|, |s_{RR}|). \quad (8.2)$$

This choice reflects the fact that if any single wheel experiences significant slip, the corresponding wheel-speed measurement becomes unreliable. Although the remaining wheels may still provide useful information, the estimator is effectively operating with at least one corrupted measurement.

The high-slip regime is defined using a percentile threshold on this signal. Samples for which $s_{\max}$ exceeds the 90th percentile of the signal over the entire test dataset are classified as high-slip conditions, while all remaining samples are classified as regular conditions. To avoid fragmentation caused by short slip spikes, segments shorter than 0.1 s are reclassified as regular conditions. With the described configuration, 75 segments were classified as high-slip in the test dataset. The resulting segmentation is illustrated in Fig. 8.3.



Figure 8.3: Slip proxy signals and identified high-slip conditions for the test run.

Estimator Performance Under Slip Conditions

Estimator performance under slip conditions is evaluated using the same metrics introduced in the previous analysis. The RMSE and high-error concentration ratio (HECR) are computed separately for high-slip and regular conditions. The results are summarised in Tables 8.4 and 8.5.

Table 8.4: RMSE of the estimators under high-slip and regular conditions in the test dataset.

Estimator	High slip	Regular	Overall
VDC ref	1.560	1.387	1.403
KF01	1.090	0.857	0.881
KF02	1.582	0.888	0.970
MLP	2.321	0.541	0.864
LSTM	1.748	0.330	0.610
Hybrid	1.090	0.380	0.487
(samples)	2976	30337	33313

Table 8.5: High-error concentration ratio (HECR) of the estimators under high-slip and regular conditions.

Estimator	High slip	Regular
VDC ref	1.66	0.94
KF01	1.90	0.91
KF02	3.50	0.76
MLP	5.03	0.61
LSTM	5.09	0.60
Hybrid	4.63	0.64

The results reported in Tables 8.4 and 8.5 show that all estimators exhibit degraded performance during high-slip conditions. Both the RMSE and the high-error concentration ratio increase relative to the overall performance levels, confirming tyre slip as a key mechanism underlying estimation errors. Among the developed estimators, the largest relative increase in RMSE is observed for the LSTM estimator, whose RMSE rises from 0.610 m/s overall to

1.748 m/s under high slip, while the smallest increase occurs for KF01, increasing from 0.881 m/s to 1.090 m/s. Similarly, the HECR values exceed unity for all estimators in the high-slip regime, indicating that large estimation errors occur disproportionately often during slip events.

The data-driven estimators are the most strongly affected by slip. For both the MLP and LSTM, high-error samples occur around five times more frequently during high-slip conditions than on average across the dataset. This behaviour is also reflected in the RMSE values, while both estimators achieve low errors during regular driving (0.541 m/s for the MLP and 0.330 m/s for the LSTM), their RMSE increases substantially under high slip (2.321 m/s and 1.748 m/s, respectively). This indicates that the data-driven estimators are particularly sensitive to operating conditions where wheel-speed measurements become unreliable.

Among the model-based estimators, KF02 shows a significantly stronger sensitivity to slip than KF01. The HECR increases to 3.50 for KF02 compared to 1.90 for KF01, and the RMSE under high-slip conditions rises to 1.582 m/s compared to 1.090 m/s for KF01. This is notable because KF02 incorporates a more complex vehicle model including explicit slip states, whereas KF01 relies on a simpler kinematic formulation. These results are in disagreement with the expected behaviour of KF02 discussed in Section 5.4.6. Additionally, they reinforce the observation from the global performance metrics that the additional modelling complexity of KF02 does not translate into improved performance.

The hybrid estimator also shows a strong increase in both RMSE and high-error concentration, with an HECR of 4.63 and an RMSE of 1.090 m/s under high slip. However, this result must be interpreted in the context of its overall performance. Although its RMSE increases substantially under high slip (1.090 m/s), this value remains comparable to KF01 and substantially lower than those observed for the purely data-driven estimators. The KF01 baseline therefore appears to provide a degree of robustness even when the learned correction becomes unreliable.

