Framework for reproducible validation of a real-time capable vehicle lateral stability controller

TRISTANO, Mariagrazia

Framework for reproducible validation of a real-time capable vehicle lateral stability controller

Mariagrazia Tristano

A thesis submitted in partial fulfilment of the
requirements of
Sheffield Hallam University
for the degree of Doctor of Philosophy

September 2024

Abstract

Following the rise in technological advancements, in recent years the idea of mobility has undergone significant transformation. The growing tendency towards electrification has prompted innovative control strategies, which in turn have introduced new challenges. Current research goals in the automotive field may be summarised in two categories: ensuring passenger safety and enhance the robustness and reliability of such technologies. Within this framework, driver assistance systems like the Electronic Stability Control (ESC) are effective in avoiding potential loss of lateral stability. Extensive efforts have been made in the literature on the topic to refine control algorithms of this sort, but most of them show little tendency to progress beyond theoretical definition or software simulations. Despite such efforts, a staggering number of annual casualties is still registered, which calls for an enhancement of stabilization control strategies.

This thesis proposes a rigorous validation workflow for an individual-wheel-torque-based vehicle stability controller, aimed at establishing a robust and replicable testing protocol for a generic lateral stability controller to run in real time. The sequence of progressive validation steps starts with an initial design phase, where the controller is formulated according to specific performance requirements. It then envisions an offline co-simulation phase, where the controller undergoes preliminary testing in the form of software, running concurrently with a suitable vehicle testbench. Later, the controller and vehicle model are moved to a software-independent real-time-enabled testing platform in the so-called real-time co-simulation phase. Finally, the controller is turned into code and deployed to its target hardware (Hardware-in-the-loop), ultimately enabling full-scale vehicle validation.

Further noteworthy contributions include a novel strategy for vehicle reference behavior design, an innovative method to estimate what is arguably the strongest vehicle stability indicator, i.e. the sideslip angle, using both a machine learning-based and a parametric approach, and the analytical assessment of the vehicle stability region to reliably assess the operating condition of the vehicle.

Keywords: vehicle dynamics, sideslip angle, estimation, torque vectoring.

I hereby declare that:

1. I have not been enrolled for another award of the University, or other academic or professional organisation, whilst undertaking my research degree.
2. None of the material contained in the thesis has been used in any other submission for an academic award.
3. I am aware of and understand the University's policy on plagiarism and certify that this thesis is my own work. The use of all published or other sources of material consulted have been properly and fully acknowledged.
4. The work undertaken towards the thesis has been conducted in accordance with the SHU Principles of Integrity in Research and the SHU Research Ethics Policy.

Name: Mariagrazia Tristano

Date: Monday 16th September, 2024

Award: PhD in Material Science and Engineering

Director of Studies: Dr. Alex Shenfield

Supervisory team: Dr. Basilio Lenzo

Dr. Alan Holloway

Dr. Stephen Agha

Dr. Enrico Risaliti

Dr. Erik Wilhelm

Table of contents

List of abbreviations

CAN	Controller Area Network
CoG	Centre of gravity
DAQ	Data Acquisition
DOF	Degree(s) of freedom
ESC	Electronic Stability Control
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
GRNN	General Regression Neural Network
HiL	Hardware-in-the-loop
IMU	Inertial Measurement Unit
IIHS	Insurance Institute for Highway Safety
LQR	Linear Quadratic Regulator
MAB	MicroAutoBox
MAP	Map(s) of Achievable Performance
MPC	Model Predictive Control
OBD	On-Board Diagnostics
PI	Proportional-integral
PID	Proportional-integral-derivative
PINN	Physics-informed Neural Network
RMSE	Root-mean-square error
RNN	Recurrent Neural Network
RT	Real-time
RTPC	Real-time Personal Computer

List of symbols

a_1	Distance of front axle from centre of gravity (single-track model)
a_2	Distance of rear axle from centre of gravity (single-track model)
a_x	Longitudinal acceleration
a_y	Lateral acceleration
a_y^*	Limit lateral acceleration
$a_{y,max}$	Maximum lateral acceleration
c_0	Starting interpolation parameters to initiate optimal search
$c_{1,2,3}$	Root-rational tyre model interpolation parameters
c_{opt}	Optimised interpolation parameters
C_1, C_2	Front and rear cornering stiffnesses (single-track model)
C_α	Cornering tyre stiffness
C_λ	Longitudinal tyre stiffness
d_{lat}	Lateral IMU to S-motion distance
d_{long}	Longitudinal IMU to S-motion distance
$D0, \dots, D3$	Training amd validation dataset labels
e	Yaw rate error
e_β	Sideslip angle error
$E(e_\beta)$	Expected value of sideslip angle error
F_x	Longitudinal force
F_y	Lateral force
F_z	Vertical force
$G_{r,\delta}$	Input-to-disturbance Laplace gain
G_{r,M_z}	Input-to-output Laplace gain
I_β	Sideslip angle stability index

$I_{\max}$	Maximum stability index
I_r	Yaw rate stability index
I_t	Stability index threshold
J_z	Vehicle yaw inertia
k	Radians-to-degrees conversion factor
k_s	Scaling factor for stability yaw rate reference
K_i	Integral gain for PI controller
K_{lin}	Understeering gradient
K_p	Proportional gain for PI controller
l	Vehicle wheelbase
m	Vehicle mass
M_z	Direct yaw moment
N	Number of samples for RMSE evaluation
p_1, p_2	Parametric sideslip angle interpolation parameters
r	Yaw rate
r_h	Handling yaw rate reference
r_{IMU}	Yaw rate about IMU-centered axis
r_{ref}	Reference yaw rate
r_s	Stability yaw rate reference
$\dot{r}$	Yaw acceleration
r_{min}, r_{max}	Minimum and maximum yaw rate stability thresholds on phase plane
r_{wp}	Working point yaw rate coordinate on phase plane
R	Radius of curvature
R_w	Effective wheel radius
s	Laplace variable
s_l	Longitudinal slip ratio (Dugoff model)
s_s	Lateral slip ratio (Dugoff model)
v	Vehicle velocity
v_x	Longitudinal vehicle velocity
$v_{x,\mathrm{f}}$	Longitudinal velocity centered in S-motion
$v_{x,\mathrm{IMU}}$	Longitudinal velocity centered in IMU
v_y	Lateral vehicle velocity

$v_{y,\mathrm{f}}$	Lateral velocity centered in S-motion
$v_{y,\mathrm{IMU}}$	Lateral velocity centered in IMU
t_w	Track width
ΔT_i	Additional torque effort
$W(s)$	PI control law in Laplace domain
X_b	Longitudinal axis in car body reference frame (15 DOF model)
Y_b	Lateral axis in car body reference frame (15 DOF model)
Y_{sr}	Steering rack displacement with respect to car body reference frame (15 DOF model)
Z_b	Vertical axis in car body reference frame (15 DOF model)
Z_w	Vertical spindle lift with respect to car body reference frame (15 DOF model)
α	Tire slip angle
β	Sideslip angle
$\dot{\beta}$	Sideslip rate
β_{dyn}	Dynamic sideslip angle
β_{est}	Estimated sideslip angle
r_{IMU}	Sideslip angle at the IMU
β_{kin}	Kinematic sideslip angle
β_{meas}	Measured sideslip angle
β_{min}, β_{max}	Minimum and maximum sideslip angle stability thresholds on phase plane
β_{wp}	Working point sideslip angle coordinate on phase plane
δ	Steering angle
δ_{dyn}	Dynamic steering angle
$\delta_{dyn,max}$	Maximum dynamic steering angle
δ_{kin}	Kinematic steering angle
ε	Reference weight factor
μ	Friction coefficient
μ_{max}	Maximum friction coefficient
ω	Wheel rotational speed
ϕ	Roll angle
ψ	Yaw angle

ρ	Curvature
ρ_{max}	Maximum curvature
σ_{data}	Standard deviation
σ_x	Longitudinal slip
σ_y	Lateral slip
σ^2	Sideslip error variance
θ	Pitch angle
θ_w	Wheel/spindle relative rotation (15 DOF model)
ξ	Slip distribution coefficient (Dugoff model)

List of figures

Declaration of Ownership: All figures in this thesis have been generated by the author, except where otherwise indicated.

Chapter 1

Introduction

The current challenges of mobility can be summarised in two main objectives: ensuring safety and increasing the reliability of current technologies. To this end, driver assistance systems have a very important role, having become increasingly popular on passenger vehicles [1].

Among those, Electronic Stability Control (ESC) has proven to be an attractive feature. This system was first introduced by Bosch in 1995 [2] with the purpose of preventing a potential loss of lateral stability. ESC has been extensively explored in the literature: Zhai *et al.* in [3] designed an ESC algorithm for a 4 in-wheel motor electric vehicle, making use of a PID controller and fuzzy logic. Gimondi *et al.* [4] proposed a yaw rate tracking, rule-based ESC that would split the computed braking effort between the front and rear axle, favouring a more fail-safe approach. The combined advantages of ESC and active front steering are evaluated in [5], where integrated rule-based control aims to enhance driving dynamics.

To deal with lateral stability control, a number of vehicle quantities (called states) are chosen as insightful stability indicators, and they are closely monitored: when they reveal a potential stability loss, a selective braking action is applied on individual wheels. The accustomed driver may have recognised this need in many familiar driving situations: for instance, when entering a curve while the driving surface is slippery because of adverse weather conditions. It is critical that the detection and subsequent intervention to prevent a loss of stability happen in real time, so that rapidly-changing or even unexpected events can be managed to not cause any harm to the car occupants.

At present state, the ESC that is commonly installed on passenger vehicles is designed to intervene through friction braking only when imminent loss of stability is detected. More recently, the rising interest in electric mobility enabled a re-framing of the problem for vehicles featuring in-wheel motors, allowing independent allocation of both braking and traction torques on all four wheels. One of the most prominent techniques to do this is torque vectoring ([6],[7]). Torque vectoring calls for the definition of a continuously active controller that is able to constantly follow a given desired yawing behaviour, rather than only acting when triggered. This feature ensures complete freedom in managing stability correction while also avoiding energy dissipation of friction braking, which has made torque vectoring a topic of paramount interest in the research community.

Researchers in industry and academia keep striving to improve the performance of these kinds of control algorithms so as to significantly lower the number of fatalities caused by loss of stability of the vehicle. However, despite such proliferation of efforts, an impressive number of casualties is still registered: the World Health Organization estimates that every year 1.3 million people die in car accidents ([8]). According to the Insurance Institute for Highway Safety (IIHS) [9], in 2019 single-vehicle crashes, meaning accidents only involving a single vehicle, accounted for 44% of car occupant deaths: this category includes crashes occurring because of adverse weather, poor drainage of driving surfaces and insufficient steering while cornering. It is clear that devising better-performing stabilisation control strategies could reduce these casualties.

To make sure any devised control strategy will respond in the intended way, independently of the operative condition of the vehicle and the surrounding environment (e.g. adherence, atmospheric conditions), extensive testing is required. However, this implies an investment of both money and time, as well as physical resources. Testing each set of manoeuvres on a physical vehicle is unthinkable given the large number of testing scenarios of interest. For this reason, it is best to refrain from testing on the full-scale vehicle early on and thoroughly follow a gradual validation path instead.

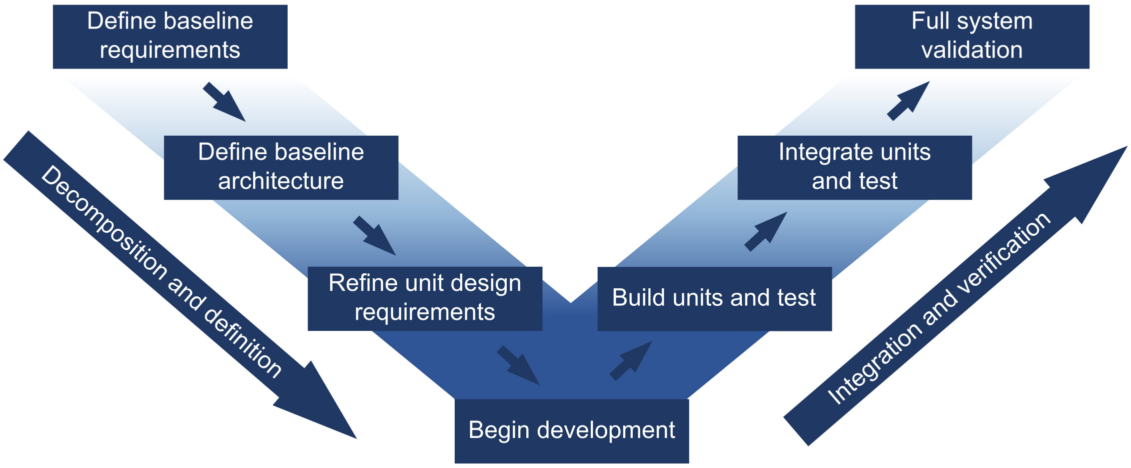

Define baseline requirements
Define baseline architecture
Refine unit design requirements
Begin development
Build units and test
Integrate units and test
Full system validation
Decomposition and definition
Integration and verification

Figure 1.1: Vee chart representation (adapted from [10] and [11]).

The control strategy evaluation is performed in simulation environments at first using software containing high-fidelity models mimicking the behaviour of the target vehicle. Next, dedicated hardware is introduced so that the control application can be tested in increasingly more realistic driving scenarios. This procedure requires a number of intermediate steps, where hardware is gradually integrated to replace simulation blocks.

A clear graphical portrayal of the validation process is introduced in [10], as a “Vee” chart. As evidenced in Fig. 1.1, the proposed procedure splits the flow of operations into a descending branch and an ascending one. The descending branch features an initial design phase, where user requirements for the product are analysed and decomposed into unit-sized problems. Design requirements are then established for such units, enabling application development. This key operation is the link between the descending branch and the ascending one, in which individual units are tested and integrated until the full product can be validated, ensuring compliance to standards set in the design phase.

This thesis presents the validation workflow of an individual-wheel-torque-based vehicle stability controller, with the purpose of introducing a solid, reproducible testing procedure for a generic lateral stability controller. The proposed controller is continuously active rather than being event-triggered, and constantly provides the vehicle with a reference that prioritises handling or safety according to the outcome

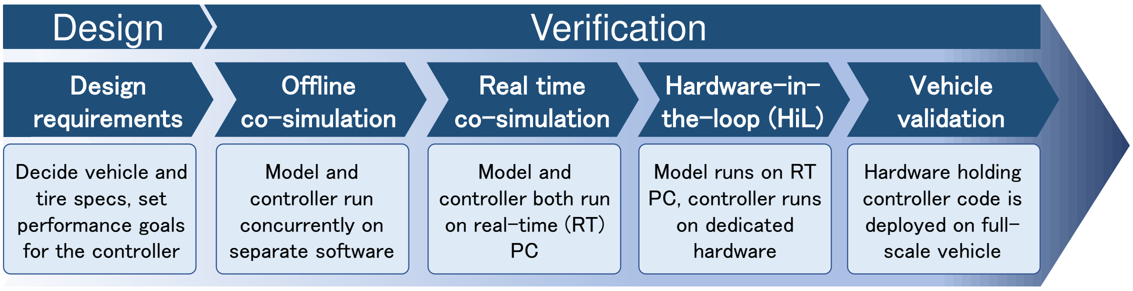

Design
Verification
Design requirements
Offline co–simulation
Real time co–simulation
Hardware–in–the–loop (HiL)
Vehicle validation
Decide vehicle and tire specs, set performance goals for the controller
Model and controller run concurrently on separate software
Model and controller both run on real–time (RT) PC
Model runs on RT PC, controller runs on dedicated hardware
Hardware holding controller code is deployed on full–scale vehicle

Figure 1.2: Proposed validation workflow, adapted from the Vee chart.

of a real-time stability assessment. Controller development is conducted alongside the layout of an appropriate test bench in charge of assessing the vehicle response to the proposed control strategy throughout the validation stages. The test bench mimics the full-scale vehicle behaviour through a suitable vehicle model.

Although the "Vee" chart is very well-renowned in the literature, its practical application is subject to the application at hand, and heavily relies on both the chosen test bench and the purpose of the control application. As evidenced in [11], the length of the descending branch of the "Vee" chart is subject to the level of detail one aims to achieve: a larger number of units to be designed implies a longer branch. In the broader context of the validation framework, the controller and the vehicle model are the only units of interest. The controller is the core unit to be designed and tested throughout consecutive stages, while the vehicle model is a pre-existing companion platform that does not require any testing. The descending branch would only feature the design of the controller, while the ascending branch would include all the progressive validation steps integrating both pre-existing and newly-conjured units. Consequently, a linear workflow may be envisioned instead, presented in Fig. 1.2. The proposed workflow preserves the nomenclature from the "Vee" chart, as it features a design phase and a verification phase, and clearly highlights details of the latter stage from simulation to full validation.

The adapted structure in Fig. 1.2 features the following of steps:

1. **Design phase.** The controller performance requirements are stated. Controller design is then closely paired with vehicle model design, to accommodate the testing needs of the control system.

2. **Offline co-simulation.** The controller is realised in a simulation environment using MATLAB/Simulink ®, and the vehicle model is built using Simcenter ™ Amesim ™. The two are interfaced and run concurrently in a co-simulation framework.

3. **Real-time co-simulation.** Both the controller and the vehicle model are run in co-simulation on a software-independent real-time platform to ensure the controller in principle meets real-time requirements. Any complications related to hardware can be addressed here. Progress up to this point was presented at the 2022 IEEE Vehicle Power and Propulsion Conference (VPPC) [12].

4. **Hardware-in-the-loop.** The controller algorithm is moved to the dedicated hardware in charge of running it on the full-scale vehicle. Such hardware is interfaced with the previously-introduced real-time platform, which runs the vehicle model. The two separate hardware elements communicate using the Controller Area Network (CAN) protocol, a widely used and well-established communication standard in automotive applications [13]. The presented workflow including the hardware-in-the-loop results was published on IEEE Transactions on Vehicular Technology (Q1 quartile) [14].

5. **On-vehicle validation.** The dedicated hardware running the controller algorithm is deployed on the full-scale vehicle.

Within this framework, the following novel contributions are achieved:

- **A novel strategy for reference behaviour design.** For the controller to affect the vehicle behaviour, both a control variable and a target behaviour have to be chosen. A novel strategy to design the target reference vehicle behaviour is proposed in Chapter 3.4.2. This work was presented at the 2021 International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD 2021) [15].

- **A novel method to tackle stability indicator estimation for control purposes.** For the correct triggering of the control algorithm a number of relevant stability indicators have to be both accurate and readily available. In the context of vehicle dynamics one of the most powerful stability indicators, the sideslip angle, is difficult and expensive to measure. If measuring an indicator is not an option, estimating it often provides a quite compelling alternative. Two estimation strategies have been devised to that end, both relying on easily-measured vehicle quantities:

  - A machine learning-based sideslip angle estimator, discussed in Chapter 4.1 and presented at the 2023 International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD 2023) [16].
  - A parametric sideslip angle estimator (Chapter 4.2, to be presented at the 2024 International Symposium on Advanced Vehicle Control (AVEC 2024) [17] and accepted for publication in Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering [18]).

  The latter strategy has been validated in a full-vehicle test programme, yielding promising real-time results.

- **A novel analytic assessment of the vehicle stability region.** The controller should not always intervene when the vehicle is running: in fact, its action is only required when vehicle stability is at stake. To assess whether the operating condition of the vehicle is stable or not, the literature offers a plethora of graphical and numerical methods. Here, an innovative, analytical and reliable approach to tackle the same issue is presented in Chapter 3.4.1, readily providing an assessment of the operating condition stability at each instant in time (manuscript currently under review for publication on Vehicle System Dynamics [19]).

The thesis is structured as follows: Chapter 2 gives an overview of the key vehicle dynamics concepts needed to address the challenges discussed above; Chapter 3 provides a detailed description of all the components present in the validation framework; Chapter 4 presents two novel sideslip angle estimation strategies; Chapter 5 follows the linear validation flow providing an appropriate set of results; Chapter 6 explores the real-time, full-scale vehicle feasibility of the novel parametric sideslip angle estimator. Conclusions are reported in Chapter 7.

Chapter 2

State of the art

Vehicle dynamics is the study of the forces that govern the behaviour of a vehicle. This branch of the automotive field encompasses the complex interactions between the vehicle's components, such as suspensions, tyres, and chassis, as well as external factors like friction effects and driver inputs. When aiming to investigate the impact of such phenomena on the stability and response of the vehicle across diverse driving conditions, knowledge of some pivotal concepts is required.

2.1 Reference frame

Setting a frame of reference for the vehicle is crucial, as it establishes a solid basis for motion analysis. All the evaluations conducted in this body of work make use of the ISO 8855 intermediate system [20], shown in Fig. 2.1. Strict adherence to ISO 8855 requires the lateral force and tyre slip angles to have opposite signs: however, aiming to match the sign convention in [21], positive lateral forces are hereby associated to positive tyre slip angles.

The chosen frame of reference is centred on the vehicle centre of gravity (CoG) and features three mutually perpendicular directions of motion, respectively along the roll (longitudinal) axis, pitch (lateral) axis and yaw (vertical) axis. Furthermore, the considered ISO standard specifies the sign convention for longitudinal and angular motion as follows:

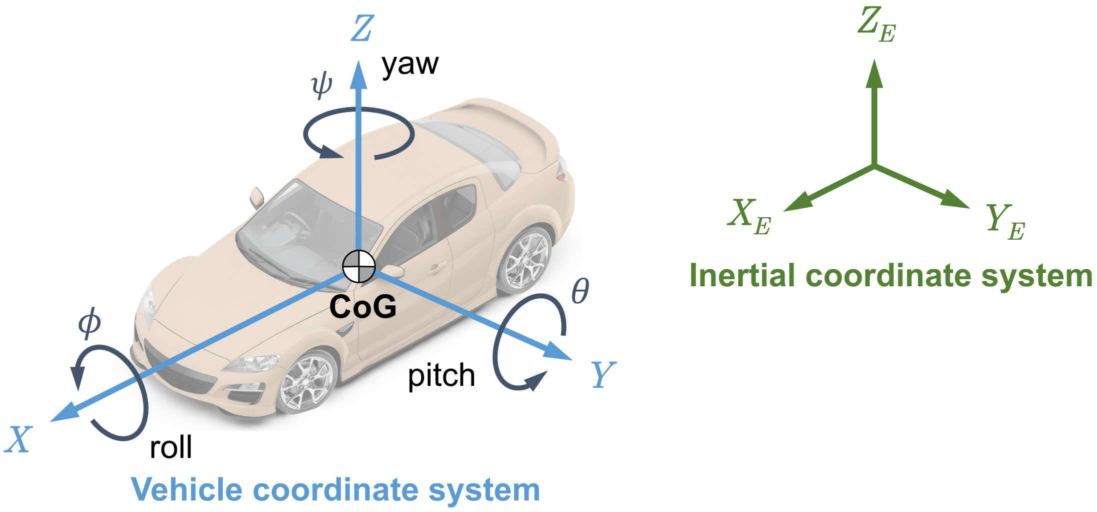


Figure 2.1: ISO 8855 intermediate system reference frame [20].

- The X axis describes longitudinal motion. It is aligned with the vehicle's travel direction, perpendicular to the gravity vector. The positive longitudinal direction corresponds to the forward direction of motion: a positive rotation is consequently identified through the right-hand rule. A rotation about the X axis is quantified by the roll angle θ.

- The Y axis represents lateral motion. It is perpendicular to both the X axis and the gravity vector. It is positive when pointing left with respect to the direction of travel. The right-hand rule defines a positive rotation around it, amounting to the pitch angle ϕ.

- The Z axis defines the vertical motion. It is parallel to the gravity vector and positive when pointing upward. The positive direction of rotation is found through the right-hand rule and amounts to the yaw angle ψ.

For the sake of completeness, an earth-fixed (inertial) reference frame ($\mathrm{X_E}, \mathrm{Y_E}, \mathrm{Z_E}$) has also been added to Fig. 2.1, following the same sign convention as the vehicle coordinate system. More specifically:

- The $\mathrm{X_E}$ axis is aligned with the vehicle's travel direction and is positive when following the forward direction of motion.

- The $\mathrm{Y_E}$ axis is perpendicular to both the $\mathrm{X_E}$ axis and the gravity vector, and is positive when pointing left with respect to the direction of travel.
- The $\mathrm{Z_E}$ axis defines the vertical motion and is positive when pointing upward.

This thesis focuses on vehicle yaw stability for control purposes, hence roll and pitch dynamics will be neglected.

2.2 Single-track model

Characterising the dynamic behaviour of a vehicle involves creating a vehicle model, which is a mathematical representation of the vehicle itself. The literature offers an abundance of vehicle models with different representation abilities: those featuring a considerable number of simplifications with respect to the full-scale vehicle will have a low computational load, but also a lower accuracy when attempting to portray phenomena like load transfer and suspension response, especially during dynamically challenging manoeuvres. On the other hand although more accurate models capture dynamic behaviour better, their elaborate and detailed formulation will require a higher computational time. Since the work in this thesis focuses on yaw behaviour, disregarding the roll and pitch dynamics of the vehicle, the single-track model is considered for this application.

The single-track model is a simplified representation of a vehicle that assumes the vehicle has one front and one rear wheel. Motion is hereby characterised by the lateral position and yaw angle of the vehicle: these two represent the degrees of freedom (DOF) of the model [22]. Despite it lacking a more nuanced representation of phenomena within the vehicle, it is still regarded as a strong candidate for vehicle lateral stability assessment due to its straightforward formulation and adequately representative results. Fig. 2.2 shows the single-track model with the relevant geometrical parameters: the CoG distance from the front and rear axle, a_1 and a_2 respectively, and the distance between the front and rear axle l, called wheelbase.

Fig. 2.2 also features a number of relevant variables:

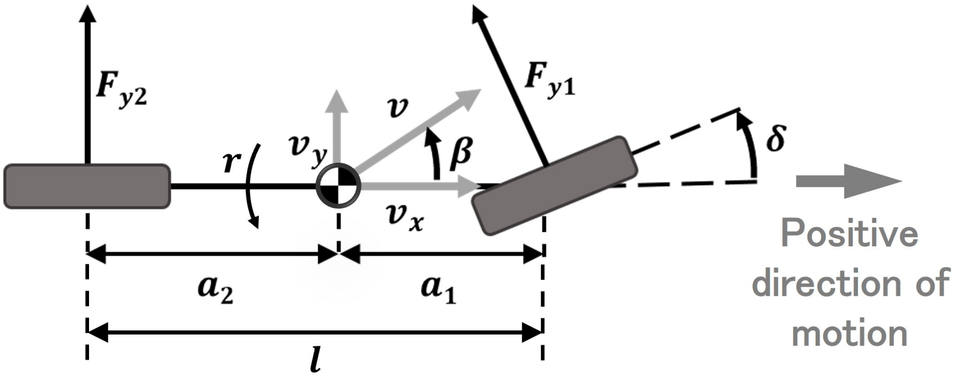


Figure 2.2: Single-track model.