Finally, since braking was the dynamic regime that produced the largest performance degradation in the dynamic-regime analysis, it is instructive to compare the results obtained for braking with those observed under high-slip conditions. For the model-based estimators, HECR is higher in the braking regime than in the high-slip regime (KF01: 4.17 in braking compared to 1.90 in high slip; KF02: 4.10 in braking compared to 3.50 in high slip). In contrast, the data-driven estimators show higher error concentration during high-slip conditions than during braking (MLP: 5.03 in slip compared to 3.84 in braking; LSTM: 5.09 in slip compared to 3.09 in braking), while the hybrid estimator exhibits elevated sensitivity in both regimes. A similar trend is observed in the RMSE values. This observation suggests that data-driven estimators appear more directly affected by tyre slip itself, whereas model-based estimators show greater sensitivity to the broader dynamic regime in which slip occurs. The hybrid estimator shows more balanced sensitivity to both conditions, consistent with its architecture combining model-based and data-driven components. This result is highlighted in Table 8.6, where $\Delta RMSE$ represents the percentage change in RMSE within the specified condition relative to the overall RMSE on the test dataset. It is important to note that this observation is specific to the

estimator configurations and test conditions considered in this work, and further investigation would be needed to determine whether this pattern generalises to other scenarios.

Table 8.6: Relative RMSE increase and HECR for braking and high-slip conditions.

Estimator	RMSE Δ braking	RMSE Δ high slip	HECR braking	HECR high slip
VDC ref	+8.1%	+11.1%	1.82	1.66
KF01	+88.2%	+23.8%	4.17	1.90
KF02	+128.5%	+63.1%	4.10	3.50
MLP	+57.0%	+168.5%	3.84	5.03
LSTM	+29.4%	+186.5%	3.09	5.09
Hybrid	+172.8%	+123.8%	6.45	4.63

8.4 Qualitative Scenario Analysis

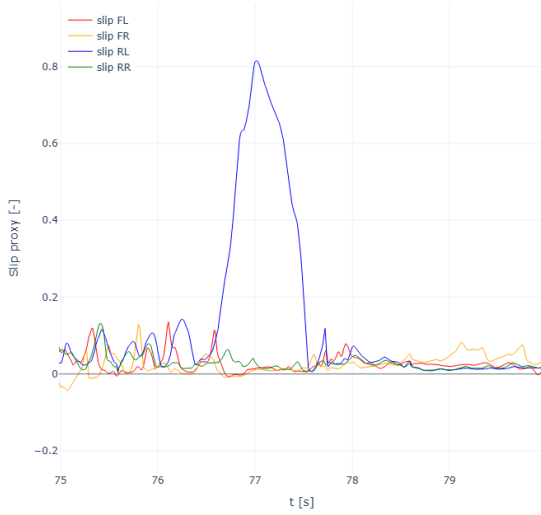
To further illustrate the behaviour of the estimators under, two cases from the dataset are analysed in more detail. Because the preceding slip analysis classified a time step as high slip whenever any wheel exceeded the defined threshold, the selected cases represent two distinct situations: one in which high slip occurs at a single wheel, and another where elevated slip is present across all wheels.

Single-Wheel Slip

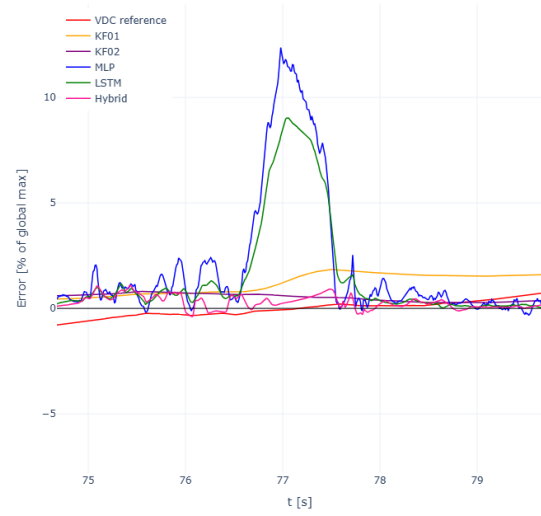
The first case corresponds to a situation in which only one of the wheels is at high slip. This case also coincides with the segment during which the two data-driven estimators experience their highest MaxAE values, 12.37% and 9.03% for the MLP and LSTM respectively. As shown in Fig. 8.4aa, one of the wheels experiences very high slip while the other wheels remain well within the previously defined threshold. Fig. 8.4b shows the behaviour of the estimators in this time span.

As can be seen, while the data-driven estimators experience the significant errors reported earlier, the model-based and hybrid estimators continue operating well. The third worst-performing estimator in this time span is KF01, reaching a maximum normalised MaxAE of 1.84%. It is worth noting that the hybrid estimator, despite incorporating a neural network component, exhibits the second-best behaviour with a MaxAE of only 0.91%. This supports the findings of the preceding analysis, which suggest that even when input signal degradation occurs, the hybrid estimator may handle this more effectively since it partially relies on signals coming from the model-based estimator.

Conversely, Fig. 8.5aa shows the second considered case. As seen in the figure, all four wheels exceed the previously defined high-slip threshold (the 90th percentile of the signal defined in Eq. (8.2)). Looking at Fig. 8.5b, all estimators exhibit a degradation of their respective performance in this situation. While the MLP and LSTM are still the worst-performing estimators, the performance across the estimator set is much more uniform.



(a) Wheel slip during the single-wheel slip event. One wheel exhibits high slip while the remaining wheels remain below the high-slip threshold.

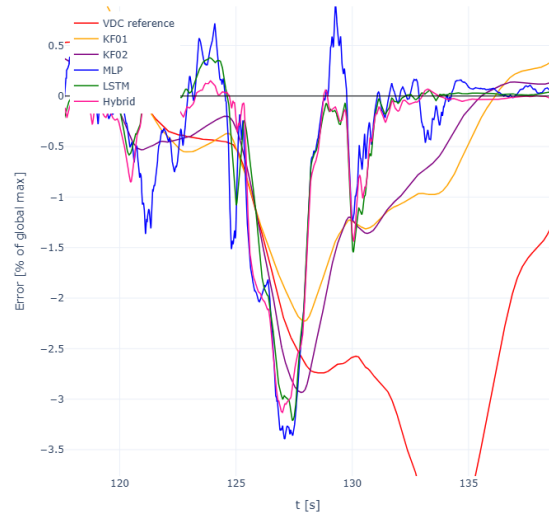


(b) Behaviour of the evaluated estimators during the same time interval.

Figure 8.4: Single-wheel slip scenario.



(a) Wheel slip during the multi-wheel slip event. All wheels exceed the high-slip threshold.



(b) Behaviour of the evaluated estimators during the same time interval.

Figure 8.5: Multi-wheel slip scenario showing the degradation of estimator performance when wheel-speed observability of v_x is severely compromised.

This observed difference suggests that model-based estimators can tolerate partial signal degradation by relying on their underlying physical model. However, severe degradation of wheel-speed-based observability of v_x still poses a major challenge for longitudinal velocity estimation. Another important observation is the magnitude of the errors experienced by the data-driven estimators in the first case. The magnitude of these errors highlights a potential

limitation of purely data-driven approaches in safety-critical applications. Overall, these observations highlight the importance of robustness to slip-induced measurement degradation in longitudinal velocity estimation and illustrate the differing behaviours of model-based, data-driven, and hybrid approaches when operating conditions deviate from nominal assumptions.

8.5 Summary of Experimental Findings

The results of this chapter highlight several key insights regarding the behaviour of the evaluated estimators. All five developed estimators outperform the industrial reference in terms of RMSE on the test dataset, demonstrating the effectiveness of both model-based and learning-based approaches when applied within a consistent estimation framework. Among the proposed methods, the hybrid estimator achieves the best overall performance in terms of both RMSE and MaxAE, indicating that combining model-based structure with data-driven correction can effectively leverage the strengths of both paradigms. Within the model-based approaches, the simpler kinematic filter KF01 outperforms the more complex KF02, suggesting that increased model fidelity and explicit slip modelling do not necessarily translate into improved estimation performance. Among the data-driven estimators, the LSTM consistently outperforms the MLP, confirming that incorporating temporal context improves the ability of neural models to infer longitudinal velocity from the available signals.