- **Vehicle velocity** v, along with its longitudinal and lateral components v_x and v_y.
- **Steering angle** δ: angular distance between the orientation of the steering wheel at the front and the longitudinal direction of travel of the vehicle.
- **Yaw rate** r: rate of change of the yaw angle.
- **Front and rear lateral forces** F_{y1}, F_{y2}: lateral force components generated from the contact between the wheel and the ground, with direction perpendicular to the travel direction of the wheel.
- **Sideslip angle** β: angle between the vehicle velocity v and its longitudinal component v_x. It may be defined at any point within the vehicle frame, but it is commonly observed at the four wheels or at the CoG, as depicted in Fig. 2.3.

Visibly from Fig. 2.3, one could define the tangent of the sideslip angle as the ratio between the lateral velocity v_x and the longitudinal velocity v_y: in practice, the operative range of the sideslip angle is so narrow that the sideslip angle β can simply be approximated to the aforementioned ratio of velocity components. This introduces the so-called small angle approximation, considered in numerous studies within the literature, namely [23, 24], as argued in [25]. The small angle approximation is expressed in Eq. (2.1) and considered to be true from this point forward.

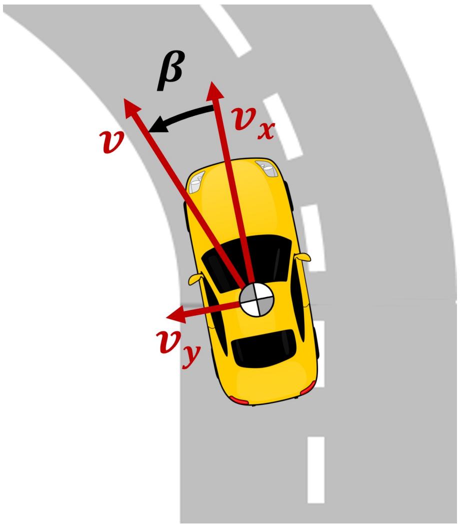


Figure 2.3: Vehicle sideslip angle, centered at the vehicle CoG.

$$\beta = \text{atan}\left(\frac{v_y}{v_x}\right) \approx \frac{v_y}{v_x} \tag{2.1}$$

For this reason, the single-track model is expressed using two dynamic equations, reported in Eq. (2.2). Further parameters are included, namely the vehicle mass m and the vehicle yaw inertia J_z.

$$\dot{\beta} = \frac{F_{y1} + F_{y2}}{m \cdot v_x} - r \tag{2.2a}$$

$$\dot{r} = \frac{F_{y1}a_1 - F_{y2}a_2}{J_z} \tag{2.2b}$$

A few important remarks arise:

1. The steady-state cornering response is expressed by two dynamic equations featuring dependence on the lateral forces. As a matter of fact, motion is a consequence of the shear forces occurring between the wheels and the relative motion of the ground: this calls for the need of modelling tyre dynamics as well as vehicle dynamics, since the latter would be meaningless without the former. Tyre models are discussed in Section 2.3.

2. Single-track model equations convey changes in time of the sideslip angle β and of the yaw rate r: both of them are regarded as strong stability indicators,

since they both quantify the turning speed of the vehicle about its yaw axis. For a successful stability assessment, both quantities need to be monitored. Notions of stability are provided in Section 2.4.

2.3 Tyre models

For the vehicle motion to be initiated, the torque coming from the transmission system has to be conveyed to the road. This happens by means of pneumatic tyres. Tyres are made of rubber, a viscoelastic material whose intrinsic properties include the ability to deform upon contact with the road, localised in the area called contact patch. Tyres essentially cover two tasks:

- Sustain the vehicle weight by means of the inflation tension caused by the pressure of the air in their chamber: a resultant reaction vertical force can be then observed on the contact patch, the normal load F_z.
- Convert the torques coming from the wheel rims into contact forces at the contact patch. Tyre motion will be affected by rolling resistance, due to the viscoelastic properties of rubber. A lower traction force than the one being conveyed by the transmission will be available at the tyre level, resulting in a lower velocity at the contact patch.

As explained in [21] the difference between the conveyed and the actual traction force translates to discussing respectively two different velocities:

- Pure rolling velocity, observed in theoretical absence of rolling resistance.
- Wheel travel velocity, actually registered at the tire as a result of rolling resistance.

The discrepancy between these two velocities is quantified by the so-called slip. Since both velocities feature a longitudinal and lateral component, it is natural to define a longitudinal slip σ_x as well as a lateral slip σ_y. Wheel slip knowledge is very valuable for tyre modelling, where it often appears under the form of slip ratio, where the slip

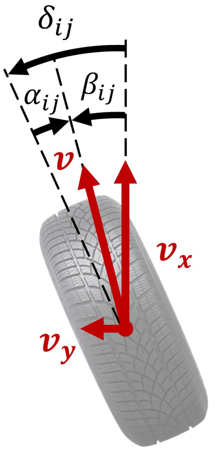


Figure 2.4: Velocities and relevant angles at the wheel level.

is essentially normalised over the wheel velocity. The change in velocity components induced by slip leads to an angle forming between the orientation of the wheel and its actual travel direction, called tyre slip angle α. The tyre slip angle α may be defined at each wheel, using the notation α_{ij}, where $i \in \{\text{front,rear}\}$ and $j \in \{\text{left,right}\}$, as portrayed in Fig. 2.4. The congruence equation (Eq. 2.3) explains the tyre slip angle in relation to other relevant angles at the wheel, namely the steering angle δ and the sideslip angle β.

$$\alpha_{ij} = \delta_{ij} - \beta_{ij} \tag{2.3}$$

Given that Eq. (2.3) refers specifically to the wheel frame, the role that the entire vehicle body plays within this formulation must be considered. The tyre slip angles at the front and rear single-track axles, respectively α_1 and α_2, are strongly affected by the vehicle yaw rate. As the yaw rate is positive counterclockwise and the tyre slip angle is positive clockwise (following the sign convention in [21]), the yaw rate component will be subtracted from the front tyre slip angle (as it will counter its increase) and summed to the rear tyre slip angle (as it favours lateral slipping of the rear wheel). Final expressions are detailed in Eq. (2.4).

$$\begin{cases} \alpha_1 = \delta - \beta - \dfrac{ra_1}{v_x}, & \text{(2.4a)} \\ \alpha_2 = -\beta + \dfrac{ra_2}{v_x} & \text{(2.4b)} \end{cases}$$

The opening statement in Section 2.2 highlighted the importance of using a mathematical representation of the vehicle for its dynamics to be investigated. This also stands true when discussing tyre behaviour: it is indeed important to examine the tyre behaviour in a range of scenarios that is accordingly broad, both in terms of wheel motion and road friction conditions [26]. Investigating the stability of a vehicle implies studying the critical friction phenomena happening at the contact patch, which inherently means that equations modelling the vehicle behaviour are of limited value unless they are linked to a tyre model formulation.

A diverse assortment of tyre models is available in the literature, all aiming to mimic the relationship between the tyre slip angle α and the lateral force F_y [27]. For small angles, the two variables seemingly maintain a proportional relationship through a constant C_α called cornering stiffness [26]: this dependence is expressed in Eq. (2.5), the so-called constitutive equation.

$$F_{y,ij} = C_{\alpha,ij}\alpha_{ij} \tag{2.5}$$

The cornering stiffness is observed to be higher for higher vertical loads F_z, as portrayed in Fig. 2.5. As discussed in Chapter 2, positive tyre slip angles are associated to positive lateral forces, which explains the constitutive curve being in the first and third quadrant of Fig. 2.5.

Similarly, a longitudinal stiffness C_λ can be introduced, to account for the proportionality between the longitudinal tyre force $F_{x,ij}$ and the longitudinal slip σ_x.

However, for tyre slip angles exceeding this range, an increasingly higher non-linear behaviour is observed due to the tyre reaching its peak lateral force (hence critical friction condition): consequently, further increasing the tyre slip angle will not actually result in a proportional increase in lateral force. From this perspective,

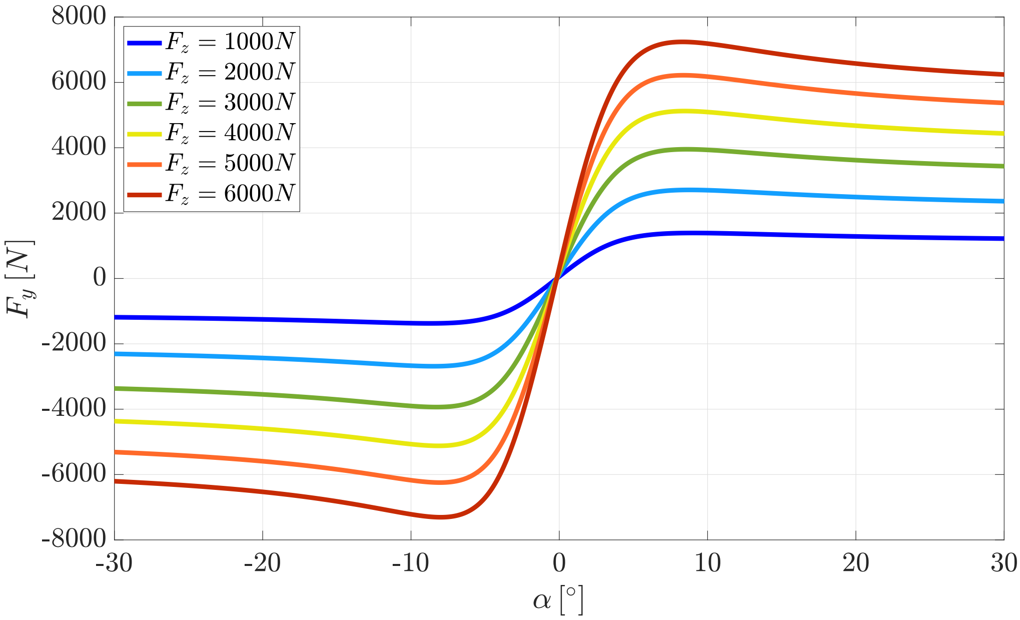


Figure 2.5: Common appearance of a constitutive plot for varying normal loads F_z on the wheel, using the Magic Formula tyre model [28]. Positive tyre slip angles are associated to positive lateral forces.

it appears evident that non-linear models are required for a correct portrayal of the phenomena in safety-critical cornering conditions, as that inherently involves working at the tyre friction limits.

An empirical approach, referred to as Magic Formula, was proposed in [28]: it accounts for non-linearities by describing empirical fitting coefficients. Its versatility allows it to represent diverse tyre types and behaviours, making it applicable to various vehicles and driving conditions. As shown in Fig. 2.5, for small tyre slip angles the Magic Formula envisions a linear proportionality with the lateral force, which then becomes non-linear until a maximum lateral force is reached, corresponding to the peak adhesion condition. Following the peak, further increases of the tyre slip angle lead to progressively lower lateral force values. Despite its accuracy in predicting tyre forces, the Magic Formula requires extensive tyre testing for parameter identification, whose accuracy heavily depends on reliability of available tyre data. The model's computational complexity can be a drawback, impacting real-time applications. Moreover, concerns about overfitting may arise due to the need for a large number of parameters.

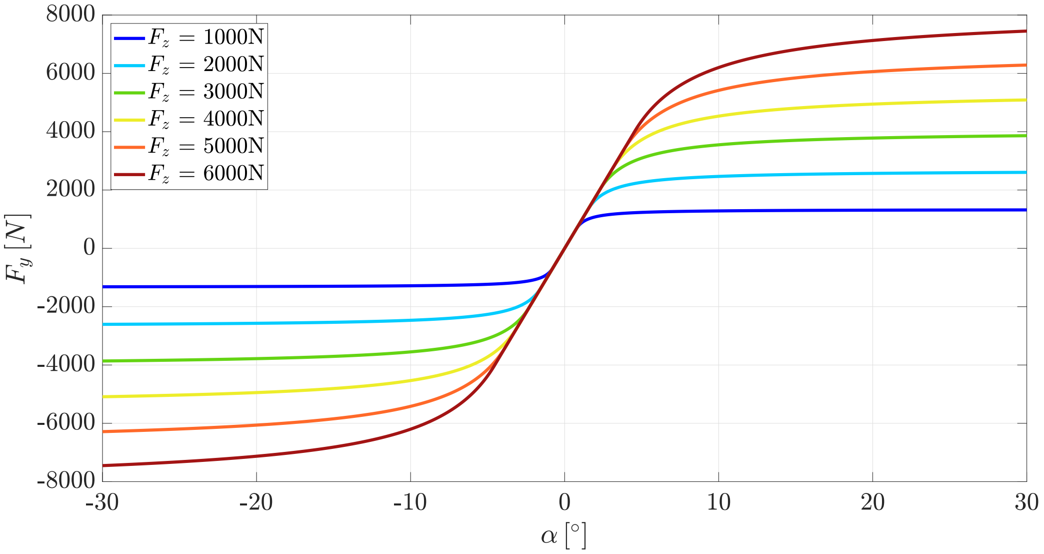


Figure 2.6: Common appearance of a constitutive plot for varying normal loads F_z on the wheel, using the Dugoff tyre model [28]. Positive tyre slip angles are associated to positive lateral forces.

An analytical formulation for tyre forces under combined longitudinal and lateral load, which later became known as Dugoff tyre model, was presented in [29]. Differently from the Magic Formula tyre model, the Dugoff tyre model formulation does not present a maximum adhesion peak, as shown in Fig. 2.6. This would significantly impact stability investigations for slip conditions close to lateral adhesion limits. To accurately capture the nonlinear tyre behaviour at the stability limits, a modified Dugoff model was proposed in [30], able to achieve a behaviour comparable to that of the Magic Formula tyre model with a lower computational load. As this work also seeks to investigate the tyre slip angle with respect to changes in lateral dynamics, invertibility of the tyre model is required. The Magic Formula tyre model and the modified Dugoff are not bijective functions, hence they are not invertible. On the other hand, the original Dugoff curve shape recalls that of a bijective function: if an invertible equation can be conjured to fit that curve, tyre slip angle retrieval is ensured. For this reason, as well as to retain the advantages of its simple formulation, the original Dugoff model is used within this thesis.

The longitudinal tyre force F_x and lateral tyre force F_y as defined in the Dugoff model are detailed in Eq. (2.6). Both forces are proportional to the longitudinal slip ratio (s_l) and lateral slip ratio (s_s), respectively defined in Eq. (2.9) and Eq. (2.10). Slip ratios are functions of the wheel rotational speed ω, its effective radius R_w and the wheel travel longitudinal and lateral speeds, respectively v_x and v_y. Proportionality occurs through the longitudinal tyre stiffness C_λ and the cornering tyre stiffness C_α. Other than the independent definition for longitudinal and lateral stiffnesses, the combined load assumption is dealt with through $f(\xi)$, which distributes the total available friction force depending on the normal load F_z, a constant approximation of the maximum friction coefficient μ_{max}, and the effect of the combined slip ratios (Eq. 2.8).

$$F_x = C_\lambda \frac{s_l}{1+s_l} f(\xi), \qquad F_y = C_\alpha \frac{\tan(s_s)}{1+s_l} f(\xi) \tag{2.6}$$

$$\text{where} \quad f(\xi) = \begin{cases} (2-\xi)\xi, & \xi < 1 \\ 1, & \xi \geq 1 \end{cases} \tag{2.7}$$

$$\text{with} \quad \xi = \frac{\mu_{max} F_z (1+s_l)}{2\sqrt{(C_\lambda s_l)^2 + (C_\alpha \tan(s_s))^2}} \tag{2.8}$$

$$\text{where in turn} \quad s_s = \begin{cases} \dfrac{v_y}{\omega R_w}, & \text{if driving} \\ \dfrac{v_y}{v_x}, & \text{if braking} \end{cases} \tag{2.9}$$

$$\text{and} \quad s_l = \begin{cases} \dfrac{\omega R_w - v_x}{\omega R_w}, & \text{if driving} \\ \dfrac{\omega R_w - v_x}{v_x}, & \text{if braking} \end{cases} \tag{2.10}$$

This body of work will make use of both the Dugoff model (in the outlined formulation) and another model, labelled root-rational tyre model and presented in Section 3.4.1, that closely mimics the former model while catering to the need of inverting the equations to closely monitor angles rather than forces.

2.4 Notions of stability

At its core, the concept of stability encapsulates the ability of any system to maintain predictable motion throughout a wide range of scenarios. A stable system is prone to controlled behaviour, and to resist tendencies towards unstable deviations due to external perturbations. When tackling vehicle stability, one must bear in mind a key notion: capturing the lateral dynamics of a vehicle inherently means representing the latter as a dynamic system and studying its evolution in time.

In essence, the single-track model in Section 2.2 is described using two dynamical equations, capturing time evolutions of both the yaw rate and the sideslip angle. A sensible first step towards stability assessment using the latter model is analysing its steady-state behaviour, which means studying the system when changes in both states of interest are zero. Different approaches exist within the literature to address the matter; however, the two most widely acknowledged ones are based respectively on the handling diagram and phase-planes.

2.4.1 The handling diagram

The handling diagram is a valuable tool in refining dynamics of any vehicle to ensure safe and predictable behaviour during cornering [31]. It displays the response of the vehicle, in terms of steady-state lateral acceleration a_y, prompted by variations of the steering angle δ. A representative portrayal of a handling diagram is provided in Fig. 2.7, where additional quantities have been highlighted following the work in [32], because they are descriptive of the understeering behaviour:

- K_{lin} is the understeering gradient, regulating the linear proportionality existing between δ_{dyn} and a_y in mild cornering scenarios.
- a_y^* represents the limit lateral acceleration for the above-mentioned linear behaviour. This is the theoretical threshold that marks the beginning of the non-linear tyre behaviour, affecting in turn the linearity of the relationship between δ_{dyn} and a_y.
- $a_{y,max}$ is the maximum (asymptotically) achievable lateral acceleration.

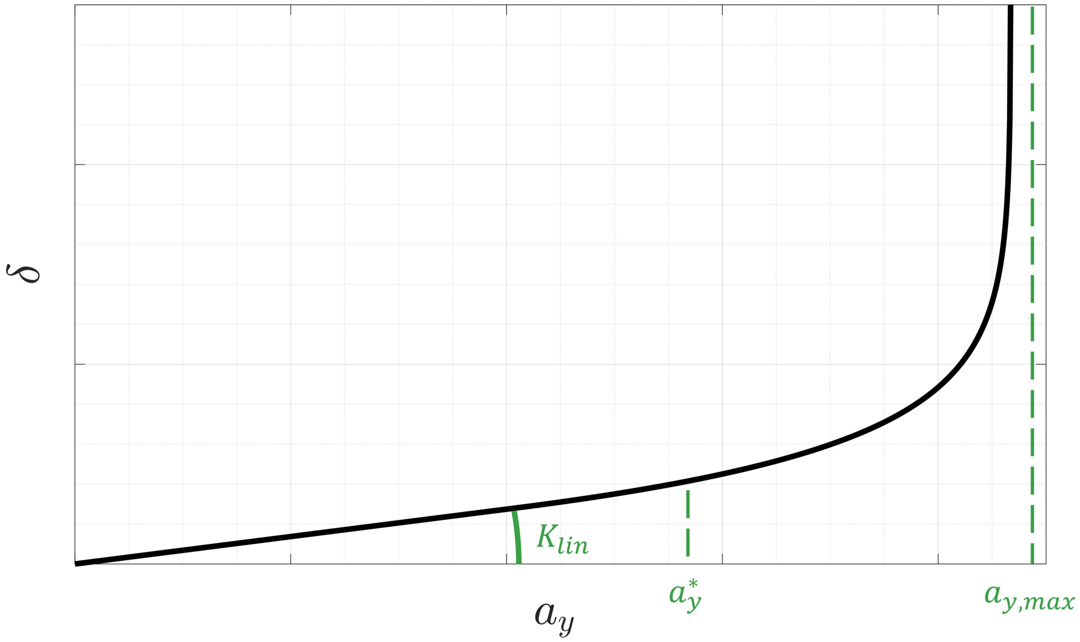
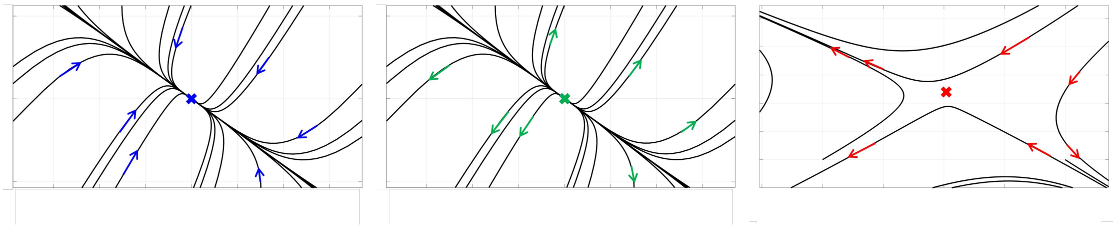


Figure 2.7: Representative depiction of a handling diagram, including the understeering gradient K_{lin}, the linear limit lateral acceleration a_y^* and the maximum asymptotically achievable lateral acceleration $a_{y,max}$.

Although handling diagrams provide a straightforward understanding of the vehicle cornering response, recent discussions questioned their unequivocal applicability. In [21], Guiggiani claims that the handling diagram only offers a partial overview of the complex, multifaceted concept of vehicle stability, and that capturing the vehicle behaviour entails a much broader study of vehicle states of interest. An alternative to the handling diagram is proposed in [21], aiming to overcome the limits of the latter by reverting to a multi-dimensional approach under the name of Maps of Achievable Performance (MAPs), able to depict the attainable operational domain for any state of interest, hence characterising the vehicle cornering behaviour from multiple perspectives.

2.4.2 The phase-plane method

The phase-plane method is particularly attractive, as it involves plotting system state variables against each other on a two-dimensional plane, creating a visual representation of the system's evolution in time. The resulting trajectory, often referred to as a phase trajectory, provides insights into the system's stability. By examining the shape and characteristics of the trajectory, equilibrium points of the system may be discerned, and stability regions may be identified. Stable systems typically exhibit convergence towards a fixed point, called stable node (Fig. 2.8a), while unstable systems may display divergent trajectories originating from unstable nodes (Fig. 2.8b) or saddle points (Fig. 2.8c).



(a) Stable node. **(b)** Unstable node. **(c)** Saddle point.

Figure 2.8: Standard appearance of relevant equilibria types.

The phase plane method is particularly useful in analysing system non-linearities like those exhibited by vehicles close to cornering-induced instability. As emphasised in [33], phase plane analysis has substantial significance in gaining qualitative insights into complex non-linear systems, offering a graphical tool for stability assessment. This attribute makes them a valuable tool for synthesising vehicle control strategies. Bobier-Tiu *et al.* [34] leverage phase portraits to devise control strategies for vehicles, using planar dynamics for a more accessible and intuitive analysis.

An effective control strategy requires an equally effective choice of states to be displayed. One existing approach in the literature favours portraying the sideslip angle against its rate of change ($\beta - \dot{\beta}$), aiming to enhance robustness towards dynamic changes ([35]) and to facilitate a straightforward delineation of the stability region, as argued by Inagaki *et al.* ([36]). A second approach instead favours displaying the sideslip angle against the yaw rate ($\beta - r$). As an example, the work in [34] sets the problem in a more pragmatic frame, since the yaw rate is both typically measured using common on-board sensors and easily controlled using readily-available actuators. This thesis aims to operate with real-time control, hence the latter approach was adopted due to the existing evidence supporting this type of application.

However, the sideslip angle (β) is not as easy to measure as the yaw rate (r). Sensors devoted to that task are available but they are very expensive and sophisticated, hence require an effort not suitable for large-scale distribution on vehicles. Consequently, estimation techniques are frequently used to retrieve the sideslip angle. A comprehensive review on state estimation is presented in Section 2.5.

2.5 State estimation

Knowledge of vehicle states is of paramount importance to enable a comprehensive understanding of the vehicle behaviour. When investigating lateral stability, constant monitoring of insightful states plays a pivotal role in ensuring safety, stability, and optimal performance. However vital, signal acquisition through measuring is not always a straightforward task: Section 2.4 highlighted the practical unfeasibility of monitoring the sideslip angle through the use of dedicated sensors due to the related costs and complexity. To eliminate the need for specific measuring instruments, an attractive substitute has been found within the vehicle stability-related literature in state estimation.

Estimation helps bridge the gap between sensor measurements and the precise knowledge required for effective system control. Its strength resides in the simplicity of the principle behind it: knowledge of easy-to-measure quantities is coupled with mathematical models to determine the current state of a dynamic system. Perhaps the most acclaimed estimation technique existing in the literature is the Kalman filter [37]. The Kalman filter is an iterative algorithm: it uses combined knowledge of real-time measurements and mathematical models to develop a prediction of the vehicle state. Then, using incoming measurements from the next iteration, it aims to minimise the difference between predicted and observed values by tweaking the estimate value based on its relative uncertainty. By doing so, the Kalman filter estimates the state of a dynamic system by operating a balance between the inherent modelling uncertainty and the precision of sensor data. In its original formulation, the Kalman filter only guarantees optimality if the mathematical model at hand is made of linear equations. Given that the scenario of interest entails dealing with the highly non-linear vehicle cornering behaviour, a linear portrayal of the vehicle could be a detrimental choice.

The application under scrutiny relies on data acquisition, which is the operation of collecting measurements at consecutive, equally-spaced instants of time. This means that the computations at hand will be performed in the discrete time domain. This last characteristic enables at each time instant to linearise the model around the latest available state estimate. This procedure, as described in [38], represents a viable solution for non-linear models, and sets the ground for the well-known Extended Kalman Filter (EKF). Due to its versatility and ability to handle non-linear systems, the EKF finds large application in the literature. In [39] an EKF is assembled to estimate lateral tyre forces and use them to fit a tyre model. The work in [40] targets sideslip angle estimation through an EKF designed for heavy-duty vehicles.