Despite their strong average accuracy, the purely data-driven estimators exhibit significantly larger MaxAE values and occasional error spikes, indicating a less predictable worst-case behaviour compared to the model-based approaches. In contrast, the model-based estimators produce smoother estimates with lower maximum errors, benefiting from the constraints imposed by the underlying physical model. The analysis across dynamic operating regimes further shows that estimation errors are not uniformly distributed across driving conditions. Regimes associated with significant tyre slip, particularly braking events, lead to the largest performance degradation, whereas conditions with limited slip, such as traction phases, tend to be easier for most estimators. High lateral acceleration with low longitudinal loading proved comparatively favourable for the data-driven estimators. While the condition is favourable in terms of wheel-speed observability of v_x , the underlying modelling approximations of model-based estimators may have limited further improvement in their performance.

A direct analysis of tyre slip confirms that slip itself is a major mechanism underlying estimation degradation. In particular, the results indicate that data-driven estimators are more susceptible to performance deterioration under high-slip conditions, where they can exhibit large errors even when only a subset of the wheels experiences slip. Model-based estimators, by contrast, appear comparatively more sensitive to the broader dynamic regimes in which such events occur. Although this observation emerges consistently across the analyses, the present work does not attempt to fully explain the underlying causes and further investigation would be required to generalise this finding. Finally, the qualitative scenario analysis illustrates that partial slip events primarily affect the data-driven estimators, whereas widespread slip across

all wheels degrades the performance of all approaches, confirming that the loss of wheel-speed-based observability ultimately limits longitudinal velocity estimation.

9 Conclusion and Future Work

9.1 Overview of the Study

This thesis investigated the problem of estimating the longitudinal velocity of a vehicle using data generated by a Driver-in-the-Loop (DiL) simulator. The data was provided by the industrial partner of the study and is based on a high-fidelity model of a high-performance production vehicle. The study examined three different approaches to longitudinal velocity estimation: model-based, data-driven, and hybrid methods combining elements of both paradigms. To represent these approaches, five estimators were developed. Two datasets were used for estimator development and parameter tuning, while a third dataset was reserved exclusively for performance evaluation.

Because the simulator provides idealised sensor signals, IMU measurements were artificially corrupted using noise and bias characteristics derived from specifications of available industrial IMU units in order to approximate realistic sensing conditions. Estimator development followed a design principle of starting from simple models and introducing additional complexity only when necessary.

Within the model-based class, two Kalman-filter-based estimators were developed: KF01, a linear kinematic Kalman filter, and KF02, which extends this formulation by incorporating wheel dynamics and longitudinal slip through the use of an Extended Kalman Filter. Within the data-driven class, two neural estimators were implemented: a Multilayer Perceptron (MLP), learning instantaneous mappings between input signals and longitudinal velocity, and a Long Short-Term Memory (LSTM) network that exploits temporal information. Both architectures were selected from a broader set of tested configurations. Finally, a hybrid estimator was proposed in which KF01 serves as a baseline estimator whose output is corrected by an LSTM-based residual model.

All estimators were evaluated and compared both against one another and against an industrial reference estimator provided by the project partner. Performance was primarily assessed using root-mean-square error (RMSE) and maximum absolute error (MaxAE). In addition to global performance metrics, the estimators were analysed across different dynamic operating conditions and levels of tyre slip in order to gain insight into the comparative behaviour of the different estimation approaches.

9.2 Main Findings

The results of this study demonstrate that all three investigated estimation paradigms—model-based, data-driven, and hybrid—are capable of achieving strong overall performance for longitudinal velocity estimation within the considered experimental framework. All five developed

estimators outperform the industrial reference estimator in terms of RMSE on the test dataset. However, these results should be interpreted in light of the experimental setting. While the design and tuning strategy of the industrial reference estimator are not known, it is reasonable to expect that, as an industrial product, it was developed to operate with real sensor signals subject to noise and imperfections. The result therefore mainly indicate that the proposed estimators provide a suitable level of performance within the present experimental setting, rather than constituting a definitive performance advantage over the industrial solution.

A closer examination of the results reveals a clear trade-off between model-based and data-driven estimation approaches. Despite their strong average accuracy, the purely data-driven estimators exhibit significantly larger MaxAE values and occasional error spikes, indicating a less predictable worst-case behaviour compared to the model-based approaches. In contrast, the model-based estimators produce smoother estimates with lower maximum errors, benefiting from the constraints imposed by the underlying physical model.

The hybrid estimator combines elements of both paradigms and achieves the best overall performance in terms of both RMSE and MaxAE, demonstrating that combining a physically structured baseline estimator with a data-driven correction mechanism can effectively leverage the complementary strengths of model-based and learning-based approaches.

Within the model-based approaches, KF01 outperforms the more complex KF02. This result indicates that increasing model complexity, does not necessarily lead to improved estimation performance. Within the data-driven approaches, the LSTM estimator consistently outperforms the MLP, suggesting that incorporating temporal context improves the ability of neural models to infer longitudinal velocity from the available signals.

Finally, the analysis shows that estimator performance depends strongly on the dynamic operating regime, with tyre slip emerging as a major mechanism underlying estimation degradation. While all estimators are affected by both dynamic loading and tyre slip, the results suggest a distinction in sensitivity: data-driven estimators appear more directly affected by tyre slip itself, whereas model-based estimators show greater sensitivity to the broader dynamic regimes in which such events occur. This observation warrants further investigation in future work.

9.3 Limitations of the Study

The findings of this study should be interpreted within the context of the experimental framework used. Several aspects of the present work limit the extent to which the results can be generalised beyond the considered setup.

The evaluation was conducted using data generated by a Driver-in-the-Loop (DiL) simulator rather than measurements obtained from a real vehicle. While the simulator provides a high-fidelity representation of vehicle dynamics and allows controlled experimentation across a range of operating conditions, real-world factors such as hardware imperfections, road irregularities, and environmental influences are not fully represented.

The work was conducted utilising a set of vehicle signals available within the simulator environment. While these signals are available on the reference vehicle, estimator performance may depend on signal quality. Because the simulator provides idealised sensor signals, IMU measurements were artificially corrupted using noise and bias characteristics derived from specifications of industrial IMU units in order to approximate more realistic sensing conditions. The IMU was selected for signal degradation since velocity estimation approaches relying on inertial measurements are particularly susceptible to performance degradation caused by measurement imperfections. Although this approach allows the influence of IMU imperfections to be studied, real sensors exhibit more complex behaviour, including temperature-dependent drift, calibration errors, and cross-axis effects, which were not explicitly modelled. In addition, other signals used by the estimators were not corrupted, meaning that the observed estimator performance may represent an optimistic estimate compared to real-world sensing conditions.

Finally, the estimators were evaluated in an offline setting and were not implemented on embedded automotive hardware. Consequently, aspects related to real-time execution, computational constraints, and integration within production vehicle control systems were not addressed in this work.

9.4 Future Work

Several directions for future work emerge from the present study. First, the proposed estimators should be evaluated using real vehicle measurements. While the Driver-in-the-Loop simulator provides a controlled and high-fidelity environment for algorithm development, validation under real operating conditions would allow the estimators to be tested in the presence of sensor imperfections, environmental disturbances, and other factors that are difficult to fully reproduce in simulation. In addition, deployment and testing on embedded automotive hardware would provide insight into real-time performance and integration within production vehicle systems.

Furthermore, a broader exploration of estimation approaches within each paradigm would allow a more comprehensive comparison between model-based, data-driven, and hybrid methods. While the present work focused on representative estimators within each class, other observer formulations for model-based estimation, as well as alternative neural network architectures such as gated recurrent units (GRUs), could be investigated to better understand the capabilities and limitations of each paradigm.

Further development of hybrid estimation strategies represents a promising research direction. In this work, the learning-based component was implemented as a residual corrector applied to a model-based baseline estimator. Although this was the best performing solution, neural networks could also be integrated in other ways, for example as virtual sensors providing providing pseudo-measurement signals within a larger estimator architecture.

Finally, the present work focused exclusively on the estimation of longitudinal velocity. In many practical applications, vehicle state estimators are designed to estimate multiple states simultaneously, including lateral velocity, sideslip angle, and tyre slip. Extending the approaches

investigated in this thesis to full vehicle state estimation could therefore provide further insight into the interaction between estimation architectures and vehicle dynamics.