Despite the ease of adaptability and relative simplicity of the EKF, this technique also presents a number of drawbacks. Linearisation may introduce errors and can be computationally intensive, especially if the underlying system is highly non-linear. Inaccurate linearisation of system dynamics and measurement models can lead to suboptimal performance, since the search for the optimal estimate may end up being trapped in a local minimum rather than reaching a global one. Moreover, the performance of the EKF is sensitive to the choice of initial conditions, with poor initial estimates potentially resulting in divergence, or convergence to incorrect solutions. Many of these weaknesses arise because of Kalman filtering relying on a model of the vehicle, relying on non-linear equations that require complex algebraic manipulations to fulfil the estimation purposes.

To deal with these issues, more recently new estimation techniques that do not require resorting to a model have been proposed in the literature. Machine learning-based algorithms, including neural networks, make exceptional candidates to estimate vehicle states. In addition to not requiring a model to function, they are able to establish a purely numerical correspondence between provided inputs and their target output, by making use of a measured equivalent of the latter in their training phase.

In Section 4.1, a neural network-based estimator is proposed for the sideslip angle β, exploiting carefully-chosen inputs to avoid overfitting to the training data and promoting easy adaptability to different data. With the purpose of overcoming the training effort that using a neural network entails, a parametric approach to tackle the estimation problem is proposed in Section 4.2, featuring a straightforward interpolation that only uses knowledge of the lateral acceleration a_y and the steering angle δ. A real-time test of this application was also successfully performed, whose outcomes are reported in Chapter 6.

2.6 Torque vectoring

The possibility of harnessing the potential of vehicle states knowledge for the purpose of control was underlined in Section 2.4. When referring to lateral dynamics, the literature often resorts to yaw control, achieved by considering an additional direct yaw moment M_z in Eq. (2.2b). Controlling M_z can lead to a significant stability and manoeuvrability improvement. This results in a safety enhancement, since it enables the vehicle to respond quickly and accurately to driver inputs and, in turn, avoid driving conditions that put the safety of the driver and potential other passengers at stake.

One of the most prominent technologies to achieve direct yaw moment control is torque vectoring, which consists of allocating torques to individual wheels. Torque vectoring has garnered substantial attention recently, mainly due to it constituting a gateway to a superior vehicle performance in electric vehicles, as highlighted in [41]. Mangia *et al.* [42] aim to optimise energy distribution within the electric vehicle by integrating two operational modes, relying on energy efficiency and handling respectively, in a torque-vectoring control framework. The work in [43] instead transcends traditional applications and delves deeper in a less conventional application, targeting performance enhancement for controlled drifting scenarios, hence considering high-sideslip manoeuvres.

In this thesis, the torque vectoring potential is limited to the sole braking action on the individual wheels, aiming to attain an improvement in both performance and safety by prioritising one or the other, judging from the stability assessment performed at each time iteration.

Chapter 3

Framework description

As stated in Chapter 1, this thesis aims to propose a validation workflow for a generic lateral stability controller, by exploiting individual torque allocation. The controller fits within a whole validation framework, including a vehicle model, defined to reproduce suitably the expected vehicle response. The overall control scheme is presented in Fig. 3.1: due to the overwhelming number of featured blocks, a short content breakdown follows.

A driver will provide a speed and steering angle profile to both the vehicle model, which will respond with a change of its states, and to a reference generator, in charge of creating an appropriate, desired behaviour to be followed by the vehicle. The actual and desired behaviour will then be compared and, on the basis of the magnitude of their discrepancy (called error) an appropriate control action will be generated by the high-level controller to affect the vehicle behaviour. The low-level controller will then translate it into suitable actuator inputs to be fed to the vehicle, for the latter to follow the desired behaviour.

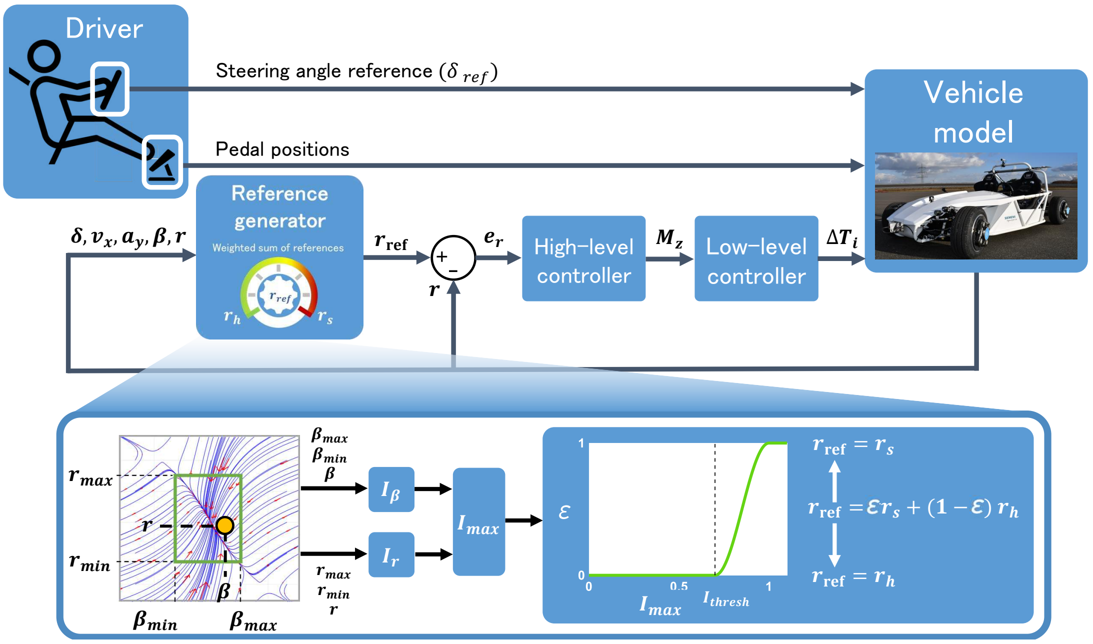


Figure 3.1: Control scheme overview.

A more in-depth description of the individual components of Fig. 3.1 is provided in the following sections.

3.1 Driver model

In a generic real driving scenario, vehicle motion is initiated when the driver imposes a set of inputs to the vehicle itself. Shifting the focus back to simulation, motion commencement will not actually differ greatly from the above-described scenario: the vehicle will move in response to an imposed driving profile, based on longitudinal speed v_x and steering angle δ. A throttle regulator within Amesim ® will then be in charge of converting these signals into pedal positions, ultimately translated into wheel torques and realised by actuators, i.e. in-wheel motors.

The driver has unlimited capability of scenario generation, provided that the desired actions stay within the physical limits of both steering angle and pedal motion, as well as respecting the motion limits of the overall vehicle. However, with the goal being stability assessment, a more accurate selection of manoeuvres is required, able to specifically challenge the vehicle lateral dynamics. Another problem is that the chosen architecture for the driver, despite its simplicity, does not allow to deal with trajectory-based manoeuvres but only with input-based ones: this automatically excludes the celebrated double lane-change manoeuvre [44]. For this reason, a single lane-change manoeuvre is preferred.

To carry out thorough vehicle testing, the same manoeuvre has to be performed in increasingly more challenging driving scenarios to assess the vehicle stability limits. In this work two scenarios, corresponding to two sets of inputs, are envisioned:

- **Mild**: ordinary operating condition. Designed for the control strategy to allow the driver complete handling freedom.

- **Challenging**: lateral stability is investigated closer to its limits. Designed for the control strategy to intervene when a safety-concerning proximity to stability limits is detected.

Both sets of inputs are presented in Fig. 3.2. In the mild scenario, a sinusoidal steering action with an amplitude of 50° is performed as the vehicle cruises at a speed of 25 m/s, while the challenging scenario features the same steering input while the speed is still increasing until 40 m/s: this is done in order to induce an unstable condition in the vehicle. In both cases, the steering action is initiated after 10 s and lasts for a total of 2 s.

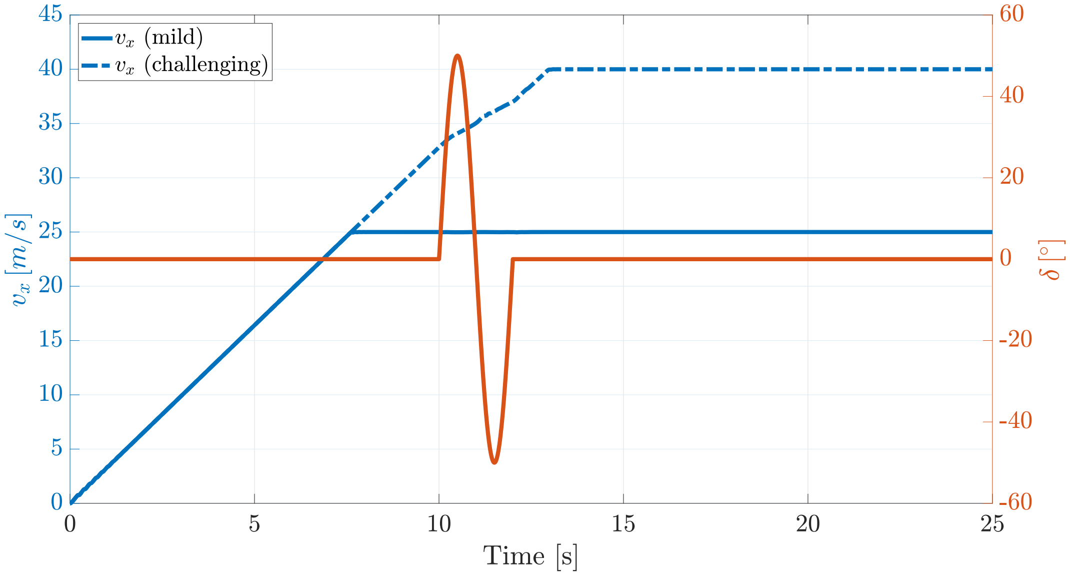


Figure 3.2: Single-lane manoeuvre inputs, both for mild and challenging scenarios.

To summarise, the actions initiating vehicle motion are carried out by a driver model, i.e. a set of signals mimicking the inputs that would normally be fed to the vehicle by a human pilot. Two single lane-change scenarios have been crafted to showcase the performance of the controller respectively in a mild scenario as well as one challenging the lateral stability of the vehicle.

3.2 Vehicle model

As evidenced in Chapter 1, to cater to the testing needs of the controller an appropriate platform has to be devised, containing a model that represents closely the actual vehicle behaviour while trading off accuracy with a manageable complexity. For ease of initiation of the testing activities, a single-track model was chosen (Section 2.2), due to it being described by two differential equations while still offering a fairly good portrayal of the lateral dynamics of the vehicle. The simplicity of the single-track model is extremely beneficial to facilitate preliminary tuning of the controller parameters, and ensure compliance to the performance goals set in the design phase.

The single-track model only acceptably mimics a realistic vehicle behaviour when it operates in linear regime [21]: however, phenomena related to loss of stability start to occur at the handling limits, i.e. when the vehicle exhibits an increasingly stronger non-linear behaviour. To avoid an inaccurate portrayal of the vehicle in the operating conditions that are most relevant for the stated purposes, after tuning the controller the test bench was changed to a 15 DOF model.

The 15 DOF model captures the vehicle behaviour using a multibody chassis model, composed of a main car body, four wheels, four spindles and a steering rack. Each body normally has six degrees of freedom, namely displacement along and rotation about a three-dimensional coordinate system. However, kinematic constraints restrict the relative motion of the bodies. As evidenced in Fig. 3.3:

- Relative motion between the car body and individual spindles only occurs through vertical displacement of the latter. The relative degree of freedom is indicated as Z_w and is observed on all four spindles (4 DOF).

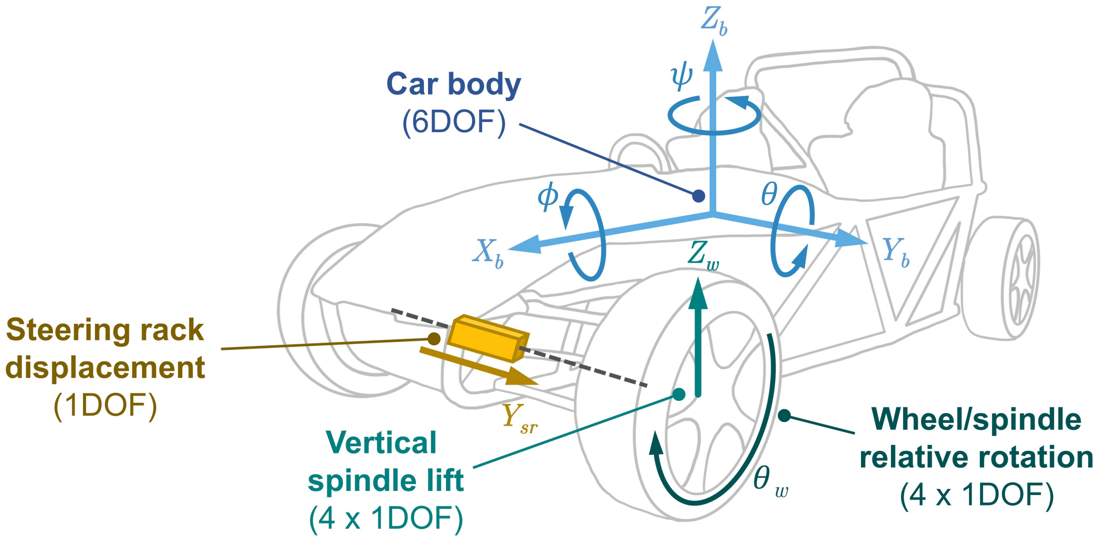


Figure 3.3: Degrees of freedom for a 15DOF model. For simplicity, wheels and spindles-relative degrees of freedom are shown for the front left wheel only, but they are present at all four wheel/spindle joints.

- Wheels and spindles are constrained through revolute joints, as their only relative motion is the wheel rotation. Their relative rotation angle is indicated as θ_w, which is present between all four wheel/spindle joints (4 DOF).

- The steering rack translates along the car body on one axis only, hence they are constrained through a prismatic joint. Their relative displacement is exemplified by the translational degree of freedom Y_{sr} (1 DOF).

Summed up to the 6 degrees of freedom of the car body, which translates along and rotates about the axes X_b, Y_b, Z_b (Fig. 3.3), a total of 15 DOF is obtained.

This number of degrees of freedom allows a much higher level of detail within simulation, including aerodynamic phenomena, effect of suspensions and dependency on road conditions among others. The presence of virtual sensors is particularly of interest, because it allows retrieving a virtual measurement from the vehicle model as if it were sourced from a sensor on the real vehicle.

Both the single-track and the 15 DOF model are based on the SimRod battery electric vehicle [45] (Figure 3.4), as made available by Siemens ®. Parameters for the single-track model are reported in Table 3.1.

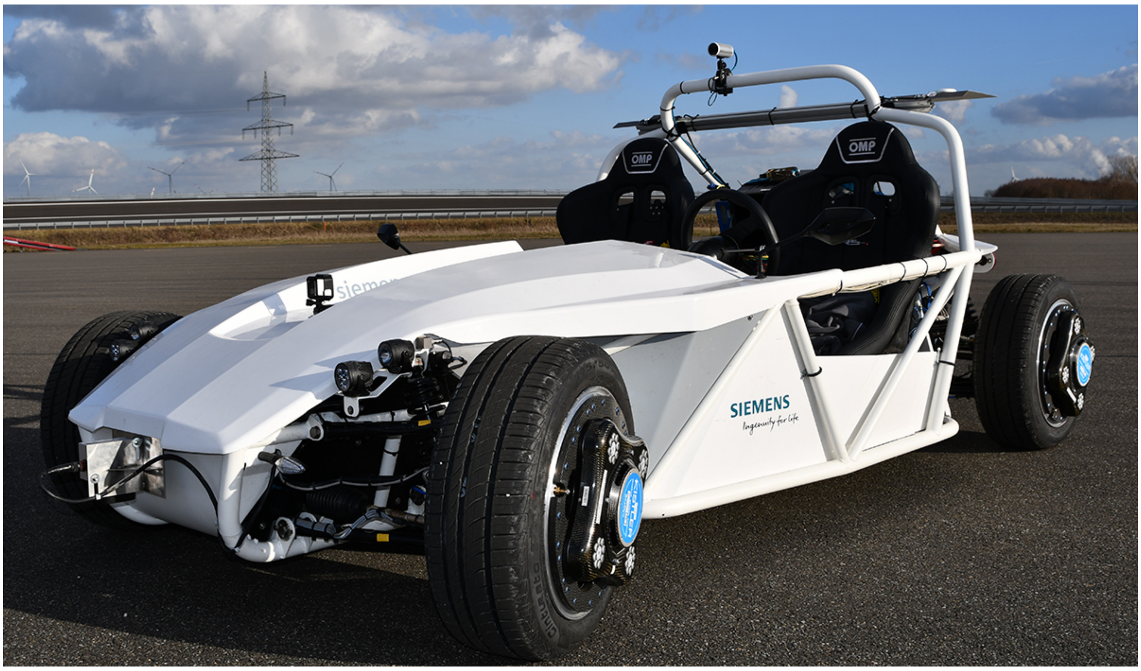


Figure 3.4: SimRod batttery electric vehicle (Source: Siemens Digital Industries Software ®).

Table 3.1: Vehicle parameters.

Parameter	Description	Value	Unit
m	Total vehicle mass	860	kg
l	Vehicle wheelbase	2.335	m
a_1	Distance of front axle to CoG	1.171	m
a_2	Distance of rear axle to CoG	1.164	m
t_w	Track width	1.428	m
h_{CoG}	Height of CoG from ground	0.1	m
J_z	Yaw inertia	700	$\text{kg} \cdot \text{m}\ ^2$

3.2.1 Tyre model

The Dugoff tyre model (Section 2.3) was firstly defined and coupled with the single-track model, then naturally progressed to being paired with the 15 DOF vehicle model as they run concurrently in the same framework, which is Amesim ®. Tyre parameters for the application at hand are reported in Table 3.2.

Table 3.2: Tyre parameters.

Parameter	Description	Value	Unit
C_λ	Tyre longitudinal stiffness	37500	N/rad
$C_{\alpha,f}$	Tyre lateral stiffness (front)	37816	N/rad
$C_{\alpha,r}$	Tyre lateral stiffness (rear)	52140	N/rad
μ_{max}	Maximum friction coefficient	1	-
R_w	Effective rolling radius	0.302	m

3.2.2 Stability analysis of vehicle model equations

The differential equations governing the 15 DOF model essentially form a state-space description of the actual vehicle. The accuracy of states throughout the simulation strongly relies on the employed integration algorithm, whose main task is going to be linearising the non-linear model equations. Consequently, an appropriate solver will be chosen based on its linearisation ability.

As detailed in Chapter 2, linearisation is performed by considering a perturbation of the operational state so small in time that it allows to approximate the non-linear system with a linear equation. For time variant systems, the step size in question is the sampling time: in principle, a smaller sampling time should produce a more stable linear approximation because it acts in a small enough neighbourhood of the operating point that the risk of diverging from the true non-linear behaviour is proportionally lower. In reality, a too small sampling time is too computationally demanding for the model to retain real-time assumptions. The definition of a sampling time allows using an iterative integration algorithm, meaning the non-linear model will be approximated between two consecutive time steps.

The simplest method for numerical integration of ordinary differential equations is undoubtedly the Euler method, which consists of breaking down the interval of interest into small steps and iteratively using the derivative at each step to estimate the next value. However simple, the Euler method strongly relies on sampling time: vehicle stability problems deal with very fast state transients, which often leads to them being classified as stiff problems. The Euler method does not guarantee numerical stability for stiff problems unless a very small sampling time is chosen, which is computationally very expensive and could easily lead to lose real-time capability [46].

Due to its intrinsically more stable formulation, second-order Runge-Kutta was chosen as solver algorithm. Runge-Kutta methods allow multiple assessments of the model derivative within the same time step [46]. Second-order Runge-Kutta allows this evaluation to happen twice within one time step: first, it estimates the midpoint of the time interval using the initial derivative, and then it computes the final value by taking a weighted average of the initial and midpoint derivatives. Because of its two-step integration scheme, second-order Runge-Kutta is more stable and accurate than the Euler method. Runge-Kutta algorithms have more accurate formulations up to the fourth order. However, for many practical problems the accuracy of second-order Runge-Kutta is adequate, and the additional computational cost of fourth-order Runge-Kutta may not result in a proportionate gain in accuracy, particularly when the solution is not highly sensitive to small step sizes [47]. Second-order Runge-Kutta is then chosen to keep the computational complexity manageable while ensuring adequate accuracy.

With the aims of both demonstrating model convergence and establishing a solid sampling time for the solver, a stability analysis is conducted on the model at hand. To simplify the task, the non-linear model is replaced with a linear approximation in the proximity of given operating points. This allows to study the second-order model dynamics and assess how small perturbations from said points will affect the model response.

Stability around an operating point is determined from system eigenvalues. Amesim ® allows to compute and display system eigenvalues as well as the system stability region on the Argand plane: if all the system eigenvalues fall within the stability region, system stability is guaranteed. A more accurate linearisation gives more valuable insight on the true non-linear system stability: for this reason, eigenvalues are studied at different instances corresponding to different sampling times and operating scenarios. Observing changes in the pole location throughout such studied cases, it appears that a sampling time higher than 5.67 ms causes one of the computed poles to cross the boundaries of the stability region. A sampling time of 1 ms was hence chosen for the vehicle model.

3.3 Vehicle dynamics controller

Bearing in mind that the proposed validation workflow aims to assess the correct functioning of a vehicle stability controller, the controller itself is the core element of the framework: it is in charge of regulating the actual vehicle behaviour in order for it to match as closely as possible a reference (desired) one, based on some established performance requirements. Carrying out the desired control action entails the definition of both a high-level controller, computing the control action itself, and a low-level controller, transforming the control action into inputs for the available actuators. A detailed description of both follows.

3.3.1 High-level controller

Vehicle instability is often related to friction loss phenomena, which happen in a highly non-linear regime: an appropriate control action must be devised to govern the vehicle especially in the non-linear domain to ensure no unexpected divergent behaviour is imparted on the vehicle. For this to be possible, a quantity characterising the vehicle behaviour (a so-called state, as introduced in Chapter 1) has to be picked as control objective. The control action will then affect the vehicle by operating a certain algorithm to track a reference defined for that control objective. Given that this is a lateral stability controller, the control objective must be both an insightful stability indicator and easily sourced from the vehicle. The latter implies both that the signal is available in real time, and that it can be reliably obtained. Among the common solutions discussed in the literature the yaw rate is perhaps the most popular, because it is measured using the well-established technology of Inertial Measurement Units (IMUs). Given that the comparison and tracking rely almost entirely on the knowledge of the actual state, the yaw rate is the control variable of choice for the proposed controller as well.

Numerous control methods are suitable for the task at hand, but they are usually too computationally complex to handle real-time applications (Chapter 2). Looking ahead at the timeline and feasibility of the whole validation plan, it is wiser to pick a simple control strategy to guarantee real-time requirements are successfully met. Provided they are, there will be room for further investigation on applicability of other control methods that are more computationally demanding. Given that the primary performance requirement is to maintain stable and accurate tracking of a single variable (i.e. the yaw rate), a simple control strategy may be devised for the task. A proportional-integral (PI) controller is selected, as its simplicity and lower computational demands make it particularly well-suited for real-time implementations. PI controllers aim to minimise the difference (or error) between the reference and measured control variable (yaw rate in this case) by means of a proporional term, providing an immediate response to the current error, and an integral one, eliminating any remaining steady-state error by considering the accumulated past errors: this level of simplicity is often favoured in industrial applications [48]. The PI controller output is chosen to be the vehicle direct yaw moment M_z, whose regulation allows to significantly improve vehicle stability and manoeuvrability (Section 2.6). The PI control law is written in Eq. (3.1), where K_p represents the proportional gain, K_i is the integral gain and e is the error, hence the difference between the reference yaw rate (r_{ref}) and the actual yaw rate (r).

$$M_z = K_p \cdot e + K_i \cdot \int e \, \mathrm{d}t, \quad \text{where} \quad e = r_{\text{ref}} - r \tag{3.1}$$

In order to account for the dynamic changes prompted by the evolution in time of the vehicle operating conditions, while still retaining the design and computational advantages of an inherently simple PI controller, gain scheduling was implemented. Gain scheduling allows the controller to update the PI gains to maintain optimal performance over a wide range of driving scenarios. For the described implementation, a similar tuning procedure to the one detailed in [49] was used. Relevant steps are listed:

1. Consider the single-track model coupled with a linear tyre model, whose constant front and rear cornering stiffnesses C_1 and C_2 are the linear approximation, under the small-angle assumption, of the constitutive curves (Section 2.3). Assume steady-state operating conditions, setting the speed to be constant.

2. Rewrite the control objective, i.e. the yaw rate r, in the Laplace domain: r will be related to the output M_z (the direct yaw moment) through an appropriate gain G_{r,M_z}, and to the input disturbance δ (the steering angle) through a suitable gain $G_{r,\delta}$. The Laplace transform of r is shown in full in Eq. (3.2), where the gains G_{r,M_z} and $G_{r,\delta}$ are retrieved using stability derivatives as defined in [49].

$$r = G_{r,M_z}(s)M_z(s) + G_{r,\delta}(s)\delta(s) \tag{3.2}$$

3. The reference direct yaw moment $M_z(s)$ is computed in Eq. (3.3) by multiplying the difference between the reference ($r_{\text{ref}}(s)$) and the actual ($r(s)$) yaw rate by the controller law $W(s)$ as defined in Eq. (3.4), where K_p and K_i are respectively the proportional and the integral PI gains.

$$M_z(s) = W(s)(r_{\text{ref}}(s) - r(s)) \tag{3.3}$$

$$W(s) = K_p + \frac{K_i}{s} \tag{3.4}$$

Table 3.3: Closed-loop gain scheduling coefficients corresponding to specific speeds, along with resulting bandwidth (targeting 1.6 Hz).

Speed (m/s)	K_p (Nms/rad)	K_i (Nm/rad)	Bandwidth (Hz)
5	18645	26000	1.6010
10	13264	26000	1.6001
15	8519	26000	1.6003
20	6106	26000	1.6000
25	4549	26000	1.6000

4. To simplify the gain tuning, K_i is kept constant while K_p is changed in order to ensure the frequency response of the vehicle model has the same magnitude for a wide range of speeds. The vehicle model is a closed-loop system, hence its frequency behaviour is dictated by its tracking bandwidth. The chosen bandwidth value should guarantee good reference tracking while still ensuring vehicle driveability. Following the reasoning in [49], the tracking bandwidth is set to 1.6Hz, which is the frequency at which the closed-loop system response magnitude drops of $-6\,\text{dB}$. The proportional gains are accordingly tuned for five different speeds in the range 5 m/s - 25 m/s. The proportional and integral gains K_p and K_i as well as the resulting bandwidth for each selected speed are reported in Table 3.3. The Bode plots showing the resulting closed-loop system response are reported in Fig. 3.5.

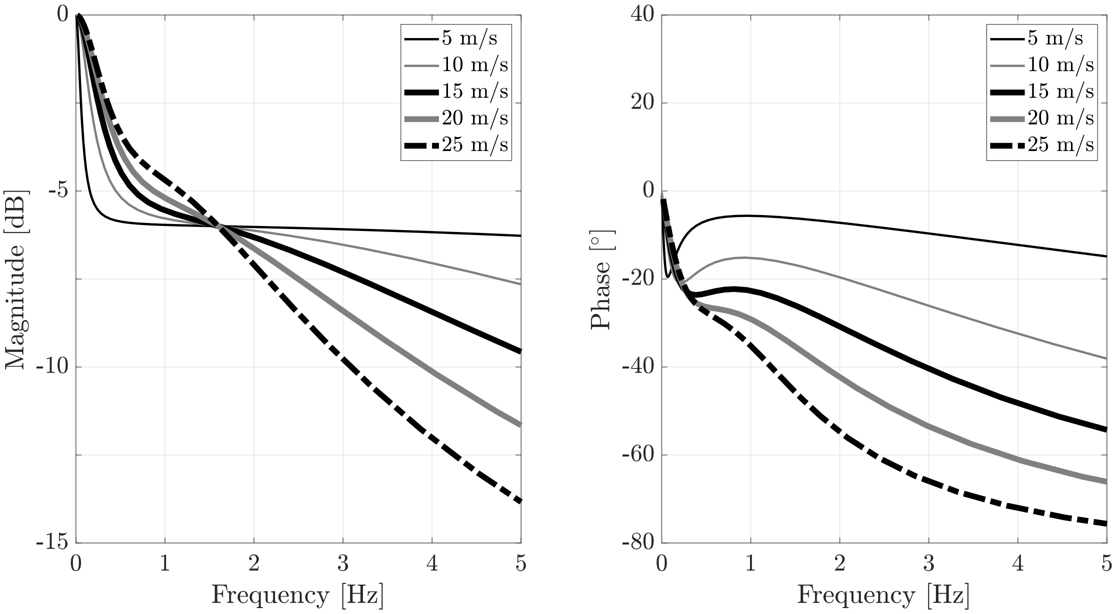


Figure 3.5: Bode plots of the closed-loop system for speeds listed in Table 3.3.

5. Controller gains for interstitial speed values were computed by means of interpolation, and introduced in the controller scheme by means of a lookup table with a single input (the vehicle speed), for ease of computation and usage from the controller itself.

3.3.2 Low-level controller

Following from what was stated earlier in this Section, the controller will first compare the expected yaw rate behaviour to the actual one measured on the vehicle and then manipulate their difference (error) to close the gap between the two. The controller output, the direct yaw moment M_z, is the moment about the yaw axis that would realign the vehicle to follow the provided reference behaviour. Thinking of a real scenario however, it becomes immediately evident that the computed direct yaw moment cannot be imposed on the vehicle as is, but rather an implementation strategy must be conjured, one where actuators are involved. In other words, a low-level controller must be defined. Supposing the vehicle supports individual wheel braking, brake actuators are a perfect fit for the task: as illustrated in Fig. 3.6, a counterclockwise stabilising direct yaw moment can be achieved by braking the left-side wheels. Conversely, a clockwise direct yaw moment is accomplished by braking the right-side wheels. Notice that in the presented sign convention, a counterclockwise moment is positive while a clockwise one is negative.

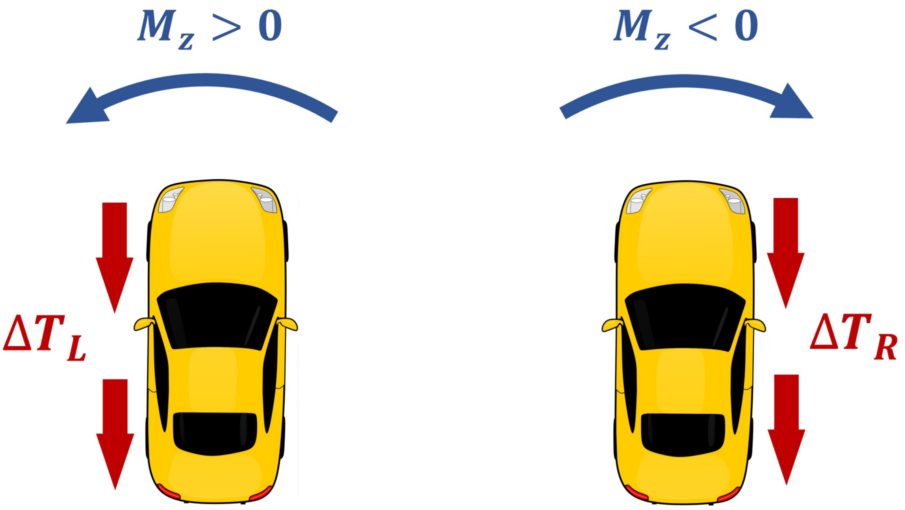


Figure 3.6: Direct yaw moment and corresponding braking torque allocation.

Braking the wheels means transmitting to them a torque effort: to begin with, the torque effort for the left and/or right side is equally split between the front and rear wheel for simplicity. The torque allocation is hence performed following Eq. (3.5), where R_w represents the wheel radius, t_w is the vehicle track (front and rear are assumed to be equal) and ΔT_i is the additional torque effort associated to the control action, to be then split equally between front and rear wheel of the left or right side.

$$\Delta T_i = \begin{cases} 2\dfrac{M_z R_w}{t_w}, & \text{i = left} \quad \text{if} \quad M_z > 0 \\ -2\dfrac{M_z R_w}{t_w}, & \text{i = right} \quad \text{if} \quad M_z < 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

Besides designing a more meticulous splitting strategy for the torques, an immediate extension of the actuation strategy is towards all-wheel drive vehicles: they would grant also the allocation of positive torques and broaden the domain of torque allocation strategies thanks to their overactuation capabilities.

To summarise, the complete controller structure is illustrated as follows: the difference, or error, between the reference and actual yaw rate is fed to a high-level controller, in charge of applying the control strategy to the incoming signal to generate the needed direct yaw moment M_z according to the control law in Eq. (3.1). Finally, M_z enters the low-level controller and is converted into individual additional torque efforts ΔT_i to be carried out by the brake actuators.

3.4 Reference generator

Section 3.3 explained the devised control strategy to impose a certain desired behaviour to the vehicle, aiming to improve its stability and performance. However, such reference cannot be established a priori, but rather needs to adapt to the current vehicle operating condition. This calls for the need to implement a reference yaw rate generator, able to assess the stability condition of the vehicle and accordingly produce a desired yaw rate, which will be later translated into a direct yaw moment by the controller.

This section contains individual descriptions of the modules constituting the reference generator: a stability assessment is performed at first and discussed in Section 3.4.1, which determines whether the vehicle is in a safe operating condition. Section 3.4.2 introduces two yaw rate references, prioritising respectively safety or handling, as well as a unique referencing strategy merging both. Furthermore, two novel sideslip angle estimation methods are presented: Section 4.1 discusses a neural network-based estimation strategy, while Section 4.2 describes a parametric approach to the same problem.

3.4.1 Stability assessment

The first step towards having a descriptive picture of the stability condition of the vehicle is to represent it as a dynamic system, to then observe its evolution in time: therefore, a rational first step involves studying its steady-state behaviour.

As starting point for the analytic investigation, the single-track model is recalled as defined in Eq. (2.2). The single-track model is a good starting point to have initial insight on the vehicle cornering behaviour: the steady-state assumption is accounted for by considering the steering angle δ and the longitudinal speed v_x as constant inputs of the model. Nevertheless, the dynamic system under scrutiny is notably influenced by non-linearities, primarily stemming from tyre friction phenomena. Among other techniques, identification of non-linear dynamical systems is performed by linearising it around its equilibria and monitoring the system behaviour when its current operational state is located nearby an equilibrium point [50]. Phase portraits (Section 2.4) are an excellent visual tool to study the system dynamics, as they display the evolution in time of one vehicle state against the evolution in time of another vehicle state. Determining the location of equilibria on phase portraits and classifying their nature means being able to outline exactly the vehicle stability region, meaning the set of operating conditions that ensure vehicle safety.

The open-loop dynamics of an understeering vehicle travelling at $v_x = 15\,m/s$ are shown on a $\beta - r$ phase plane in Fig. 3.7, using a steering angle $\delta = 0°$. State trajectories are indicated by means of red arrows, while stable equilibria are visible as yellow dots. One stable equilibrium point is located in the origin of the phase plane and two saddle points are also present, whose coordinates reflect the yaw rate and sideslip angle corresponding to the tyre saturation condition [34]. Although tyre saturation is regarded as an opportunity to achieve controlled drifting, such an application is beyond the scope of this thesis: the saddle points coordinates are hence hereby regarded as limit stability conditions.

Almost the entirety of the literature touching on equilibria location is constituted by numerical methods, which in turn are based on the notion of discretisation. Given somewhat reasonable ranges of values for both states, those sets of values may be split into smaller time intervals: a grid of coordinates is then defined, and each of them is iteratively scanned and replaced in Eq. (2.2). Provided that those intervals are small enough, coordinates that make Eq. (2.2a) - (2.2b) simultaneously equal to zero (or reasonably close to zero) are classified as equilibria. However, this leads to a clear conclusion: the numerical solution is only as accurate as the discretisation allows it

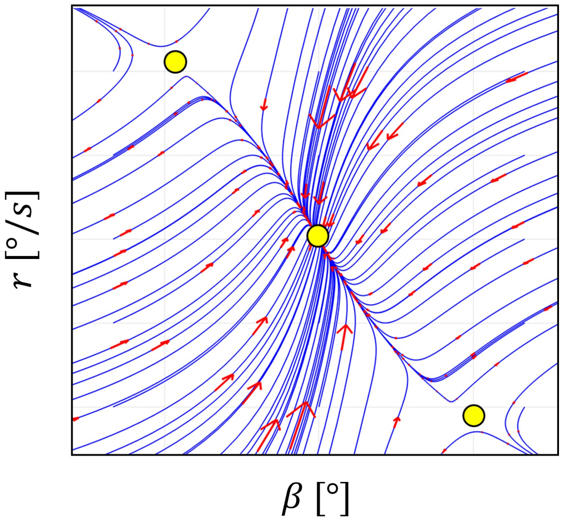


Figure 3.7: Open-loop dynamics of an understeering vehicle on a $\beta - r$ phase plane, using $\delta = 0°$ and travelling at a speed of $v_x = 15\,m/s$. Red arrows indicate state derivatives, hence the evolution in time of the state trajectories. Yellow dots indicate equilibria.

to be. A more dense discretisation, while allowing to refine the search, leads to a larger computational load and still does not guarantee finding an accurate enough solution. As stated in [51], the single-track model equations form a non-linear system that can only be solved numerically, using a constant value for the steering angle as well as for the longitudinal speed. More specifically, the inherent obstacle preventing solving the problem analytically is the use of piecewise-defined, non-invertible tyre models.

A novel approach is hereby proposed, where this limitation is overcome by introducing a root-rational tyre model, essentially constituted by a parametric interpolation of tyre models present in the literature. The function inversion problem is then transformed into a much simpler root location of a third-degree polynomial, and the equilibria are easily retrieved.

The root-rational tyre model

The differential equations describing the single-track model (Eq. 2.2) do not feature the states in explicit form. More precisely, the front and rear axle lateral tyre forces F_{y1} and F_{y2} cannot be substituted further to formulate them as functions of the sideslip angle β. Non-linear tyre models existing in the literature are either piecewise-defined, as the Dugoff model presented in Eq. (2.6), or present trigonometric functions that make the inversion problem analytically intractable, like the well-renowned Magic Formula by Pacejka [28]. Supposing that a tyre model representing an invertible relationship between lateral forces and tyre slip angles exists, the single-track model equations can be analytically solved.

The proposed root-rational tyre model is described in Eq. (3.6), where $k = 180/\pi$ is merely a radians-to-degrees conversion factor. Recalling that the tyre model equations will be interfaced with the single-track model, a constitutive formulation is presented for each axle. According to the chosen sign convention, a negative tyre slip angle corresponds to a negative lateral force: this calls for the need of an odd function, hence the choice of including a square root at the denominator.

$$F_{yi} = \frac{c_{1i} k \alpha_i}{c_{2i}\sqrt{(k\alpha_i)^2 + c_{3i}}} \quad i \in \{f, r\} \tag{3.6}$$

The scope of the root-rational tyre model is to conjure an invertible adaptation of a generic tyre model: this is done by introducing three coefficients per axle, hence a total of six coefficients $c_{1i} \dots c_{3i}$ where $i \in \{f, r\}$, whose choice is dictated by the need to mimic an existing constitutive curve. Coefficients are retrieved by solving a non-linear least squares problem, which is translated to a simple curve fitting optimisation: the objective is to bring the interpolated constitutive curve as close as possible to the original one through the manipulation of available coefficients. Looking at the practical implementation, coefficients have been retrieved by using the MATLAB ® function `lsqcurvefit`, and setting Levenberg-Marquardt as optimisation algorithm.

The search for an optimal solution also relies on the choice of a set of initial conditions. Rather than their exact value, a correct order of magnitude for the initial conditions is needed to successfully interpolate the curve: therefore, initial conditions were manually tuned.

For the vehicle at hand, outcomes of interpolation on the Dugoff constitutive curve are compared to the original Dugoff constitutive plots themselves in Fig. 3.8, both at the front and at the rear axle.

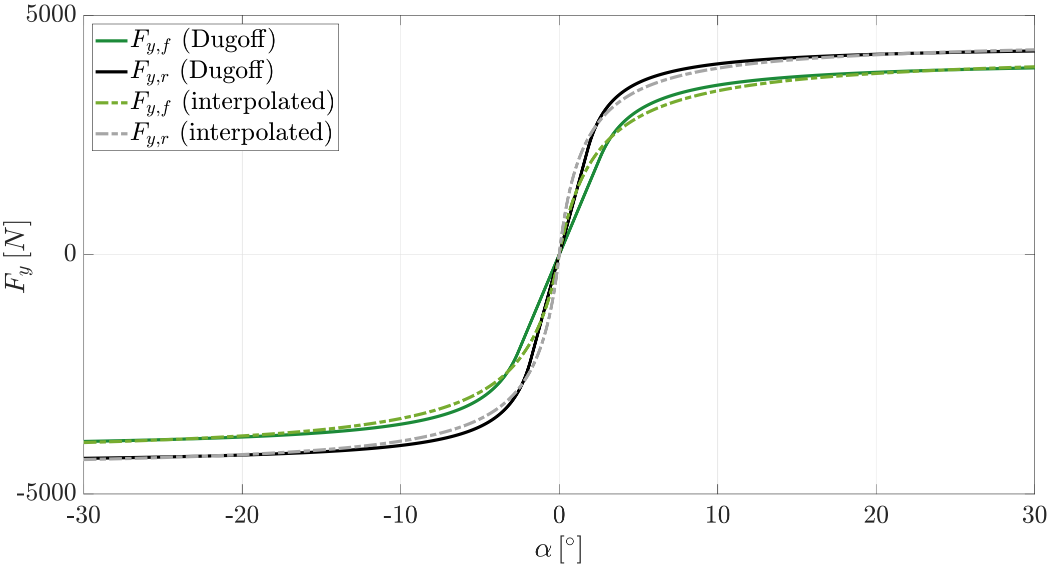


Figure 3.8: Fitting of the new interpolating function compared to original Dugoff constitutive plot, at front and rear axle.

Third-degree polynomial definition

In accordance with the observations in [50] state equations Eq. (2.2a) - (2.2b) are set to zero to locate the equilibria coordinates.

$$\begin{cases} F_{y1} + F_{y2} = m v_x r, & (3.7a) \\ \dfrac{F_{y1} a_1 - F_{y2} a_2}{J_z} = 0 & (3.7b) \end{cases}$$

Isolating F_{y2} from Eq. (3.7a) and substituting the result in Eq. (3.7b) leads to Eq. (3.8):

$$F_{y1} a_1 + F_{y1} a_2 - m v_x r a_2 = 0 \tag{3.8}$$

which in turn becomes

$$F_{y1} l = m v_x r a_2 \tag{3.9}$$

The rational tyre model for the front axle in Eq. (3.6) is then substituted in place of F_{y1}, generating Eq. (3.10):

$$\frac{c_{1f}\alpha_f kl}{c_{2f}\sqrt{(k\alpha_f)^2 + c_{3f}}} = mv_x r a_2 \tag{3.10}$$

Due to it being the argument of a square root, the tyre slip angle can only be isolated under the hypothesis that a domain has been declared alongside the solution:

$$\alpha_f = \begin{cases} \dfrac{mv_x r a_2 c_{3f}}{k(c_{1f}l - mv_x r a_2 c_{2f})} & \text{if} \quad \alpha_f \geq 0 \qquad (3.11a) \\[2ex] \dfrac{mv_x r a_2 c_{3f}}{k(c_{1f}l + mv_x r a_2 c_{2f})} & \text{if} \quad \alpha_f < 0 \qquad (3.11b) \end{cases}$$

Sourcing F_{y1} from Eq. (3.7a), a similar result can be obtained for the rear tyre slip angle:

$$\alpha_r = \begin{cases} \dfrac{mv_x r a_1 c_{3r}}{k(c_{1r}l - mv_x r a_1 c_{2r})} & \text{if} \quad \alpha_r \geq 0 \qquad (3.12a) \\[2ex] \dfrac{mv_x r a_1 c_{3r}}{k(c_{1r}l + mv_x r a_1 c_{2r})} & \text{if} \quad \alpha_r < 0 \qquad (3.12b) \end{cases}$$

The tyre slip angles have been successfully isolated: the next step is using the set of equations in Eq. (2.4) to obtain an explicit expression relating the single-track model equations to the vehicle sideslip angle β.

The denominator in Eq. (3.6) is always bigger than zero, hence the lateral force sign appears to be dictated by the numerator. Furthermore, considering Eq. (3.7b) and given that the front and rear CoG-to-axle distances a_1 and a_2 are both greater than zero, for the equilibrium condition to be satisfied, the two lateral forces must have the same sign. Strictly in a physical sense, it is possible that the front and rear tyre slip angle, hence the front and rear lateral forces, have opposite signs: however, this cannot mathematically happen when looking for equilibria, which inherently means that case scenarios where the tyre slip angles have opposite signs are immediately discarded. The remaining two cases are explored:

1. $\boldsymbol{\alpha_f \geq 0}$, $\boldsymbol{\alpha_r \geq 0}$. Subject to the boundary constraints in Eq. (3.13), Eq. (2.4) can be written in compact form as Eq. (3.14)

$$\begin{cases} v_x \neq 0 & \text{(3.13a)} \\ c_{1f} \neq \dfrac{mv_x r a_2 c_{2f}}{l} & \text{(3.13b)} \\ c_{1r} \neq \dfrac{mv_x r a_1 c_{2r}}{l} & \text{(3.13c)} \end{cases}$$

$$\begin{cases} \beta = \delta - \dfrac{ra_1}{v_x} - \dfrac{mv_x r a_2 c_{3f}}{k(c_{1f}l - mv_x r a_2 c_{2f})} \\ \beta = \dfrac{ra_2}{v_x} - \dfrac{mv_x r a_1 c_{3r}}{k(c_{1r}l - mv_x r a_1 c_{2r})} \end{cases} \tag{3.14}$$

Subsequently, the system can be solved by means of the elimination method and a unique expression, reported in Eq. (3.15), can be obtained, more specifically a third-degree polynomial made of a number of parameters and a unique variable: the yaw rate r.

$$\begin{aligned} &[-m^2v_x^2a_1a_2c_{2f}c_{2r}lk]r^3 + mv_x[mv_x^2a_1a_2(c_{3f}c_{2r} - c_{2f}c_{3r}+ \\ &+ k\delta c_{2f}c_{2r}) + l^2k(a_1c_{2r}c_{1f} + a_2c_{2f}c_{1r})]r^2+ \\ &+ l[mv_x^2(a_1c_{3r}c_{1f} - a_2c_{3f}c_{1r}) - mv_xk\delta(a_1c_{2r}c_{1f}+ \\ &+ a_2c_{2f}c_{1r}) - c_{1f}c_{1r}l^2k]r + c_{1f}c_{1r}l^2k\delta = 0 \end{aligned} \tag{3.15}$$

After a few mathematical manipulations, Eq. (3.15) reaches the more compact form presented in Eq. (3.16)

$$A_pr^3 + B_pr^2 + C_pr + D_p = 0; \tag{3.16}$$

whose coefficients are:

$$\begin{aligned}
&A_p = -m^2 v_x^2 a_1 a_2 c_{2f} c_{2r} l k;\\
&B_p = m v_x [m v_x^2 a_1 a_2 (c_{3f} c_{2r} - c_{2f} c_{3r} + k\delta c_{2f} c_{2r}) + k l^2 (a_1 c_{2r} c_{1f} + a_2 c_{2f} c_{1r})];\\
&C_p = l\{m v_x^2 [a_1 c_{3r} c_{1f} - a_2 c_{3f} c_{1r} - k\delta (a_1 c_{2r} c_{1f} + a_2 c_{2f} c_{1r})] - c_{1f} c_{1r} l^2 k\};\\
&D_p = c_{1f} c_{1r} l^2 k \delta v_x;
\end{aligned} \tag{3.17}$$

2. $\boldsymbol{\alpha_f < 0}$, $\boldsymbol{\alpha_r < 0}$. Following the same reasoning, a compact form is reached for the rear axis as well, leading to Eq. (3.18), whose coefficients are clarified in Eq. (3.19).

$$A_n r^3 + B_n r^2 + C_n r + D_n = 0; \tag{3.18}$$

$$\begin{aligned}
&A_n = -m^2 v_x^2 a_1 a_2 c_{2f} c_{2r} l k;\\
&B_n = m v_x [m v_x^2 a_1 a_2 (c_{2f} c_{3r} - c_{3f} c_{2r} + k\delta c_{2f} c_{2r}) - k l^2 (a_1 c_{2r} c_{1f} + a_2 c_{2f} c_{1r})];\\
&C_n = l\{m v_x^2 [a_1 c_{3r} c_{1f} - a_2 c_{3f} c_{1r} + k\delta (a_1 c_{2r} c_{1f} + a_2 c_{2f} c_{1r})] - c_{1f} c_{1r} l^2 k\};\\
&D_n = c_{1f} c_{1r} l^2 k \delta v_x;
\end{aligned} \tag{3.19}$$

Third-degree polynomial root location

A generic form valid for both Eq. (3.16) and Eq. (3.18) is expressed in Eq. (3.20), where r is the independent variable and $A \neq 0$.

$$Ar^3 + Br^2 + Cr + D = 0, \qquad \text{with} \qquad A, B, C, D \in \mathbb{R} \tag{3.20}$$

According to the celebrated Abel-Ruffini theorem ([52]), a third-degree polynomial may be reduced to a second-order one by finding one of the solutions to the latter, and can consequently be written as the product in Eq. (3.21).

$$Ar^3 + Br^2 + Cr + D = (r - r_1)(Ar^2 + Er + F), \qquad \text{with} \qquad E, F \in \mathbb{R} \tag{3.21}$$

A number of purely algebraic manipulations are then performed:

- Recalling that $A \neq 0$, Eq. (3.20) is multiplied by $27A^2$, yielding the expression in Eq. (3.22), whose first two terms evoke the binomial expansion of $(3Ar+B)^3$.

$$27A^3r^3 + 27A^2Br^2 + 27A^2Cr + 27A^2D = 0; \tag{3.22}$$

- A change of variable is performed and rearranged to isolate r:

$$t = 3Ar + B \implies r = \frac{t-B}{3A} \tag{3.23}$$

- The variable change for r in Eq. (3.23) is then substituted in Eq. (3.22), to then yield Eq. (3.24):

$$\begin{aligned} &27A^3r^3 + 27A^2Br^2 + 27A^2Cr + 27A^2D = \\ &\qquad = 27A^3\left(\frac{t-B}{3A}\right)^3 + 27A^2B\left(\frac{t-B}{3A}\right)^2 + 27A^2C\frac{t-B}{3A} + 27A^2D = \\ &\qquad = t^3 + (9AC - 3B^2)t + 27A^2D - 9ABC + 2B^3 \end{aligned} \tag{3.24}$$

Which in turn can be written as:

$$t^3 + pt + q = 0, \qquad \text{where} \qquad \begin{aligned} p &= 9AC - 3B^2; \\ q &= 27A^2D - 9ABC + 2B^3; \end{aligned} \tag{3.25}$$

- Excluding the trivial solution in the hypothesis of $p, q = 0$, the form of the solution t is:

$$t = u + v \tag{3.26}$$

Consequently, Eq. (3.25) undergoes a further transformation:

$$\begin{aligned} 0 = (u+v)^3 + p(u+v) + q &= u^3 + v^3 + 3u^2v + 3uv^2 + p(u+v) + q \\ &= u^3 + v^3 + 3uv(u+v) + p(u+v) + q \\ &= u^3 + v^3 + (3uv + p)(u+v) + q \end{aligned} \tag{3.27}$$

With the aim of further simplifying the equation and building a second equation to uniquely identify variables u and v, $3uv+p$ was set to zero, which transforms

Eq. (3.27) into Eq. (3.28).

$$u^3 + v^3 = -q \tag{3.28}$$

Two the unique solutions are found by solving the system in Eq. (3.29):

$$\begin{cases} u + v = t & \text{(3.29a)} \\ uv = -\dfrac{p}{3} & \text{(3.29b)} \end{cases}$$

- Coupling Eq. (3.28) with Eq. (3.29) yields the following:

$$\begin{cases} u^3 + v^3 = -q \\ u + v = t \\ uv = -\dfrac{p}{3} \end{cases} \Rightarrow \begin{cases} u^3 + v^3 = -q \\ u^3 v^3 = -\dfrac{p^3}{27} \end{cases} \tag{3.30}$$

- According to Vieta's formulas, the sum of the roots of a polynomial is equal to minus the coefficient to the first-order term, and their product is equal to the constant term of the equation: judging from Eq. (3.30), u^3 and v^3 are then roots of the second-degree polynomial of the form:

$$s^2 + qs - \frac{p^3}{27} \tag{3.31}$$

The roots in question are then computed:

$$\begin{aligned} s_1 = u^3 &= -\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}} \\ s_2 = v^3 &= -\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}} \end{aligned} \tag{3.32}$$

where the square root argument can be re-labelled as Δ:

$$\Delta = \frac{q^2}{4} + \frac{p^3}{27} \tag{3.33}$$

Eq. (3.32) marks the final result of a series of algebraic manipulations, operated to obtain the solutions of the described third-degree polynomial. The next step is starting from these results to revert back to the initial algebraic frame and obtain the original variable of interest: the yaw rate r. For that to be possible, the sign of Δ needs to be investigated to check the domain of the solutions and perform correctly the variable substitution.