Bibliography

- [1] M. Meywerk, *Vehicle Dynamics*. West Sussex, England: Wiley, 2015.
- [2] H. Pacejka, *Tire and Vehicle Dynamics*. Oxford, UK: Elsevier Science & Technology, 3 ed., 2012.
- [3] G. Genta and L. Morello, *The Automotive Chassis: Volume 1: Components Design*. Springer International Publishing AG, 2020.
- [4] F. Gustafsson, “Rotational speed sensors: Limitations, pre-processing and automotive applications,” *IEEE Instrumentation & Measurement Magazine*, vol. 13, no. 2, pp. 16–23, 2010.
- [5] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*. London: The Institution of Engineering and Technology, 2 ed., 2004.
- [6] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Boston, MA: Artech House, 2 ed., 2013.
- [7] X. Jin, G. Yin, and N. Chen, “Advanced estimation techniques for vehicle system dynamic state: A survey,” *Sensors*, vol. 19, p. 4289, 2019.
- [8] X. Wang, T. Chen, R. Wang, J. Lu, and G. Dou, “Review of state estimation methods for autonomous ground vehicles: Perspectives on estimation objects, vehicle characteristics, and key algorithms,” *Sensors*, vol. 25, p. 3927, 2025.
- [9] L. Imsland, H. F. Grip, T. A. Johansen, T. I. Fossen, J. C. Kalkkuhl, and A. Suissa, “Nonlinear observer for vehicle velocity with friction and road bank angle adaptation – validation and comparison with an extended kalman filter,” in *SAE Technical Paper*, no. 2007-01-0808, 2007.
- [10] E. Hashemi, S. Khosravani, A. Hajepour, A. Kasaiezadeh, S.-K. Chen, and B. Litkouhi, “Longitudinal vehicle state estimation using nonlinear and parameter-varying observers,” *Mechatronics*, vol. 43, pp. 28–39, 2017.
- [11] X. Xiao, X. Ren, and H. Li, “An inertial sequence learning framework for vehicle speed estimation via smartphone imu,” 2025.
- [12] Y. Liang, X. Zhang, Y. Liu, X. Tao, W. Liu, H. Jia, and H. Bai, “A robust gnss velocity estimation method combining doppler and carrier phase observations in complex urban environments,” *Advances in Space Research*, vol. 75, pp. 3838–3855, 2025.
- [13] D. Seo and J. Kang, “Dynamic-model-free vehicle velocity estimation using extended kalman filter with imu, steering angle, and wheel speed sensors,” *Measurement*, 2024.

- [14] L.-H. Zhao, Z.-Y. Liu, and H. Chen, “Design of a nonlinear observer for vehicle velocity estimation and experiments,” *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 664–674, 2011.
- [15] H. Guo, H. Chen, F. Xu, F. Wang, and G. Lu, “Implementation of ekf for vehicle velocities estimation on fpga,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 3823–3835, 2013.
- [16] J. Matuško, I. Petrović, and N. Perić, “Neural network based tire/road friction force estimation,” *Engineering Applications of Artificial Intelligence*, vol. 21, pp. 442–456, 2008.
- [17] D. Chindamo, B. Lenzo, and M. Gadola, “On the vehicle sideslip angle estimation: A literature review of methods, models, and innovations,” *Applied Sciences*, vol. 8, p. 355, 2018.
- [18] G. Napolitano Dell’Annunziata, V. M. Arricale, F. Farroni, A. Genovese, N. Pasquino, and G. Tranquillo, “Estimation of vehicle longitudinal velocity with artificial neural network,” *Sensors*, vol. 22, p. 9516, 2022.
- [19] M. Ok, S. Ok, and J. H. Park, “Estimation of vehicle attitude, acceleration, and angular velocity using convolutional neural network and dual extended kalman filter,” *Sensors*, vol. 21, p. 1822, 2021.
- [20] J. Kong *et al.*, “Vehicle state estimation based on lstm neural networks,” *World Electric Vehicle Journal*, vol. 13, no. 1, p. 1, 2021.