Being $A, B, C, D \in \mathbb{R}$ implies that $p, q \in \mathbb{R}$: however, this does not prevent the possibility of having $\Delta < 0$, which would entail complex solutions $s_{1/2}$ and in turn, a complex u. Nevertheless, a complex u does not necessarily lead to a complex yaw rate r. Eq. (3.29a) states that t (hence, in turn, r) is the sum of u and v: if the latter two are complex conjugates, their complex parts cancel out when performing their sum, hence yielding a real number. Consequently, a real solution can still be found for t although the complex domain may need to be explored by u and v for that to happen.

A convenient way of framing the problem can be introduced to include the possibility of crossing the complex domain. The roots of polynomial in Eq. (3.31) are reported below.

$$\begin{cases} u^3 = -\dfrac{q}{2} - \sqrt{\dfrac{q^2}{4} + \dfrac{p^3}{27}} & \text{(3.34a)} \\ v^3 = -\dfrac{q}{2} + \sqrt{\dfrac{q^2}{4} + \dfrac{p^3}{27}} & \text{(3.34b)} \end{cases}$$

Since they both are algebraic sum of terms $-q/2$ and $\pm\Delta$, they recall the binomial theorem in the form:

$$z^n = s \tag{3.35}$$

where $n = 3$ and for the purposes of this investigation, z and s are generalised placeholders respectively for the variables u, v and s_1, s_2. To account for potential complexity of parties involved, both z and s are defined as complex numbers and in polar coordinates, with respective magnitudes $|z|$ and $|s|$ and orientation on the Argand plane ζ and ξ. The resulting binomial equation follows, where De Moivre's theorem is used to compute the n-th power of z:

$$z^n = s \quad \Longleftrightarrow \quad |z|^n(\cos n\zeta + i \sin n\zeta) = |s|(\cos \xi + i \sin \xi) \tag{3.36}$$

Considering the orientation of s to vary as $\xi + k_\xi 2\pi$, where $k_\xi \in \mathbb{N}$, the modulus and orientation of z may be inferred:

$$\begin{aligned} &|z| = \sqrt[n]{|s|} \\ &n\zeta = \xi, \xi + 2\pi, \xi + 4\pi \cdots \Rightarrow \zeta = \frac{\xi}{n} + k_\xi \frac{2\pi}{n}, \quad k_\xi = 0, \dots, n-1 \end{aligned} \tag{3.37}$$

Substituting the modulus and orientation of z from Eq. (3.37) into Eq. (3.36) yields a final solution form:

$$z = \sqrt[n]{|s|} \left(\cos\left(\frac{\xi}{n} + k_\xi \frac{2\pi}{n} \right) + i \sin\left(\frac{\xi}{n} + k_\xi \frac{2\pi}{n} \right) \right), \quad k_\xi = 0, \dots, n-1 \tag{3.38}$$

According to the fundamental theorem of algebra, a polynomial of degree n has n roots, thus Eq. (3.25), has either three real solutions or a real one along with a pair of complex conjugate solutions. Recalling that the nature of the solutions s_1, s_2 depends on that of Δ, the problem can be split in two case studies:

1. $\Delta \geq 0$.

   Eq. (3.32) yields two real solutions s_1, s_2: for s to be real, its complex part in Eq. (3.36) has to be null, hence $\xi = 0$. Consequently, Eq. (3.38) becomes:

   $$z = \sqrt[n]{|s|} \left(\cos\left(k_\xi \frac{2\pi}{n} \right) + i \sin\left(k_\xi \frac{2\pi}{n} \right) \right), \quad k_\xi = 0, \dots, n-1 \tag{3.39}$$

   Therefore the three solutions are:

$$\begin{cases} z_1 = \sqrt[3]{s} \\ z_2 = \sqrt[3]{s}\left(\cos\left(\frac{2\pi}{3}\right) + i\sin\left(\frac{2\pi}{3}\right)\right) \\ z_3 = \sqrt[3]{s}\left(\cos\left(\frac{4\pi}{3}\right) + i\sin\left(\frac{4\pi}{3}\right)\right) \end{cases} \Leftrightarrow \begin{cases} z_1 = \sqrt[3]{s} \\ z_2 = \sqrt[3]{s}\left(-\frac{1}{2} + i\frac{\sqrt{3}}{2}\right) \\ z_3 = \sqrt[3]{s}\left(-\frac{1}{2} - i\frac{\sqrt{3}}{2}\right) \end{cases} \tag{3.40}$$

A further simplification may be operated considering that

$$\left(-\frac{1}{2} - i\frac{\sqrt{3}}{2}\right) = \left(-\frac{1}{2} + i\frac{\sqrt{3}}{2}\right)^2 \tag{3.41}$$

Finally yielding the following solutions set:

$$\begin{cases} \alpha = \left(-\frac{1}{2} + i\frac{\sqrt{3}}{2}\right) \\ \alpha^2 = \left(-\frac{1}{2} - i\frac{\sqrt{3}}{2}\right) \end{cases} \Leftrightarrow \begin{cases} z_1 = \sqrt[3]{s} \\ z_2 = \alpha\sqrt[3]{s} \\ z_3 = \alpha^2\sqrt[3]{s} \end{cases} \tag{3.42}$$

Reverting back to Eq. (3.26) and considering both s_1 and s_2, in turn related to u and v, the three solutions in Eq. (3.42) become:

$$\begin{cases} u_1 = \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \\ u_2 = \alpha\sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \\ u_3 = \alpha^2\sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \end{cases} \qquad \begin{cases} v_1 = \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \\ v_2 = \alpha\sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \\ v_3 = \alpha^2\sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \end{cases} \tag{3.43}$$

Combined sums of u and v would give a total of nine solutions, while only three are expected: this is because not all of the available solutions satisfy the constraints in Eq. (3.29b). Considering that $\alpha^3 = 1$, the three compliant solutions are:

$$\begin{cases} t_1 = u_1 + v_1 \\ t_2 = \alpha u_1 + \alpha^2 v_1 \\ t_3 = \alpha^2 u_1 + \alpha v_1 \end{cases} \Leftrightarrow \begin{cases} t_1 = \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} + \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \\ t_2 = \alpha\sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} + \alpha^2\sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \\ t_3 = \alpha^2\sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} + \alpha\sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \end{cases} \tag{3.44}$$

2. $\Delta < 0$.

For Eq. (3.32) to be solved, s_1 and s_2 have to be written as:

$$s_1 = u^3 = -\frac{q}{2} - i\sqrt{-\Delta} \tag{3.45}$$

$$s_2 = v^3 = -\frac{q}{2} + i\sqrt{-\Delta} \tag{3.46}$$

It is straightforward to notice that s_1 and s_2 are complex conjugates:

$$s_{1/2} = -\frac{q}{2} \pm i\sqrt{-\Delta} = x \pm iy \tag{3.47}$$

and the magnitude and orientation of their polar coordinates are computed:

$$|s_1| = |s_2| = \sqrt{x^2 + y^2}, \qquad \cos\xi = \frac{x}{|s|} \tag{3.48}$$

Combined with this knowledge, Eq. (3.48) becomes:

$$
\begin{aligned}
|u| = |v| &= \sqrt[3]{\sqrt{\left(-\frac{q}{2}\right)^2 + \left(\sqrt{-\frac{q^2}{4} - \frac{p^3}{27}}\right)^2}} = \sqrt{-\frac{p}{3}} \\
\cos\xi &= \frac{-\dfrac{q}{2}}{\sqrt{-\dfrac{p^3}{27}}}
\end{aligned}
\tag{3.49}
$$

which coupled with Eq. (3.38) yields:

$$
\begin{cases}
z_1 = \sqrt{-\frac{p}{3}}\left(\cos\left(\frac{\xi}{3}\right) + i\sin\left(\frac{\xi}{3}\right)\right) \\
z_2 = \sqrt{-\frac{p}{3}}\left(\cos\left(\frac{\xi+2\pi}{3}\right) + i\sin\left(\frac{\xi+2\pi}{3}\right)\right) \\
z_3 = \sqrt{-\frac{p}{3}}\left(\cos\left(\frac{\xi+4\pi}{3}\right) + i\sin\left(\frac{\xi+4\pi}{3}\right)\right)
\end{cases}
\tag{3.50}
$$

Reverting back to u and v and recalling they are complex conjugates:

$$
\begin{cases}
u_1 = \sqrt{-\frac{p}{3}}\left(\cos\left(\frac{\xi}{3}\right) + i\sin\left(\frac{\xi}{3}\right)\right) \\
u_2 = \sqrt{-\frac{p}{3}}\left(\cos\left(\frac{\xi+2\pi}{3}\right) + i\sin\left(\frac{\xi+2\pi}{3}\right)\right) \\
u_3 = \sqrt{-\frac{p}{3}}\left(\cos\left(\frac{\xi+4\pi}{3}\right) + i\sin\left(\frac{\xi+4\pi}{3}\right)\right)
\end{cases}
\tag{3.51}
$$

$$
\begin{cases}
v_1 = \sqrt{-\frac{p}{3}}\left(\cos\left(\frac{\xi}{3}\right) - i\sin\left(\frac{\xi}{3}\right)\right) \\
v_2 = \sqrt{-\frac{p}{3}}\left(\cos\left(\frac{\xi+2\pi}{3}\right) - i\sin\left(\frac{\xi+2\pi}{3}\right)\right) \\
v_3 = \sqrt{-\frac{p}{3}}\left(\cos\left(\frac{\xi+4\pi}{3}\right) - i\sin\left(\frac{\xi+4\pi}{3}\right)\right)
\end{cases}
\tag{3.52}
$$

The final expressions are obtained from reverting to Eq. (3.26):

$$\begin{cases} t_1 = u_1 + v_1 \\ \\ t_2 = u_2 + v_2 \\ \\ t_3 = u_3 + v_3 \end{cases} \Leftrightarrow \begin{cases} t_1 = 2\sqrt{-\frac{p}{3}}\cos\frac{\xi}{3} \\ t_2 = 2\sqrt{-\frac{p}{3}}\cos\frac{\xi + 2\pi}{3} \\ t_3 = 2\sqrt{-\frac{p}{3}}\cos\frac{\xi + 4\pi}{3} \end{cases} \tag{3.53}$$

In this case, for the condition in Eq. (3.29b) to be fulfilled the complex parts of the solutions have to cancel out with one another. The three obtained solutions are therefore real, although the complex domain had to be navigated in order to reach them.

Final expressions

A compact overview of the final expressions follows:

- $\Delta = 0$

$$t_1 = 2\sqrt[3]{-\frac{q}{2}} \tag{3.54}$$

$$t_2 = -\sqrt[3]{-\frac{q}{2}} \tag{3.55}$$

$$t_3 = t_2 \tag{3.56}$$

- $\Delta > 0$

$$t_1 = \sqrt[3]{-\frac{q}{2} - \sqrt{\Delta}} + \sqrt[3]{-\frac{q}{2} + \sqrt{\Delta}} \tag{3.57}$$

$$t_2 = \alpha \cdot \sqrt[3]{-\frac{q}{2} - \sqrt{\Delta}} + \alpha^2 \cdot \sqrt[3]{-\frac{q}{2} + \sqrt{\Delta}} \tag{3.58}$$

$$t_3 = \alpha^2 \cdot \sqrt[3]{-\frac{q}{2} - \sqrt{\Delta}} + \alpha \cdot \sqrt[3]{-\frac{q}{2} + \sqrt{\Delta}} \tag{3.59}$$

- $\Delta < 0$

$$t_1 = 2\sqrt{-\frac{p}{3}}\cos\frac{\xi}{3} \tag{3.60}$$

$$t_2 = 2\sqrt{-\frac{p}{3}}\cos\frac{\xi + 2\pi}{3} \tag{3.61}$$

$$t_3 = 2\sqrt{-\frac{p}{3}}\cos\frac{\xi + 4\pi}{3} \tag{3.62}$$

Ultimately, the inverse variable change in Eq. (3.23) is applied to each root to obtain the corresponding yaw rate values $r_{1...3}$. The tyre slip angles can easily be computed backtracking the initial steps, through Eq. (3.11) and Eq. (3.12): finally, the constitutive relation Eq. (2.4) can be used to obtain the sideslip angle β.

However, not all the computed yaw rate values are viable equilibria coordinates. Three aspects are investigated on solutions:

- **Domain**. Ensures that the solutions are real.
- **Consistency**. Checks that the front and rear tyre slip angles corresponding to the yaw rate solutions have the same sign.
- **Magnitude**. Prevents accepting solutions having unrealistically big magnitude by comparing them to the expected boundaries for both the yaw rate and the sideslip angle.

All three aspects are imposed as constraints on the solutions and will ultimately determine viable equilibrium points. To summarise, Section 3.4.1 proposed an analytic method to locate phase-plane equilibria for stability assessment purposes. To overcome non-invertibility problems affecting commonly available tyre models, a root-rational tyre model was designed to portray the relationship existing between the lateral force F_y and the sideslip angle β. The equilibria location problem can then easily be reduced to solving a third-order polynomial, whose solutions are finally checked for sign and domain consistency, and their magnitude is compared against realistically achievable limits of the examined states.

Results are fully discussed in [19] (under review), where the analytical procedure is compared against numerical solutions, both in terms of accuracy and required computational time. The observed tradeoff between the two is impressively advantageous: analytical solutions only differ from numerical solutions of a magnitude order up to 10^{-2} but are computed over 10^6 times faster than their numerical counterpart.

3.4.2 Desired reference design

The steady-state stability assessment conducted in Section 3.4.1 allows judging whether the vehicle operates safely or an action of the controller is needed to prevent imminent stability loss. For this to happen, the location of equilibria needs to be compared to that of the current vehicle working point. Unstable equilibria act as delimiters of the stability region (Section 2.4.2) for the present operating condition of the vehicle. If working point coordinates are too close to the unstable threshold coordinates, timely controller intervention is required by setting a so-called stability yaw rate reference (r_s). Otherwise, a more dynamic driving experience is allowed to the driver, where performance can be safely prioritised by setting a handling-enhancing yaw rate reference (r_h).

To guarantee a smooth transition between the two, a unique merged reference is defined by means of a weight factor ε that prioritises one or the other reference according to the vehicle operating condition. A more detailed description of the handling reference, stability reference and the merged reference is provided in the following subsections.

Handling reference definition

A yaw rate reference that successfully accentuates the vehicle handling capabilities requires knowledge of the vehicle cornering response. As for any other dynamical system, these evaluations are usually initiated by introducing the assumption of steady state [21], which translates to the vehicle longitudinal speed being considered constant ($\dot{v}_x = 0$). The following remarks all refer to a single-track model (Section

2.2) exhibiting understeering behaviour, since that is intrinsically more stable than an oversteering one.

The handling diagram is a well-known visual tool to study the vehicle cornering behaviour, as it displays changes in the vehicle steady-state lateral acceleration a_y prompted by variations of the steering wheel angle δ (Section 2.4.1): the observed relationship between the two is recalled to be linear at first, meaning that a change in δ leads to a proportionally equivalent change of a_y, then becomes non-linear when the vehicle comes closer to its friction limits and when the latter are reached, asymptotic.

Because of its straightforward nature, for a long time the handling diagram has been regarded as a strong stability and performance indicator [31, 53, 54]: many different reference strategies have been proposed based on this concept, namely a linear formulation for the yaw rate depending on the vehicle steering angle as in [55], and a more representative non-linear piecewise-defined law as in [32].

However intuitive, the handling diagram merely gives partial insight into the vehicle cornering behaviour, as a more thorough stability investigation requires the study of a larger amount of vehicle states. A compelling alternative was found in MAPs [21] (Section 2.4) , allowing vehicle handling behaviour assessment at a glance, by displaying on the two axes the mutually-induced changes of any two vehicle states of interest. Consider a generic $\rho - \delta$ MAP (Fig. 3.9), where the curvature ρ is defined in Eq. (3.63) as the ratio between the yaw rate r and the (constant) longitudinal velocity v_x: variations in the curvature ρ are illustrated against changes in the steering angle δ, for five different constant speeds ranging within $10 - 30$ m/s: the operative region of the vehicle is visibly encased within boundaries, constituted by the maximum steering wheel angle, the maximum lateral acceleration, the maximum and the minimum longitudinal speed.

$$\rho = \frac{r}{v_x} \tag{3.63}$$

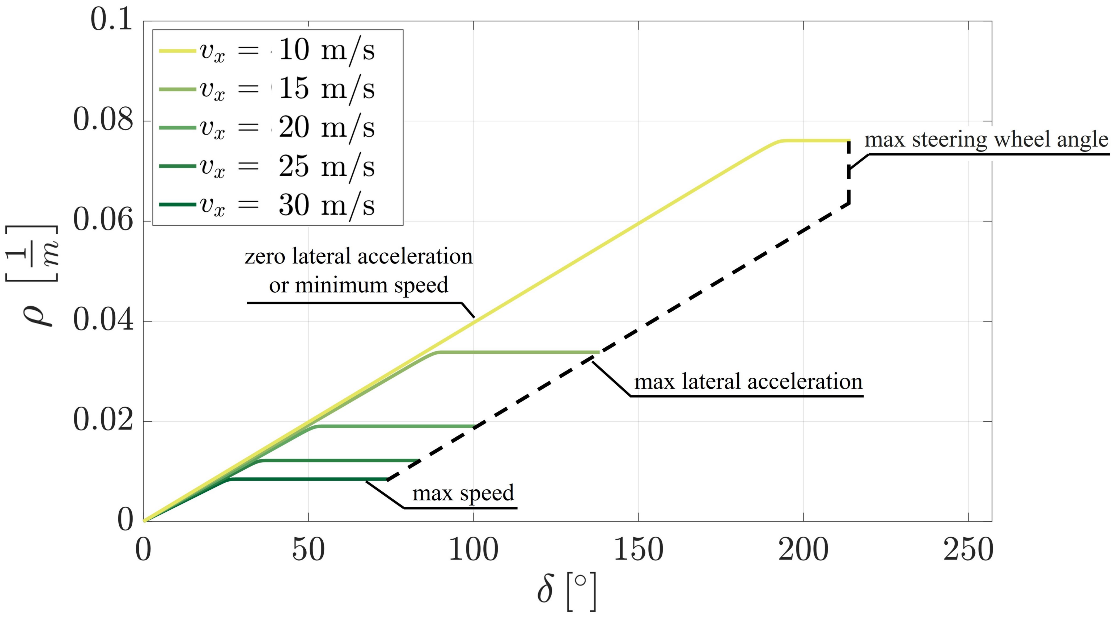


Figure 3.9: Example of cornering response of a passenger car represented through a $\rho - \delta$ MAP.

All the listed boundary quantities for MAPs are design parameters, which means that complete design freedom is guaranteed on relevant features of the MAP, and the dimension and shape of the plane region where it is defined can be customised at will. On that matter, some general design criteria have to be addressed, e.g. tackling the slopes of the constant speed lines: if they are too flat, too much steering effort is required by the driver to produce a change in curvature, but if they are too steep, even the slightest variation in the steering wheel angle would produce a huge change in the curvature.

The design steps are laid out hereafter:

1. Definition of the maximum achievable lateral acceleration $a_{y,max}$. Experimental results presented in [32] suggest that highly dynamic lateral manoeuvres for a full-scale vehicle lead to achieve a lateral acceleration of approximately 10 m/s^2. Consequently, considering the smaller size of the vehicle in use, this parameter is set to a slightly lower value of 9.7 m/s^2.

2. Definition of the desired maximum dynamic steering wheel angle for each vehicle speed, $\delta_{\text{dyn,max}}(v_x)$. The steering angle δ may be viewed as the sum of a

kinematic component δ_{kin}, that is predominant in low-speed manoeuvres, and a dynamic component δ_{dyn}, arising as a result of slip ([21]), as reported in Eq. (3.64).

$$\delta = \delta_{kin} + \delta_{dyn} \tag{3.64}$$

Consequently, δ_{kin} may be viewed as a lower boundary for steering itself: Eq. (3.65) holds, where l is the vehicle wheelbase. Furthermore, the small angle approximation (Section 2.2) allows to determine the steering angle neglecting the tangent.

$$\tan(\delta) = \frac{l}{R} = l\rho \longrightarrow \delta = l\rho \tag{3.65}$$

Because δ_{dyn} is bounded between being null and a generic value depending on the longitudinal speed $\delta_{\text{dyn,max}}(v_x)$, the steering angle δ in Eq. (3.65) is bounded between δ_{kin} and $\delta_{kin} + \delta_{\text{dyn,max}}(v_x)$. This is then reflected in curvature bounds: Eq. (3.66) conveys that the relationship between the steering angle and the curvature is proportional through the inverse of the wheelbase l, hence the $\rho - \delta$ MAP will be bounded within a stripe of plane having slope $1/l$ and width depending on the chosen maximum velocity-dependent dynamic steering angle $\delta_{\text{dyn,max}}(v_x)$.

$$\frac{\delta_{kin}}{l} \leq \rho \leq \frac{\delta_{kin} + \delta_{\text{dyn,max}}(v_x)}{l} \tag{3.66}$$

For the case at hand, $\delta_{\text{dyn,max}} = 50°$, independently on longitudinal velocity.

3. Compute the $(\rho_{max}, \delta_{max})$ coordinates at each constant speed. The maximum steering angle is simply given as a sum of its kinematic component and the chosen maximum value of its dynamic component, while the maximum curvature comes from combining the steady-state formulation for the lateral acceleration in Eq. (3.67) with Eq. (3.63), and substituting a_y with the maximum value $a_{y,max}$ set in step 1.

$$a_y = v_x r \tag{3.67}$$

The maximum steering angle and curvature are expressed respectively in Eq. (3.68) and Eq. (3.69):

$$\delta_{max} = \delta_{kin} + \delta_{\text{dyn,max}} \tag{3.68}$$

$$\rho_{max} = \frac{a_{y,max}}{v_x^2} \tag{3.69}$$

It is worth remarking that the value of the maximum curvature depends on the longitudinal velocity, hence five constant speeds equally-spaced within the range $10 - 30$ m/s were chosen.

4. Define a function appropriately representing the constant speed lines. While a considerable number of candidate functions would suit the purpose, there are a few constraints the elected function should comply by:

   - It goes through the origin and the endpoint computed in step 3, Eq. (3.70).

   $$f(0) = 0, \qquad f(\rho_{max}) = \delta_{max} \tag{3.70}$$

   - Its first derivative, hence its slope, at the origin, cannot exceed $1/l$ (assuming a target understeering behaviour) because of the reasons listed in step 2, Eq. (3.71).

   $$f'(0) \leq \frac{1}{l} \tag{3.71}$$

   - Its second derivative has to ensure a convex behaviour:

   $$f''(\rho) \leq 0 \tag{3.72}$$

   The rational function in Eq. (3.73) was designated for the purpose, featuring the design parameters p_1 and q_1 whose values are retrieved by using the constraint equations, for each of the chosen velocities.

$$\rho = \frac{p_1 \delta}{\delta + q_1} \tag{3.73}$$

5. Create a yaw rate handling reference r_h. This is done by taking the product between the newly-defined curvature (depending on the speed and the steering angle) $\rho(v_x, \delta)$ and the speed itself, as detailed in Eq. (3.74). Given a range of values for both the steering angle δ and the speed v_x, to enforce a quicker reference retrieval r_h is simply stored in a two-dimensional lookup table.

$$r_h = \rho(v_x, \delta) v_x \tag{3.74}$$

Stability reference definition

Further to the above-stated controller aims, when a potential stability loss is detected, the stability reference will be selected to help the driver smoothly regain stable control of the vehicle. Consequently, a threshold stability condition must be determined, so that a scaled-down version of that condition will represent a safe reference to be followed by the vehicle. It is straightforward that the magnitude of the scaling factor in question has to be lower than 1.

For simplicity of implementation, the steady-state approximation for the lateral acceleration in Eq. (3.67) was used as starting point to establish the threshold condition: this equation can be easily inverted to retrieve the yaw rate r. To add an extra degree of safety, the resulting yaw rate was scaled down by a constant factor k_s. The value $k_s = 0.7$ was empirically chosen, as it yielded a conservatively safe yaw rate value in simulations. The complete safety reference is expressed in Eq. (3.75).

$$r_s = k_s \frac{a_y}{v_x} \tag{3.75}$$

Adaptive merged reference definition

Two reference yaw rate formulations have been proposed thus far: the handling reference r_h is selected during stable operating conditions to leave more handling freedom to the driver, otherwise a safety reference r_s dominates the controller, so that stability can be regained. Nonetheless, it should be remarked that the controller requires a single reference rather than two: this calls for the definition of a unique, merged formulation that includes both references and is able to adaptively prioritise one or the other according to the driving scenario at hand.

A weight-based solution was then introduced, where the reference selection is regulated by means of a weight factor ε, given as the value at each time step of a smooth varying function, as proposed in [56]. To introduce ε, the stability region boundaries have to be identified, hence the location of unstable equilibria on the phase plane is established using the procedure in Section 3.4.1. The current operating point of the vehicle on the phase plane is also known: the yaw rate working point r_{wp} is available via virtual sensors initially and sourced from the IMU on the full-scale vehicle later on, while the sideslip angle working point β_{wp} is estimated because of the expensive and difficult-to-retrieve nature of such a measurement. At each time instant, the working point coordinates (β_{wp}, r_{wp}) will be compared against the stability region boundaries and their closeness to the latter will be evaluated by means of two indexes I_β and I_r, defined respectively in Eq. (3.76) and Eq. (3.77).

$$I_\beta = 1 - \operatorname{sign}((\beta_{max} - \beta_{wp})(\beta_{wp} - \beta_{min})) \times \frac{min(|\beta_{max} - \beta_{wp}|, |\beta_{wp} - \beta_{min}|)}{\frac{\beta_{max} - \beta_{min}}{2}} \tag{3.76}$$

$$I_r = 1 - \operatorname{sign}((r_{max} - r_{wp})(r_{wp} - r_{min})) \times \frac{\min(|r_{max} - r_{wp}|, |r_{wp} - r_{min}|)}{\frac{r_{max} - r_{min}}{2}} \tag{3.77}$$

The highest between I_β and I_r indicates which of the two states is closest to reaching critical conditions: consequently, the maximum between the two is compared to some threshold value I_t, and the weight factor ε is finally computed by means of Eq. (3.79). As in [56], to be more conservative on the stability threshold, the threshold value for the index is set to be $I_t = 0.7$.

$$I_{\max} = \max(I_{\beta}, I_{\mathrm{r}}) \tag{3.78}$$

$$\varepsilon(I_{\max}) = \begin{cases} 0, & \text{if}\, 0 \leq I_{\max} < I_t \\ \frac{1}{2}(1 - \cos\left(\pi \frac{I_{\max} - I_t}{1 - I_t}\right)), & \text{if}\, I_t < I_{\max} \leq 1 \end{cases} \tag{3.79}$$

Chapter 4

Sideslip angle estimation

The presented reference generation strategy entirely relies on knowledge of the stability condition of the vehicle at its current operational condition, which specifically translates to observing so-called stability indicators. As mentioned in Chapter 2, the most insightful quantities in that regard are the vehicle sideslip angle β and the yaw rate r. While the latter is measured with standard hardware, retrieving a reliable signal for the sideslip angle is a much more challenging task, due to the inherent difficulty and related cost. A plethora of alternative methods to measuring have been conjured within the literature and are currently used to make the sideslip angle available for applications requiring that information.

A renowned and documented technique to cater to this need is using estimation-based techniques, e.g. Kalman filters [37]. However easy to implement, filtering-based techniques suffer from an important downside: they rely on the linearisation of the system about an estimate of the current mean and covariance. Driving conditions that challenge the vehicle lateral dynamics, such as those the ESC acts on to ensure vehicle stability, are highly demanding for tyres which will be subject to non-linear phenomena. Entering the non-linear regime has extensive negative impact on model and parameter accuracy, affecting in turn the quality of the estimate.

With the growing interest in recent years for machine learning, these techniques naturally progressed and adapted to vehicle dynamics as well, especially neural networks. The most valuable quality that neural networks possess is being model-independent: indeed, they aim to provide a correlation between some input measurements and a target output, by manipulating such measurements to reach the closest possible approximation of the target output. This property is extremely useful for those vehicle quantities that are hard to measure or whose estimation requires the use of computationally complex non-linear filters, like the vehicle sideslip angle β. A neural network-based estimator is described in Section 4.1.

On the other hand, to overcome the major drawback of neural networks, which is needing a large amount of training data, a parametric solution is also proposed, where the sideslip angle is retrieved as the combination of a kinematic, determinsitic component and a dynamic one, coming from a simple parametric interpolation involving the steering angle δ and the vehicle lateral acceleration a_y. The interpolation-based estimator is deepened in Section 4.2.

4.1 Neural network-based estimator

Multiple examples of neural network-based sideslip angle estimation algorithms are present in the literature. In [57], Wei et al. present a sideslip angle estimation algorithm based on a General Regression Neural Network (GRNN), where the signal of interest is retrieved by correlation in time of yaw rate and lateral acceleration. Bonfitto *et al.* [58] use a pattern recognition neural network to infer the operating friction condition of the vehicle and guide a second neural network in the estimation of the sideslip angle. Gräber et al. in [59] propose coupling a Recurrent Neural Network (RNN) with a kinematic vehicle model for estimation purposes.

Despite the clear impact of the literature on the matter, what seems to be lacking is a more in-depth explanation on neural network input selection: while it is common to choose readily-available signals retrieved by popular on-board sensors, there is a concrete risk for dynamic overfitting when the neural network is trained on an excessive amount of data, albeit dynamically rich. Choosing wisely the neural network inputs not only leads to a significant reduction in the data processing and synchronisation effort, but it also promotes generalisation, which is the ability to withhold the same estimation performance in a different scenario than the training one [60].

Including the descriptive physics of the phenomenon of interest in the training process greatly improves the estimate quality. This principle was exemplified in [61], where Raissi *et al.* introduced physics-informed neural networks (PINNs): PINNs seamlessly integrate deep learning with fundamental physics principles, ensuring that the learned solutions not only fit the provided data but also comply with the equations governing the underlying physical phenomena.

A feedforward neural network is hereby proposed for sideslip angle estimation, whose inputs are the outcome of a thorough sensitivity analysis on multiple combinations of commonly-employed signals. The network estimation target is the dynamic component of the sideslip angle β_{dyn}, which in the training phase is extrapolated from the overall sideslip angle measurement β through pre-compensation, i.e. by singling out the deterministic, easy-to-retrieve kinematic component β_{kin} as in Eq. (4.1). Further insight on the kinematic and dynamic components of the sideslip angle is provided in [62]: the vehicle response corresponding to driving scenarios with little lateral excitation is captured by the kinematic sideslip angle, defined in Eq. (4.2), where one may recall a_1 and a_2 being the distances of the vehicle CoG respectively from the front and rear axle and δ being the steering angle.

$$\beta_{dyn} = \beta - \beta_{kin} \tag{4.1}$$

$$\beta_{kin} = \delta \frac{a_2}{a_1 + a_2} \tag{4.2}$$

In validation phase, β_{dyn} estimated by the neural network is summed to β_{kin} computed using the steering angle input at each time instant to reconstruct the complete sideslip angle signal β.

Filtering out the deterministic component of the sideslip angle exposes the nonlinearities reflecting the tyre slips: this means having a direct window into the remaining dynamic sideslip angle, which will then be the specific target of the neural network. Input selection targeting the dynamic component of the sideslip angle helps drawing conclusions on which signals provide the best insight into the dynamics of interest.

Infusing the neural network output with physics knowledge reintroduces the deterministic component of the sideslip angle through a simple approximation of the latter. It should be noted that the proposed application differs from the work presented in [61] because instead of informing the network by introducing the underlying physics as Partial Differential Equations in the training loss function, a simple approximation of one of the involved quantities is injected within the framework. However, this is done with the same intent, which is essentially aiding the estimation algorithm through the knowledge of available physics describing the phenomenon at hand. To avoid any confusion, from this point forward the hereby discussed approach will be referred to as “physics-infused” rather than physics-informed.

The set of analysed inputs features signals obtained through standard on-board sensors, i.e. steering angle δ, longitudinal velocity v_x, yaw rate r, longitudinal acceleration a_x, and lateral acceleration a_y. With the aim of assessing the validity of the physics-infused strategy, a benchmark approach was defined, with the same hidden architecture but targeting the whole sideslip angle and using all available inputs indiscriminately.

Comparison between the physics-infused and benchmark approach is operated by computing the root-mean-square error (RMSE) on the full sideslip angle β, both on the training and on the validation datasets, to be able to evaluate respectively the training performance and the robustness to different driving scenarios from the training ones.

A visual comparison between benchmark and proposed approach is provided by Fig. 4.1: despite the aforementioned differences, the two approaches have the same architecture, made of three layers with 20, 25 and 18 neurons each. Furthermore, networks from both methods are trained using the same BFGS-optimised backpropagation algorithm ([60]) on the same dataset, to set the ground for a fair comparison on methodology. The employed dataset is publicly available ([63, 64]), and contains test sessions performed on a 1965 Ferrari 250 LM Berlinetta GT. Data is organised in structures corresponding to their source, namely driver, sensors, suspensions, tyres. Each structure features sets of measurement structures: each measured quantity is provided as a vector, alongside the corresponding time vector, the measurement unit and the used acquisition frequency. Calibration offsets and vehicle parameters (e.g. static load distribution on the four wheels, location of the CoG) are also present in case the user needs such information for post-processing purposes. All numbers in the dataset are in double precision type. The total time length of the datasets amounts to over three hours, and features laps on the Palm Beach International Raceway in Jupiter, Florida.

To standardize data for suitable analysis and usage, a number of post-processing steps were followed for the primary variables, which are the handwheel steering angle δ, yaw rate r, longitudinal speed v_x, longitudinal acceleration a_x and lateral acceleration a_y. Firstly, to ensure comparability, normalisation to the lowest sample rate (10ms, corresponding to 100Hz) was carried out, as a difference was observed in the acquisition frequencies of primary variables. The pre-normalisation acquisition frequencies of each primary variable are reported in the last column of Table 4.1. Moreover, aiming to remove sensor noise while maintaining signal dynamics, filtering was performed using a third-order low-pass Butterworth filter with 5Hz cut-off frequency, following the indications in [65].

Table 4.1 summarises key statistics of primary variables, including minimum and maximum values, mean, median, standard deviation and sampling frequency. These statistics have been averaged over thirty laps, all performed by the same driver for increased consistency.

From Table 4.1, a few general observations arise:

- The minimum and maximum values highlight the operational range of the chosen signals in the context of lap data. The lateral acceleration a_y and the yaw rate r are particularly of interest, as their wide range indicates significant lateral dynamics. Data referring to longitudinal speeds lower than $1\,m/s$ was removed in order to avoid including vehicle standstill scenarios having little dynamic content, which explains the minimum value for the speed.

Table 4.1: Key statistics of primary variables, averaged over thirty laps. Measurement units are indicated in the first column along with the variable name, except for the frequency (expressed in Hz).

Primary variable	Min	Max	Mean	Median	σ_{data}	f_s (Hz)
$\delta\,(°)$	-103.1	98.22	-8.90	-2.44	37.02	1000
$r\,(°/s)$	-26.31	23.73	-0.98	-0.02	7.26	100
$v_x\,(m/s)$	1.02	47.71	7.91	7.28	3.11	100
$a_x\,(m/s^2)$	-5.20	4.83	0.15	0.42	3.01	100
$a_y\,(m/s^2)$	-7.34	8.09	-1.30	-0.30	5.36	100

- Mean and median values are close to zero for lateral dynamics-related quantities (as the vehicle turns right and left during laps), while they give deeper insight on the average speed per lap.
- The standard deviation σ_{data} for the steering angle δ is particularly high: this is not surprising, as it highlights the spread of wide-ranged data.

The sampling frequency column lists the different acquisition frequencies before a posteriori normalisation is performed. Measurement units are indicated in the first column along with the variable name, except for the frequency (expressed in Hz).

4.1.1 Assessment of insightful inputs

The first set goal is to determine which input combination leads to the best estimation performance. Fundamentals of combinatorics state that the total number $C_{n,k}$ of possible k-element combinations of n objects without repetition is the binomial

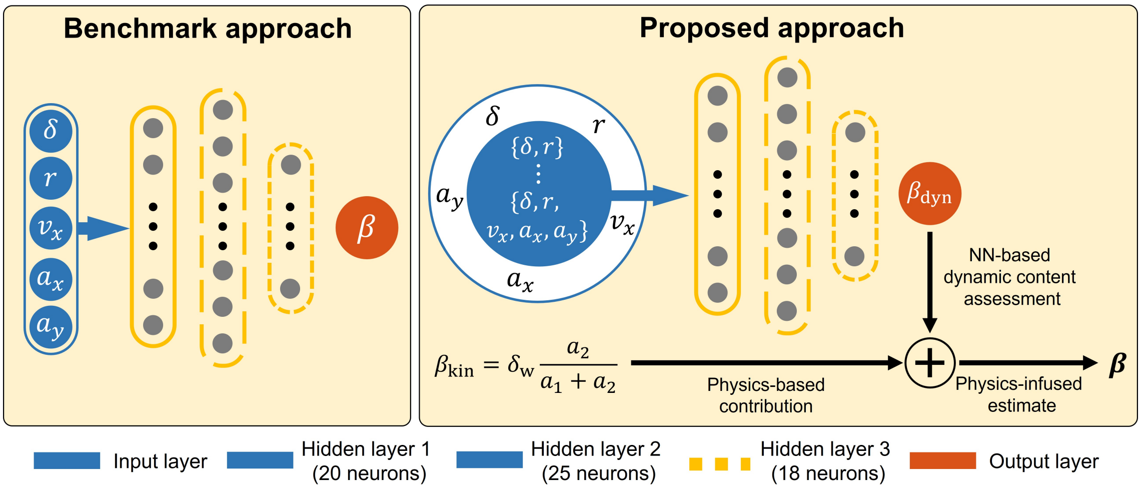

Benchmark approach
δ
r
v_x
a_x
a_y
β
Proposed approach
δ
r
a_y
v_x
a_x
$\{\delta, r\}$
⋮
$\{\delta, r, v_x, a_x, a_y\}$
β_{dyn}
NN-based dynamic content assessment
$\beta_{\text{kin}} = \delta_{\text{w}} \frac{a_2}{a_1 + a_2}$
Physics-based contribution
Physics-infused estimate
$\boldsymbol{\beta}$
Input layer
Hidden layer 1 (20 neurons)
Hidden layer 2 (25 neurons)
Hidden layer 3 (18 neurons)
Output layer

Figure 4.1: Schematic for benchmark and infused networks: the first features all five inputs and the sideslip angle as only output, while the latter explores a combination of input signals and targets the dynamic sideslip angle. They both feature the same architecture.

coefficients, which is explicited in compact form in Eq. (4.3).

$$\mathrm{C}_{n,k} = \binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{4.3}$$

In the case at hand, $n = 5$ while $k = \{1, \ldots, 5\}$: summing the binomial coefficient for each k-sized subset of inputs gives the total number of possible input combinations. Setting aside the single-input combinations (as one input is most likely not an insightful enough descriptor of the vehicle behaviour), a total of 26 combinations is obtained, as reported in Eq. (4.4).

$$\sum_{k=2}^{5} \mathrm{C}_{5,k} = 10 + 10 + 5 + 1 \tag{4.4}$$

Despite this being a manageable number of combinations, the individual value of some of the inputs at hand may be assessed in advance in order to perform a smarter search. Consequently, individual inputs are fed to the network and the obtained RMSE for each of them is computed: the least promising input is neglected in the following investigations. It is worth noting that reducing the number of viable inputs to 4 considerably reduces the possible input combinations to 11.

The resulting metrics for the single-input configurations are detailed in Table 4.2, featuring the training and validation RMSEs in degrees for all individual inputs as well as for the benchmark input set.

Table 4.2: Single input-induced training and validation RMSEs on benchmark and infused approach.

Approach	**Input set**	**Training RMSE** (°)	**Validation RMSE** (°)
Benchmark	$[\delta, v_x, r, a_y, a_x]$	0.3670	0.6515
Infused	δ	0.7878	1.0051
	v_x	2.3761	2.4099
	r	0.8135	0.7585
	a_y	0.6098	0.5786
	a_x	2.4735	2.5238

A couple of straightforward conclusions may be drawn:

- The best faring input is the lateral acceleration a_y, whose training RMSE is the lowest among the individual ones, closely followed by the steering angle δ. The former also achieves a lower validation RMSE than the benchmark approach.
- Both the longitudinal acceleration a_x and speed v_x score the worst training and validation performance. Moreover, being the longitudinal acceleration by definition the variation in time of the longitudinal velocity, it can be considered a redundant information: a_y is then excluded from following investigations.

Outcomes for two-input configurations are reported in Table 4.3.

Table 4.3: Training and validation RMSEs on benchmark and infused approach, with the latter featuring all possible two-element combinations of δ, v_x, a_y, r.

Approach	**Input set**	**Training RMSE** (°)	**Validation RMSE** (°)
Benchmark	$[\delta, v_x, r, a_y, a_x]$	0.3670	0.6515
Infused	$[\delta, v_x]$	0.7263	0.7409
	$[\delta, r]$	0.6703	0.6118
	$[\delta, a_y]$	0.4880	0.4964
	$[v_x, r]$	0.5586	0.5957
	$[v_x, a_y]$	0.5355	0.6177
	$[r, a_y]$	0.9433	0.9525

Judging from Table 4.3, some important remarks can be made:

- The best performance is obtained from the input $[\delta, a_y]$, providing a better validation RMSE than the one relative to the benchmark approach.
- Despite the weak performance of v_x as individual input, the configuration having as input $[v_x, a_y]$ proves to be the next best performer: this could be linked to the ability of the latter to portray both the longitudinal and lateral dynamics of the vehicle when evaluating the dynamics of the vehicle.

Three-input and four-input networks estimation abilities are displayed in Table 4.4, whose numerical results in Table 4.4 set the ground for a few deductions:

Table 4.4: Training and validation RMSEs on benchmark and infused approach, with the latter featuring all possible three-element combinations of δ, v_x, a_y, r, as well as the only four-input one.

Approach	**Input set**	**Training RMSE** (°)	**Validation RMSE** (°)
Benchmark	$[\delta, v_x, r, a_y, a_x]$	0.3670	0.6515
Infused	$[\delta, v_x, r]$	0.7938	0.8652
	$[\delta, r, a_y]$	0.6188	0.9805
	$[v_x, r, a_y]$	0.6386	0.6643
	$[\delta, v_x, a_y]$	0.4375	0.5585
	$[\delta, v_x, r, a_y]$	0.4608	0.4135

- The best-performing input triplet is $[\delta, v_x, a_y]$: this reflects the previous result in Table 4.3, where the most promising two-input networks feature δ, v_x, a_y.
- The validation RMSE for the former is bigger than it was for the couple $[\delta, a_y]$.

4.1.2 Physics-infused approach evaluation

Finally, the comparison between five-input physics infused configuration is performed against the benchmark approach, and results are reported in Table 4.5.

Since at this stage both configurations have the same number of inputs and architecture, the outcome will provide an impartial evaluation of the efficacy of the physics infusion: as evidenced in Fig. 4.1, the benchmark approach outputs the full sideslip angle β, while the infused approach targets the estimation of the dynamic component of the sideslip angle β_{dyn} to then enrich it with the knowledge of the kinematic sideslip angle β_{kin}, hence yielding the full sideslip angle β.

Table 4.5: Training and validation RMSEs for the full input set of benchmark and infused approaches.

Approach	**Input set**	**Training RMSE** (°)	**Validation RMSE** (°)
Benchmark	$[\delta, v_x, r, a_y, a_x]$	0.3670	0.6515
Infused	$[\delta, v_x, r, a_y, a_x]$	0.3939	0.5025

Table 4.5 indicates a slightly better training outcome for the benchmark than the one obtained for the infused approach, but a better validation RMSE for the latter. An even more remarkable result is that the validation RMSE of the five-input physics-infused network (RMSE = 0.5025°) is bigger than the one obtained for the four-input configuration, featuring $[\delta, v_x, r, a_y]$ (RMSE = 0.4135°), and the two-input configuration using $[\delta, a_y]$ (RMSE = 0.4964°). The importance of this result lies in two observations:

- The configuration making use of all inputs does not necessarily have the best generalisation ability, i.e. it does not prevent the network from overfitting to the training data.

- Among the four-input and two-input architectures outperforming the benchmark and five-input infused one, the validation RMSE of the two-input architecture is less than 0.08° higher than the one for the four-input one: this entails that a structure favouring a smaller signal processing and synchronisation effort, of half as many signals to be more precise, can be traded off for a modest drop in precision.

Ultimately, this proves that a qualitative rather than quantitative choice of inputs not only favours a better estimation performance, but it also retains the physical meaning of the estimate. The winning input alternative $[\delta, a_y]$ is finally selected.

As a concluding remark to this study, a comparison between the benchmark and selected two-input infused approach with the measured sideslip angle is displayed in

Fig. 4.2 (top). Furthermore, the difference between the measured sideslip angle and both the benchmark and proposed estimates of the latter are compared in Fig. 4.2 (bottom).

4.2 Interpolation-based estimator

A neural network-based approach to sideslip angle estimation was presented in Section 4.1. However innovative and advantageous, because of its independence from a vehicle model, machine learning-based estimation strategies suffer from a crucial drawback: the need for large datasets. A huge amount of data is required to perform training and validation in an extensive enough fashion that a high margin of confidence is associated to the estimate, in every possible driving scenario.

An important result evidenced in Section 4.1 is the indisputable importance of the lateral acceleration a_y to ensure a good estimate of the sideslip angle. According to the second principle of dynamics, summarised in Eq. (4.5), the lateral acceleration a_y is related through the mass m (assumed to be constant) to the resultant lateral force F_y.

$$\sum F_y = ma_y \tag{4.5}$$

For the sake of clarity, the single-track model is depicted in Fig. 4.3 to evidence the kinematic steering angle component, reported in Eq. (4.6a), and the kinematic sideslip angle component, written in Eq. (4.6b), in relation to basic vehicle geometry (distances of CoG from front and rear axle, respectively a_1 and a_2) and assuming the small angle approximation is true.

$$\tan(\delta_w) = \frac{a_1 + a_2}{R} \approx \delta_w \tag{4.6a}$$

$$\tan(\beta_{kin}) = \frac{a_2}{R} \approx \beta_{kin} \tag{4.6b}$$

These yield a final formulation for the kinematic sideslip angle, expressed in Eq. (4.7), where β_{kin} depends on the steering angle δ_w and the easy-to-retrieve parameters a_1 and a_2.

$$\beta_{kin} = \delta_w \frac{a_2}{a_1 + a_2} \tag{4.7}$$

The dynamic sideslip angle β_{dyn} can be demonstrated to depend on the rear tyre slip angle α_2. The latter is related to the front tyre slip angle α_1 through the dynamic component of the steering angle, δ_{dyn}, which in turn can be singled out by removing the kinematic steering angle component δ_{kin}, defined as the ratio between the wheelbase and the curvature radius, from the overall steering angle δ.

Consequently, if a relationship exists between the lateral force F_y and the lateral acceleration a_y as well as between the tyre slip angle α and the dynamic sideslip angle β_{dyn}, it is fair to expect that a similar trend may be observed between β_{dyn} and a_y as the one depicted in constitutive plots between F_y and α. In the same way the tyre model offers a close enough representation of the true tyre behaviour, a mathematical relationship defined between the dynamic sideslip angle and the lateral acceleration would constitute an invaluable approximation resource, which could potentially be scaled to match other vehicles or even other operating conditions for the same vehicle.

4.2.1 Algebraic approximation definition

In Section 3.4.1 a new formulation was proposed to offer an analytically invertible approximation of the Dugoff tyre model using a set of interpolation parameters. A similar iterative process is followed here.

The procedure is listed as follows:

1. The dynamic sideslip angle β_{dyn} is isolated from the overall sideslip angle measurement β, coming from datasets [63, 64], by subtracting to the latter, in the form of Eq. (4.1), the kinematic sideslip component as defined in Eq. (4.2).

2. Looking back at Eq. (3.6), the obtained β_{dyn} replaces α and the lateral acceleration measurement a_y replaces the lateral force F_y. The new root-rational approximation appears:

$$a_y = -\frac{c_1\beta_{dyn}}{c_2\sqrt{(\beta_{dyn})^2} + c_3} \tag{4.8}$$

   It should be noted that, as stated in Chapter 2 and in accordance to what is stated by Guiggiani in [21], the tyre slip angle and sideslip angle feature opposite signs. As such quantities reside on the x-axis of the constitutive plot and of the interpolation curve respectively, the latter will appear flipped along the y-axis with respect to the former. This can be immediately observed upon comparison between Fig. 4.4 and Fig. 3.8, and is reflected in the minus sign in Eq. (4.8).

3. Parameters c_1, c_2, c_3 are retrieved by solving a non-linear least squares problem using the Levenberg-Marquardt optimisation algorithm. Initial conditions to solve the optimisation problem are kept the same as for the root-rational tyre model, due to their order of magnitude being the only relevant information. The optimisation problem is subsequently solved using the MATLAB ® function `lsqcurvefit`.

4. The correctness of the parametric approximation is assessed by juxtaposing the obtained interpolated curve to the point cloud corresponding to the available measurements, as reported in Fig. 4.4. A more thorough examination of the interpolating function can be found in Section 6.3.

The set of initial conditions and the optimised parameters are reported in Table 4.6.

4.2.2 Overall sideslip angle estimation

At present state, Eq. (4.8) defines the lateral acceleration as a function of the dynamic sideslip angle. Bearing in mind that the final goal of this process is having

Table 4.6: Starting parameters to initiate optimal search (first row) against optimised parameters (second row).

Parameter set	c_1	c_2	c_3
c_0	7000	500	7500
c_{opt}	10542.45	560.72	2776.88

an approximation of the overall sideslip angle, Eq. (4.8) must be inverted: in such way, β can be reconstructed as sum of β_{kin}, expressed in Eq. (4.2), and β_{dyn}, fully-known as function of the lateral acceleration.

Due to the presence of a square root at the denominator, the inversion will result in a piecewise formulation of the sideslip angle, where different definitions are distinguished for positive or negative lateral accelerations. The inversion result is in Eq. (4.9).

$$\beta_{dyn} = \begin{cases} -\dfrac{c_3 a_y}{c_1 + c_2 a_y} & \text{if} \quad a_y < 0 \\ -\dfrac{c_3 a_y}{c_1 - c_2 a_y} & \text{if} \quad a_y \geq 0 \end{cases} \tag{4.9}$$

Once the dynamic sideslip angle β_{dyn} is known, the kinematic sideslip angle β_{kin} is computed using the steering angle knowledge, and the full sideslip angle β can be reconstructed.

4.2.3 Estimator validation

The validity of the estimation strategy is assessed by comparing its outcome against the measured signal: Fig. 4.5 displays the estimation outcome on a portion of dataset [63], chosen for its substantial dynamic content.

The numerical performance assessment is usually carried out with Root-mean-square error (RMSE), quantifying the deviation of the estimated signal from the measured one, averaged over the entire time duration of the instance. The RMSE formula for the instance at hand is reported in Eq. (4.10), and the resulting validation RMSEs on four different runs belonging to dataset [64] are listed in Table 4.7.

$$\text{RMSE} = \frac{\sqrt{\sum\limits_{\text{i}=1}^{N} ||\beta_{meas,i} - \beta_{est,i}||^2}}{\text{N}} \tag{4.10}$$

Table 4.7: Validation outcomes for four different datasets, featuring the obtained root-mean-square error (RMSE) as performance metric.

Validation dataset	**RMSE** (°)
`20140222_01_01_03_250lm`	0.539
`20140221_03_02_03_250lm`	0.375
`20140221_04_01_03_250lm`	0.541
`20140221_03_03_03_250lm`	0.502

Albeit straightforward, RMSE lacks judgement when it comes to the size of the error scaled to the overall magnitude of the signal: in other words, a discrepancy of 0.5° on an estimate of a signal of 10° should weigh less than the same discrepancy referring to a signal of magnitude 1°.

For this reason, a probability distribution function of the sideslip estimation error e_β, as defined in Eq. (4.11), is portrayed in Fig. (4.6) for the four datasets listed in Table 4.7. Furthermore, the expected value $E(e_\beta)$ and the variance σ^2 of the estimation error are presented in Table (4.8).

$$\text{e}_\beta = \beta_{meas} - \beta_{est} \tag{4.11}$$

Results in Table 4.8, lead to observe particularly low expected values for the sideslip angle error. Moreover, the variance values suggest little spread of the error

Table 4.8: Expected value $E(e_\beta)$ and variance σ^2 characterising the normalised probability function for the sideslip angle estimation error e_β.

Validation dataset	$\mathbf{E}(\mathbf{e}_\beta)$	$\boldsymbol{\sigma^2}$
`20140222_01_01_03_250lm`	-0.287	0.708
`20140221_03_02_03_250lm`	0.004	0.338
`20140221_04_01_03_250lm`	0.099	0.545
`20140221_03_03_03_250lm`	0.078	0.336

around its expected value, indicating in turn small error fluctuations throughout the entire validation datasets. Both these observations prove the obtained result is a good-quality estimate.

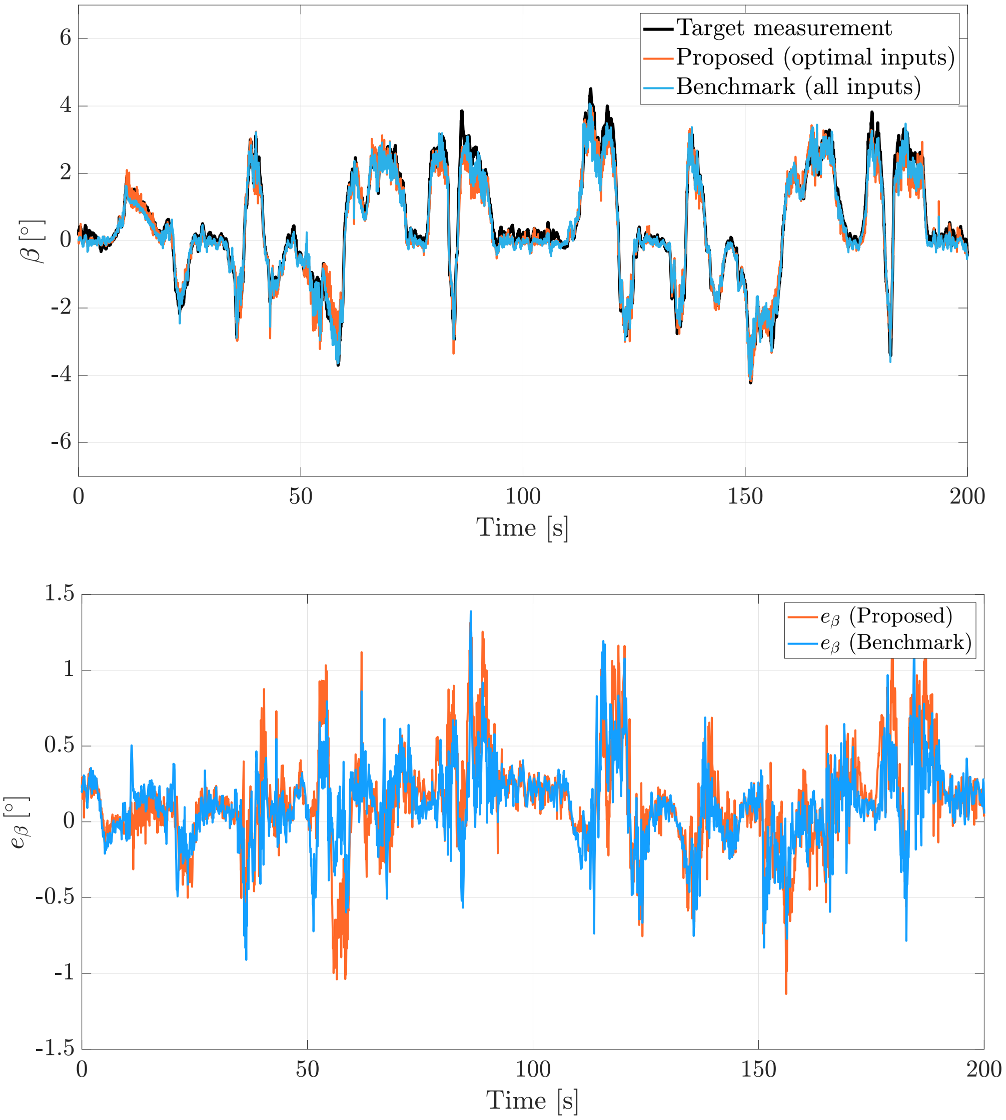


Figure 4.2: Compared view of measured sideslip angle with benchmark and chosen infused approach, with the latter using optimal inputs: (top) Full signal comparison; (bottom) Error between both estimates and measured signal.

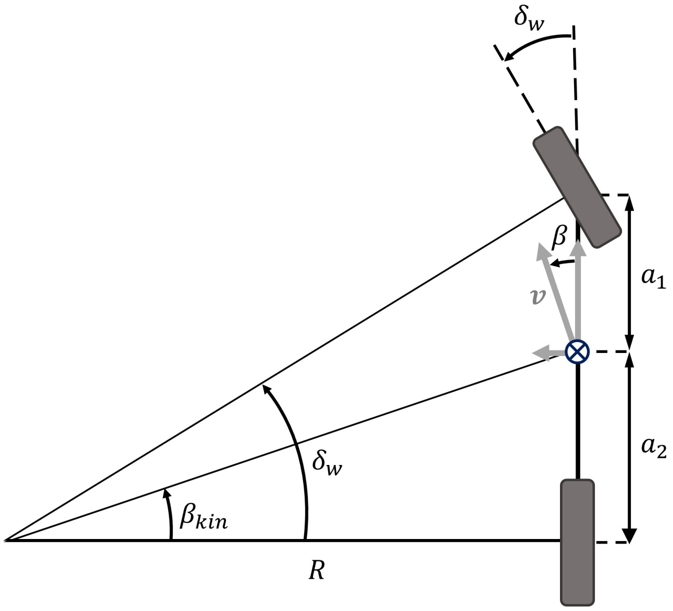


Figure 4.3: Single-track model portrayal of kinematic sideslip angle.

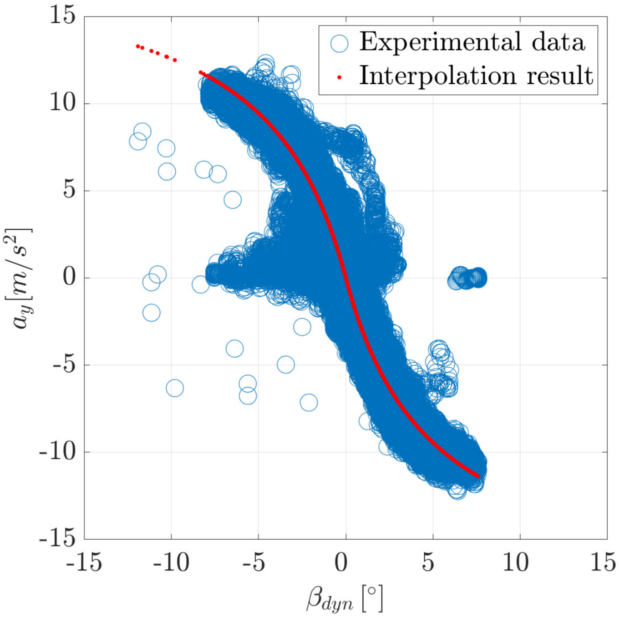


Figure 4.4: Interpolation outcome on available datasets compared to available experimental data.

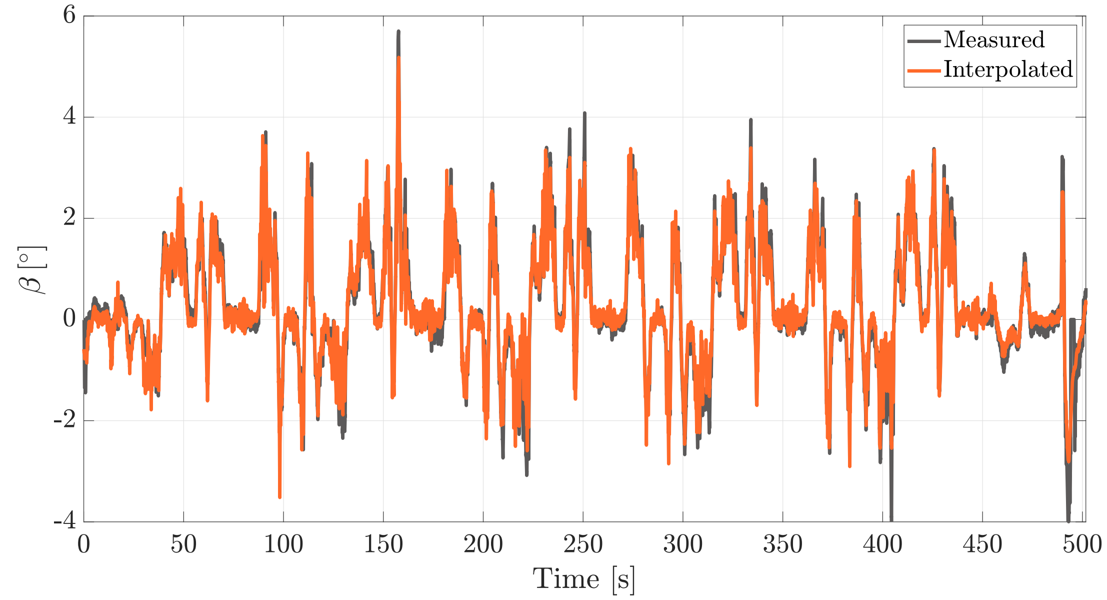


Figure 4.5: Comparison between measured and estimated sideslip angle, where the latter is evaluated through interpolation of dynamic signal and subsequent reconstruction of the complete signal.

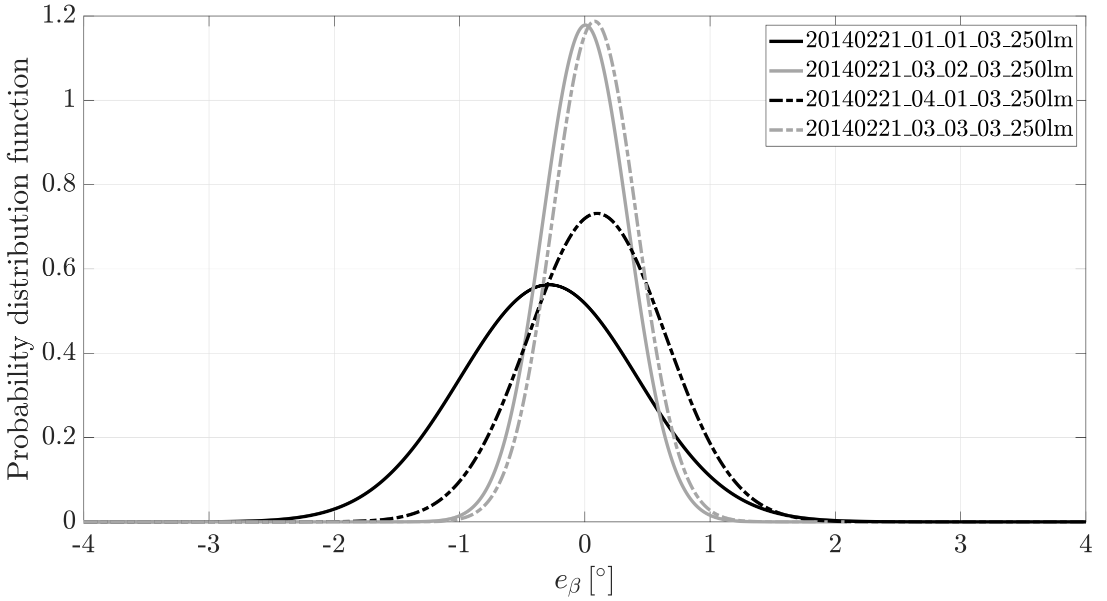


Figure 4.6: Probability distribution function of sideslip angle estimation error e_β for the datasets listed in Table 4.7.

Chapter 5

Controller validation

The framework described in Chapter 3 may essentially be categorised in two main elements: a vehicle stability controller, the validation of which is the purpose of this thesis, and an appropriate test bench for the latter, which mimics the behaviour of the full-scale vehicle closely enough to consider the examined driving scenarios to be truthful enough to the actual vehicle response. The controller must be developed alongside its test bench, for full testing support to be guaranteed to the controller.

This chapter describes every stage of the controller validation, beginning from preliminary performance assessment all the way to dedicated-platform testing. Details on both the controller and its validation test bench will be provided at each testing instance. The performed testing manoeuvre is a single lane-change, imparted on the vehicle by means of a driver model setting an input longitudinal velocity v_x and steering angle δ, as described in Section 3.1. Results on both a mild and a more challenging driving scenario are discussed, in order to assess the controller performance and ability to discern safe scenarios from unsafe ones, where instability threatens to endanger passenger safety.

5.1 Offline simulation

A sensible first step is to start by interfacing the controller, developed on Simulink ®, with a very simple single-track model approximation of the full-scale vehicle, defined on the same software: this is helpful to perform an entry-level compatibility check of the PI gains, as well as to ensure no unexpected controller responses occur in this simple scenario. The vehicle model and the controller are interfaced with Simulink ® connections, since they run within the same block scheme: more specifically, the controller-generated torques are virtually wired as inputs to the vehicle model and in turn, vehicle states serving as controller inputs are wired to the controller itself. A quick visual overview of this stage, labelled "offline simulation", is shown in Fig. 5.1.

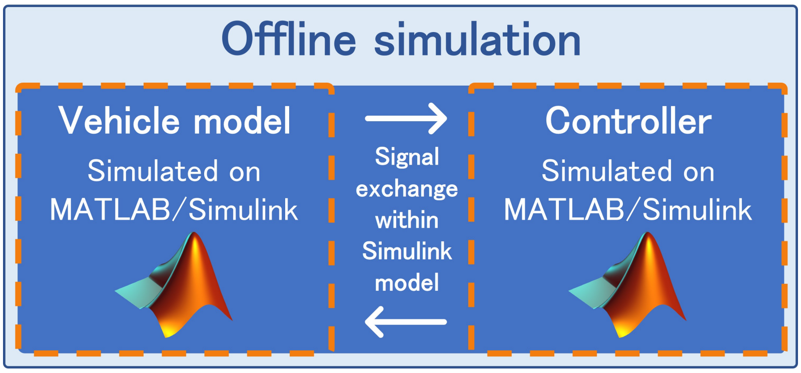


Figure 5.1: Offline simulation overview.

Results from this preliminary implementation encourage moving forward to add further complexity by performing offline co-simulations.

5.2 Offline co-simulation

Despite the single-track model being a useful tool to obtain a fairly faithful representation of the vehicle lateral behaviour, it fails to represent a considerable amount of phenomena that significantly affect the actual vehicle, as evidenced in [21]. Among these, perhaps the most relevant is the absence of lateral load transfer due to the wheels on the same axle being merged to obtain a single wheel per axle: this means neglecting rolling dynamics and in turn suspension dynamics and chassis compliance. When dealing with heavy cornering scenarios, as is the case while studying the stability of a vehicle, rolling dynamics will inevitably and consistently affect the dynamic vehicle behaviour, so they should not be neglected.

Aiming to achieve a more accurate portrayal of the test vehicle, a 15 DOF model is used, described in Section 3.2. The depiction of the vehicle as a system of equations coupling the chassis and the four wheels, considered individually as rigid bodies, constitutes a more detailed model, much better suited to the stated purposes. The 15 DOF model is defined in Simcenter Amesim ®, which is the new test bench software. A co-simulation instance is then called, enabling Simcenter Amesim ® and MATLAB/Simulink ® to communicate within the same time frame, despite running on different software. For this to happen, connections between the vehicle model and the controller have to be properly defined, meaning that the Simulink ® - defined output controller torques need to be virtually wired as Amesim ® inputs and the same needs to happen for the states. These are computed in Amesim ®, and transmitted to the Simulink ® - defined controller. Communication occurs though the presence of co-simulation interfaces in both software packages. Once the co-simulation instance is initiated on Amesim ®, the full co-simulation is called on Simulink ®. It is worth stating that co-simulation implies compatible run parameters: stemming from the linearisation analysis carried out in Section 3.2.2, a fixed-step second-order Runge-Kutta integration algorithm is chosen as vehicle model solver, where the integration step is 1 ms. The corresponding solver for the controller on Simulink ® is ode4 using the same integration time step.

A graphical overview of this stage, labelled “offline co-simulation”, is given in Fig. 5.2. It should be noted that the controller-generated torques are overlapped to the pre-existing ones within the vehicle model, which are in charge of initiating the motion of the vehicle.

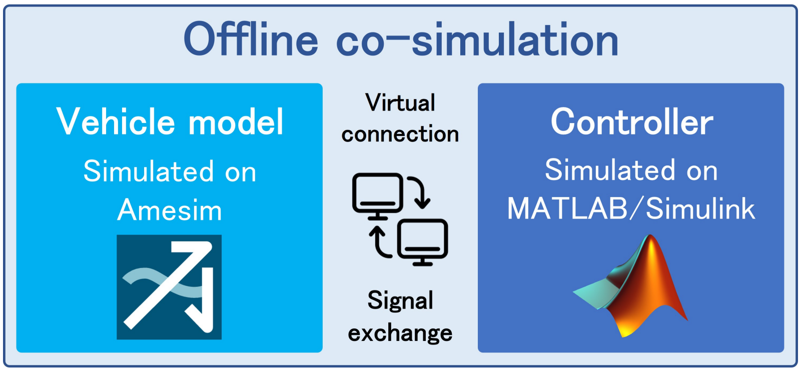


Figure 5.2: Offline co-simulation overview.

When framing the offline co-simulation instance, one can instantly notice that the run time of the simulation entirely relies upon the speed of the employed CPU, since that is the running platform for both software programs. The “offline” within the nomenclature of this stage is precisely due to this, as no indication is provided on the computational load related to the algorithm at this point of the validation journey.

5.2.1 Results

With the vehicle model being much more detailed than in the preliminary offline simulation stage, it is worth presenting some meaningful results relative to the offline co-simulation.

For the sake of comparability, testing will be performed using the same manoeuvre throughout consecutive validation steps. More specifically, as described in Section 3.1, a single lane-change manoeuvre is performed both with mild and challenging inputs so the validity of the control strategy can be assessed both in a safer operating condition and in a more unstable one. To recap from Section 3.1, the steering action is initiated after 10s and lasts for 2s, featuring an amplitude of 50° for both scenarios. The target speed is set to 25 m/s for the mild scenario, while in the challenging scenario the steering action commences as the vehicle is experiencing a 3 m/s^2 longitudinal acceleration while travelling at a longitudinal speed of approximately 33 m/s.

Results are presented in terms of the two most relevant stability indicators, i.e. the sideslip angle β and the yaw rate r respectively shown in Fig. 5.3 - 5.4 and Fig. 5.5 - 5.6.

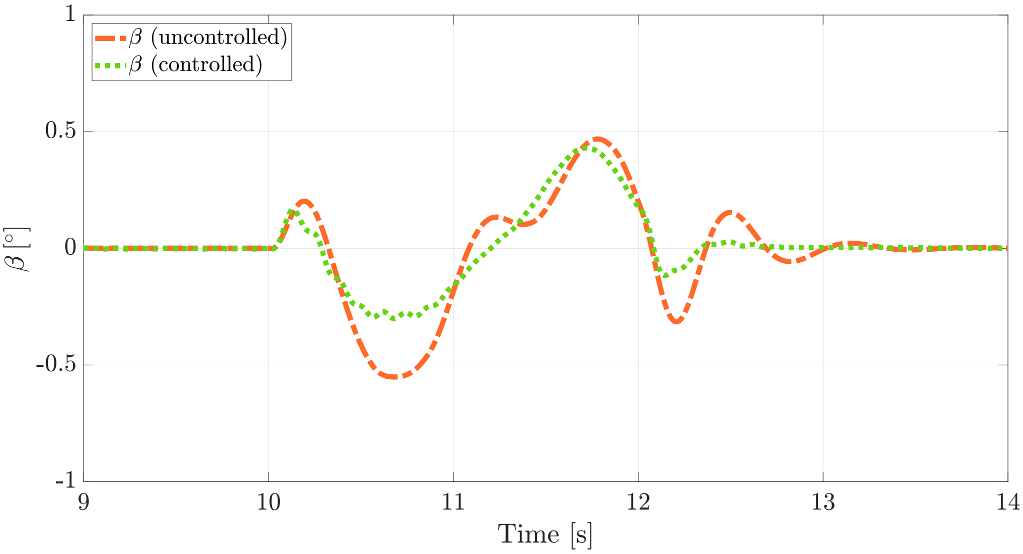


Figure 5.3: Sideslip angle response to the mild single lane-change, generated within offline co-simulation.

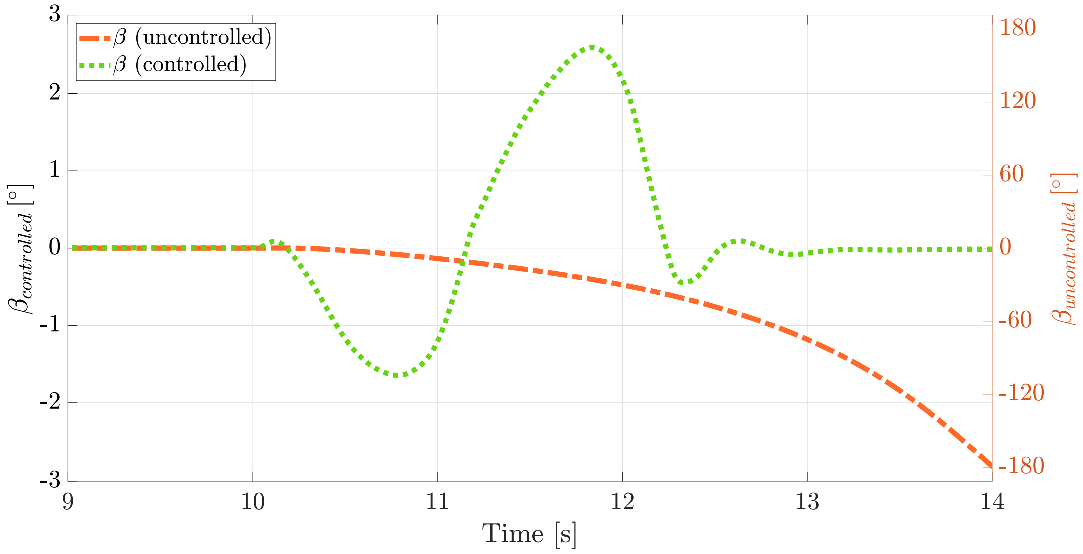


Figure 5.4: Sideslip angle response to the challenging single lane-change, generated within offline co-simulation.

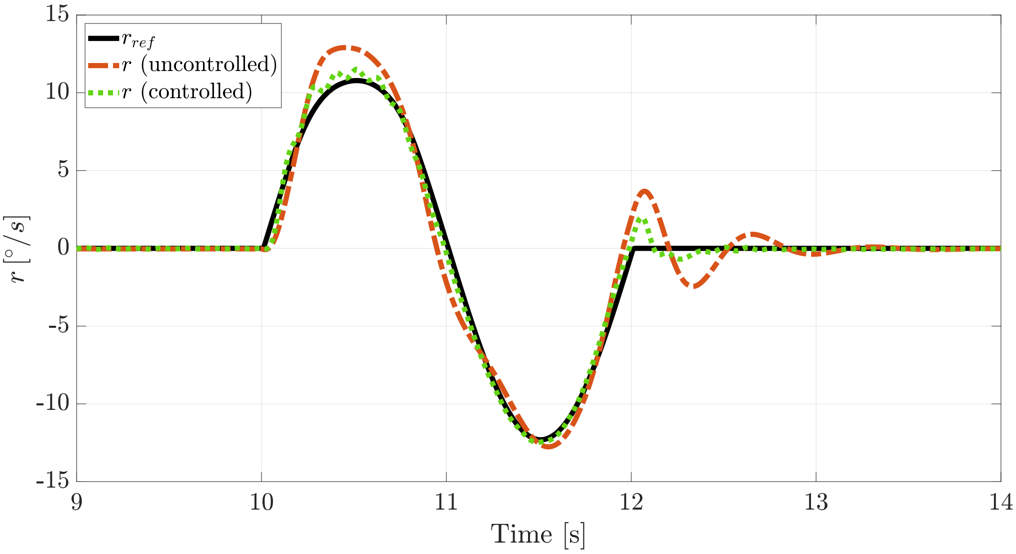


Figure 5.5: Yaw rate response to the mild single lane-change, generated within offline co-simulation.

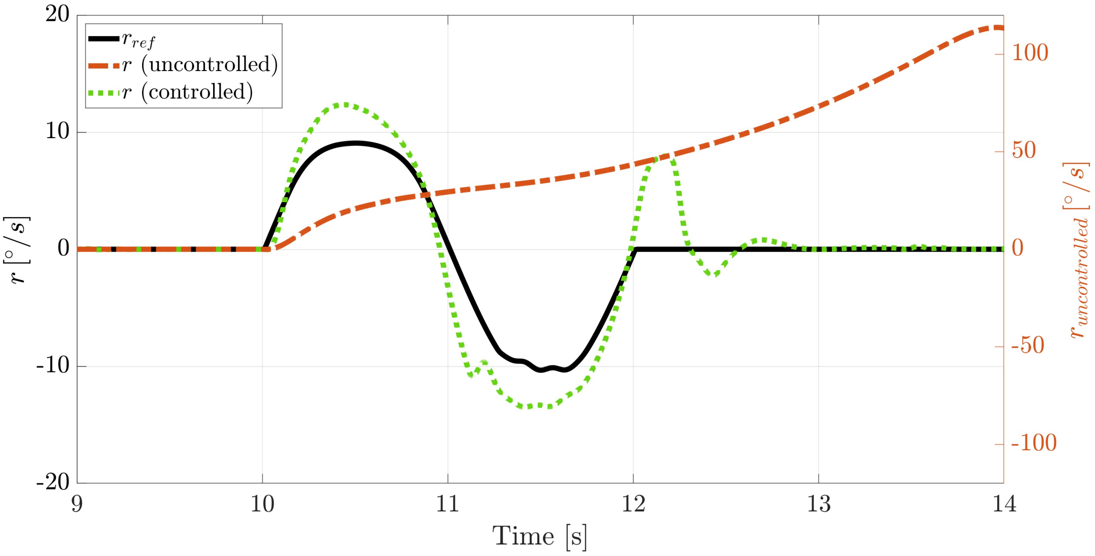


Figure 5.6: Yaw rate response to the challenging single lane-change, generated within offline co-simulation.

A couple of observations follow:

- In the mild scenario, the vehicle operates in a safe enough condition that the controller does not switch to the safe reference: the controlled signal is indeed very close to the uncontrolled one, for both the sideslip angle (Fig. 5.3) and the yaw rate (Fig. 5.5).

- The challenging driving conditions trigger huge instability in the uncontrolled vehicle, visible both in the sideslip angle (Fig. 5.4) and in the yaw rate (Fig. 5.6) where the difference between the controlled and uncontrolled response is highlighted by the presence of two different scales on the y-axis.

- The sideslip angle and the yaw rate in the challenging driving scenario reach values so extreme that they cease to make any physical sense, as an actual vehicle would never be able to reach such conditions. This phenomenon can be attributed to the lack of physically meaningful constraints in the simulated vehicle model. In contrast, the controlled response stays well within reasonable boundaries, indicating a correct triggering of the controller when it is needed.

A very insightful addition to the above results is given by looking at the stability indices I_β and I_r for both variables, as well as the weight factor ε, introduced in Section 3.4.2. It is straightforward to notice that while the indices I_β and I_r never exceed the set stability threshold within the mild scenario (Fig. 5.7), I_r does so in the challenging scenario, causing the weight factor ε to increase (Fig. 5.8).

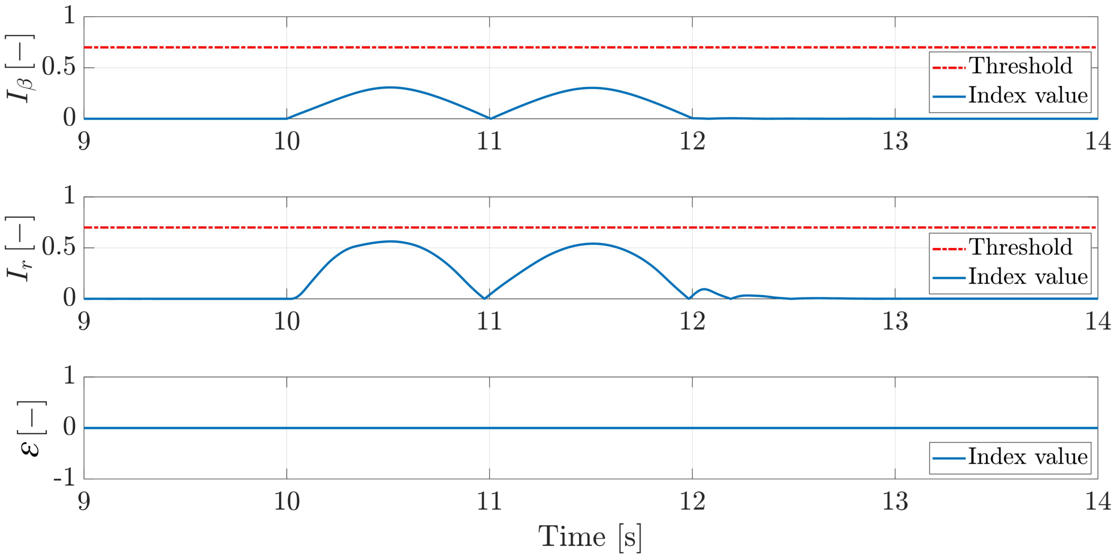


Figure 5.7: Yaw rate and sideslip stability indices and overall weight factor in response to the mild single lane-change, generated within offline co-simulation.

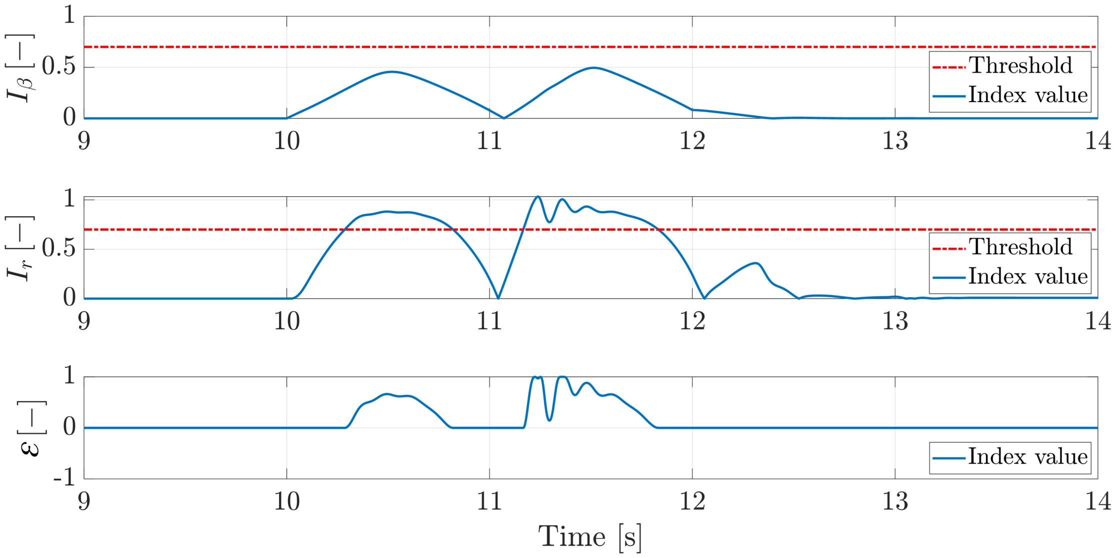


Figure 5.8: Yaw rate and sideslip stability indices and overall weight factor in response to the challenging single lane-change, generated within offline co-simulation.

Given that the set performance expectations were met, there is sufficient ground to move forward on the validation journey.

5.3 Real-time co-simulation

The offline co-simulation is a sensible first step on the validation journey, but it is far from being a truthful assessment of the controller algorithm. Indeed the full-scale vehicle will only unquestionably benefit from the controller if the latter acts immediately upon instability detection. Therefore, progression towards appropriate hardware is imperative. From here on out, the controller will be run on platforms providing progressively more substantial computational capabilities, all the way to being deployed on the real-time enabled Electronic Control Unit (ECU) that will eventually be placed in the vehicle for full-scale validation.

With that in mind, the first step is moving both the vehicle model and the controller to the powerful Siemens SIMATIC Rack IPC ® [66]. This is a real-time capable computational platform, which makes it a suitable candidate for future real-time testing. In view of eventually moving the controller to the actual ECU, besides dedicated power modules, a number of input/output (I/O) modules are wired to the

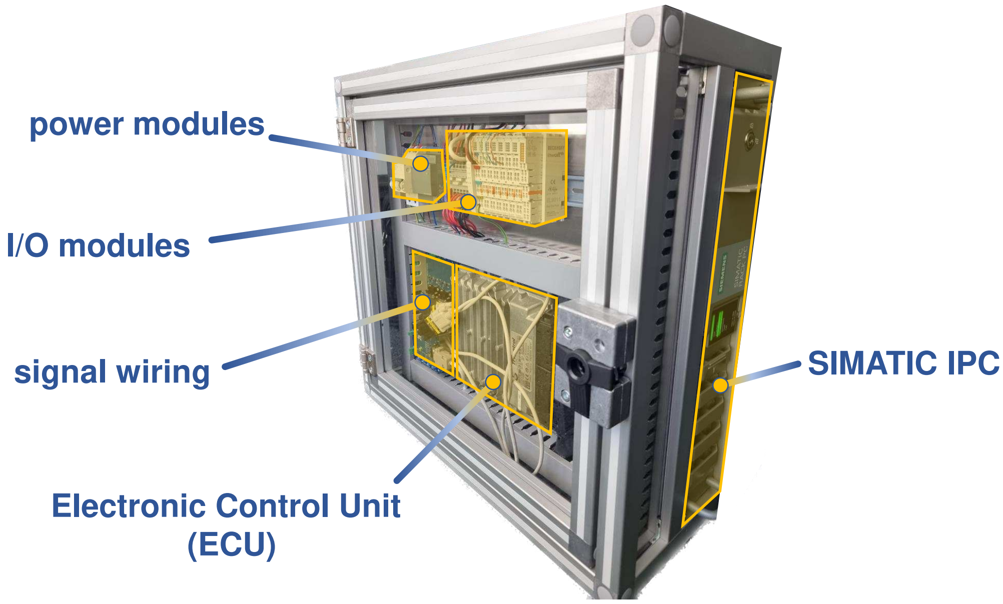


Figure 5.9: Real-time platform, featuring SIMATIC Rack IPC, ECU and associated interface components.

IPC, to enable both connection to the ECU as well as possibly sensors/actuators for fault testing. The overall ensemble of such component comes together in a unique component, referred to as real-time platform and shown in Fig. 5.9.

At the present stage, the ECU is not yet considered as it will be explored in further validation steps. The controller will instead be preliminarly deployed on the IPC to test vehicle/controller interfacing when moving away from original software. This testing stage is then labelled real-time co-simulation, as the completion time is now somewhat relevant due to the control scheme running on a real-time dedicated platform, albeit not the final one (the ECU) [67]. The real-time co-simulation is visually presented in Fig. 5.10.

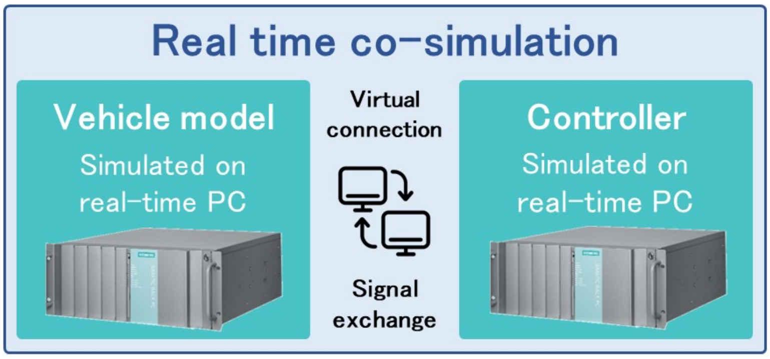


Figure 5.10: Real-time co-simulation overview.

For the vehicle model and controller to be loaded and running on the real-time platform, they have to be abstracted from their dependence on the original software. For this to happen, the vehicle model and the controller will become Functional Mock-up Units (FMUs), essentially black boxes whose interfaces are visible and properly wired to each other (e.g., output black-box controller torques will be connected as inputs to the black-box vehicle module). Complying to the Functional Mock-up Interface (FMI) standard means being able to run, on any real-time enabled hardware, dynamic models designed on any software. FMUs also feature exposed parameters, whose value may be changed without having to re-convert the whole model into a new FMU. One advantage of this feature is that it simplifies sensitivity analysis on any parameter of interest, provided that the designer has set it as exposed.

The real-time co-simulation instance is called from a Python script, which loads the FMUs as co-simulation slaves, meaning components receiving and transmitting data. The co-simulation slaves are all operating under the co-simulation master. This sets the time step for the simulation (1ms) and the simulation duration, as well as enabling simulation logging for easier debugging. The controller and test bench model building blocks are virtually wired by specifying where the input to any building block is sourced from, and where it is connected to. Finally, the co-simulation instance is called through code execution.

Simulation results are then collected in a single file, containing the input and output signals of both FMUs.

5.3.1 Results

Real-time co-simulation outcomes are reported in the same fashion as the offline co-simulation results. The sideslip angle β and the yaw rate r are shown in Fig. 5.11 - 5.12 (respectively in the mild and challenging scenario) and Fig. 5.13 - 5.14 (respectively in the mild and challenging scenario).

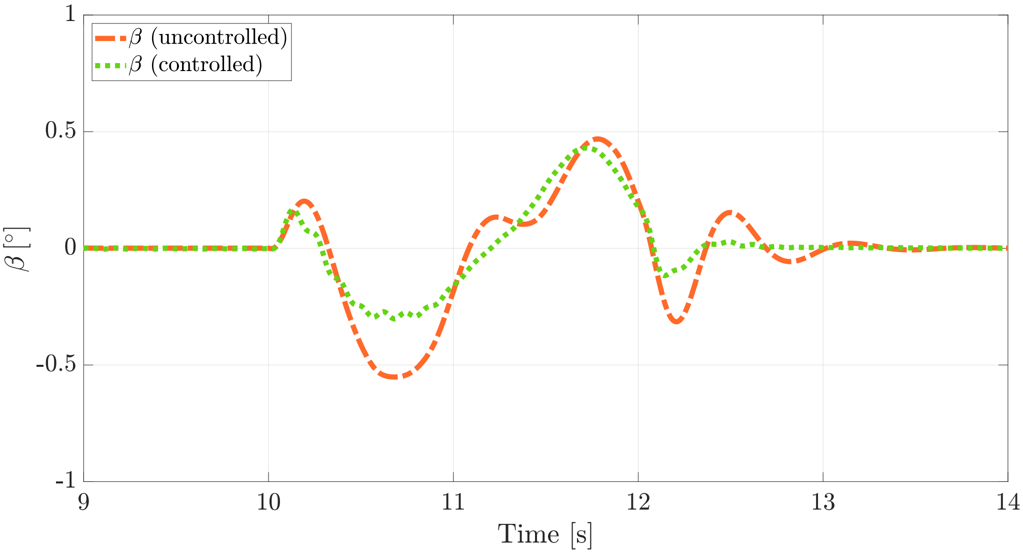


Figure 5.11: Sideslip angle response to the mild single lane-change, generated within real-time co-simulation.

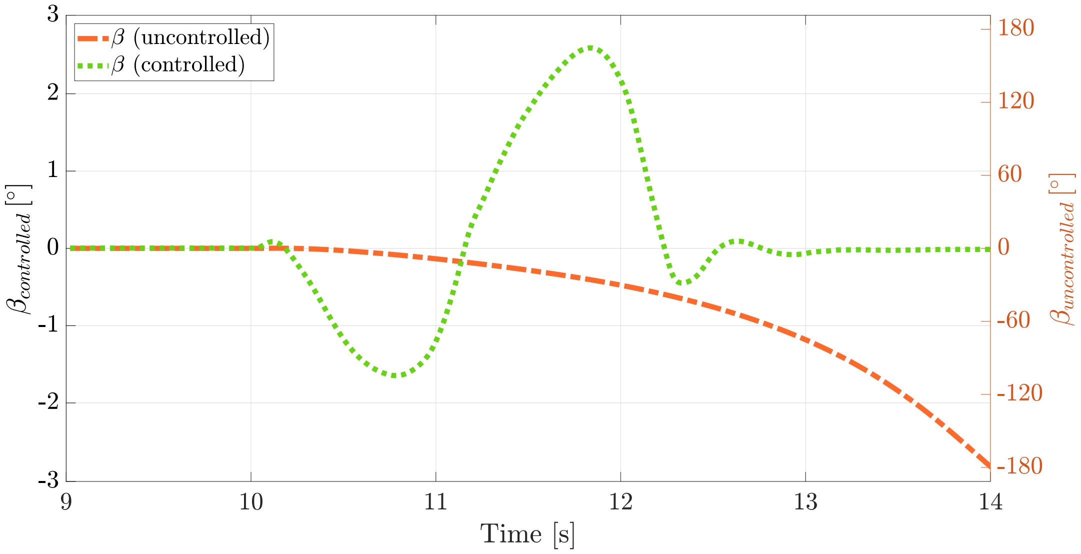


Figure 5.12: Sideslip angle response to the challenging single lane-change, generated within real-time co-simulation.

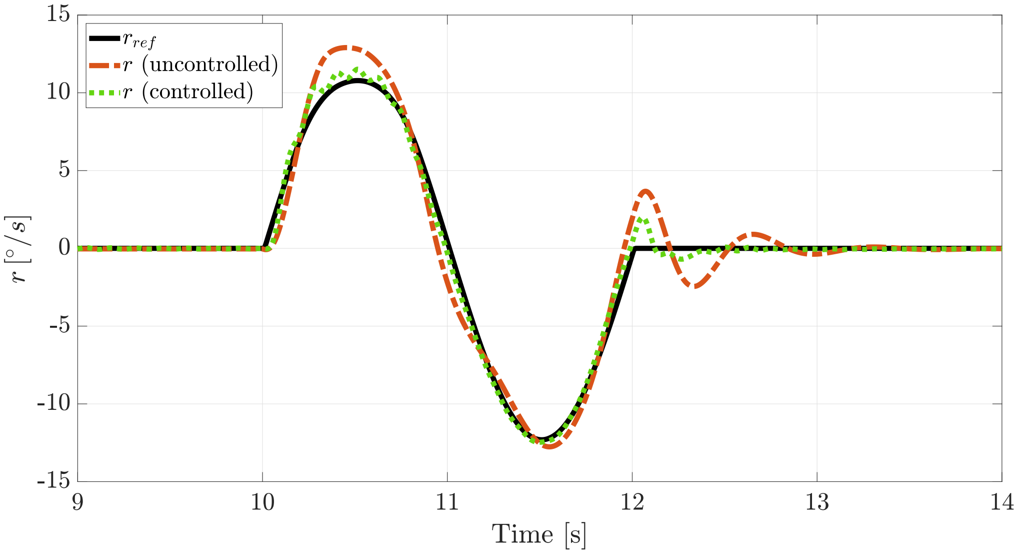


Figure 5.13: Yaw rate response to the mild single lane-change, generated within real-time co-simulation.

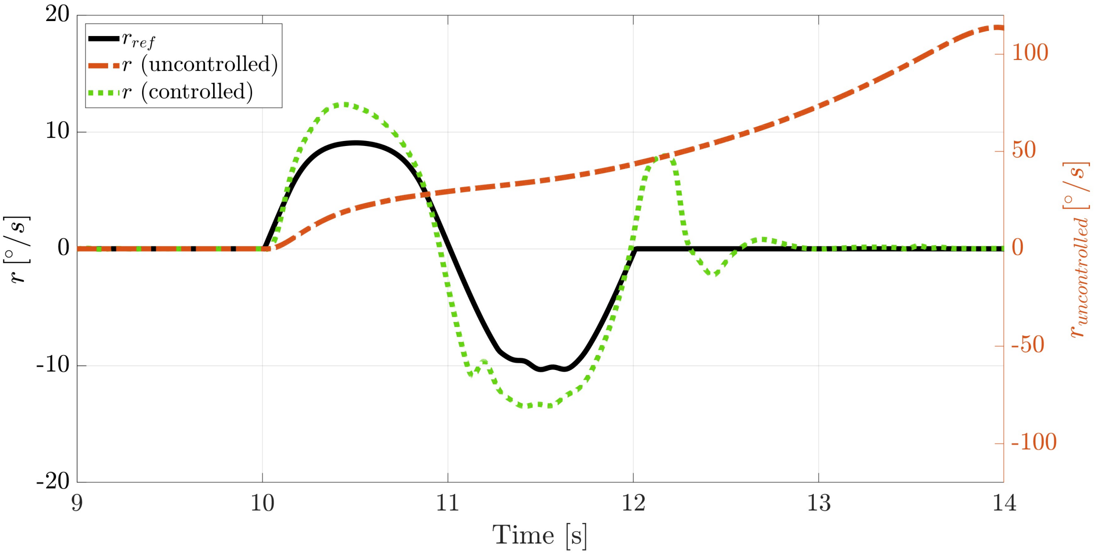


Figure 5.14: Yaw rate response to the challenging single lane-change, generated within offline co-simulation.

At a glance, these results are very similar to those presented for the offline co-simulation. To be more thorough in the performance assessment, a further comparison between results from the offline and real-time co-simulation is presented in Fig. 5.15 for the controlled sideslip angle in the challenging scenario. Two data tips are also featured, registered at the same time instant: the discrepancy between the two signals is of the order of 10^{-3}, and is most likely caused by the driving scenario and the vehicle model being defined within the same software in offline co-simulations, but as separate entities in real-time co-simulations.

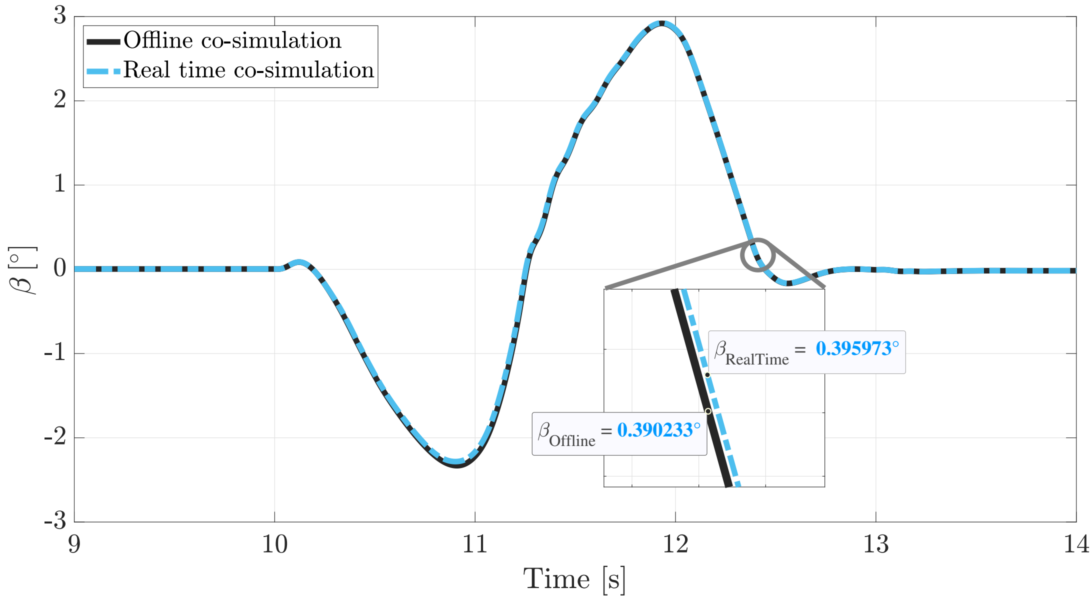


Figure 5.15: Offline and real-time co-simulation sideslip angle response to mild scenario.

As testing progresses forward, real-time constraints being met becomes a gradually more stringent requirement for the simulations to be considered successful. To that end, the real-time co-simulation instance not only produces the relevant signals for performance assessment, but also information on possible time overruns. A time overrun occurs whenever a process continues to progress beyond the expected point. In this application, a controller overrun would mean that the latter took longer than expected to compute the necessary response, i.e. violating real-time constraints. No overruns were registered in the real-time co-simulation, which encourages moving forward to hardware-in-the-loop validation.

5.4 Hardware-in-the-loop validation

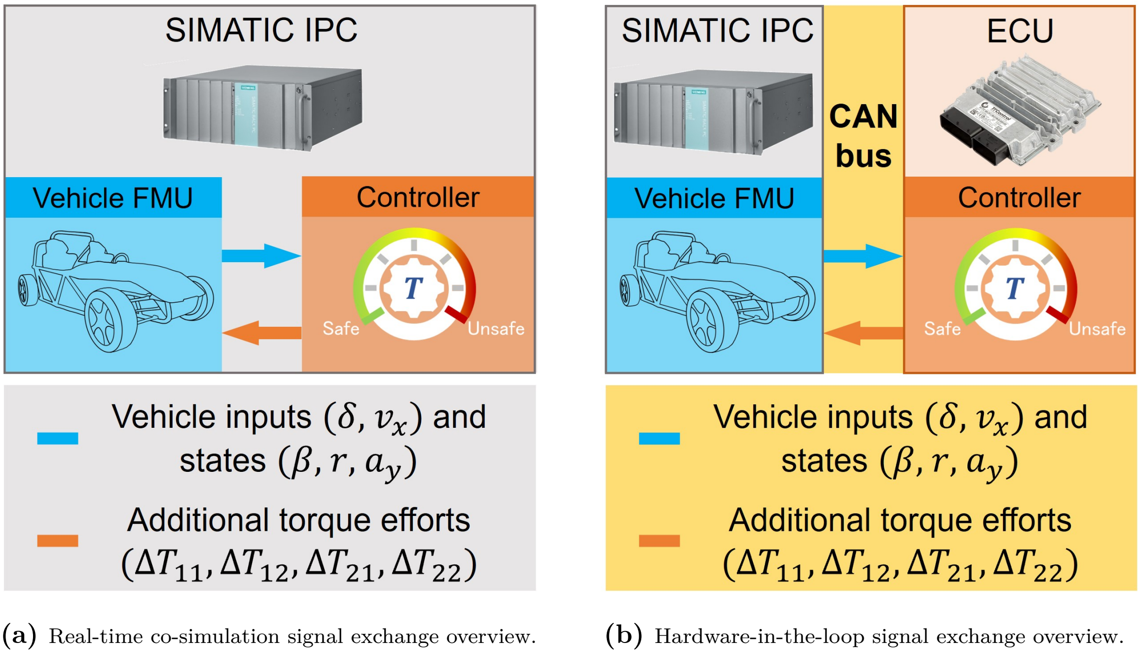
Real-time co-simulation ensured that the test bench for the proposed controller works correctly. With the aim of transitioning to a real-time controller test, the next envisioned step is to keep running the vehicle model on the IPC and deploy the

Figure 5.16: TTC-580 from TTControl (Source: TTControl ®).

controller on a dedicated platform, the aforementioned ECU. The chosen model is TTC-580 ® from TTControl [68], shown in Fig. 5.16.

The use of a communication protocol to interface different hardware modules constitutes the main difference with respect to the previous validation phase, as illustrated in Fig. 5.17. While during real-time co-simulation all signals are exchanged witin the same hardware (Fig. 5.17a), the communication between two separate hardware resources in Hardware-in-the-loop testing occurs by means of a CAN bus (Fig. 5.17b).

In order to be deployed on the ECU, the controller needs to be turned into code, specifically in hexadecimal format (`.hex`): this file is then flashed onto the ECU, which is connected to the IPC running the vehicle model, hence ready for the hardware-in-the-loop simulation instance to be called. For the control strategy to be effective, a continuous exchange of information needs to happen between the ECU and the IPC, hence a communication protocol needs to be established. In automotive applications, the choice often falls on the ISO-defined Controller Area Network (CAN) protocol [13]. CAN makes use of a serial communications bus to broadcast signals, in form of so-called messages, to the entire network of the vehicle. As they are two different pieces of hardware, the ECU is physically connected to the real-time platform by means of a DB9 cable, an analog socket with 9 pins, in charge of carrying out the CAN communication. An overview of the hardware-in-the-loop stage is provided in Fig. 5.18.


SIMATIC IPC
Vehicle FMU
Controller
T
Safe
Unsafe
Vehicle inputs (δ, v_x) and states (β, r, a_y)
Additional torque efforts $(\Delta T_{11}, \Delta T_{12}, \Delta T_{21}, \Delta T_{22})$
SIMATIC IPC
CAN bus
ECU
Vehicle FMU
Controller
T
Safe
Unsafe
Vehicle inputs (δ, v_x) and states (β, r, a_y)
Additional torque efforts $(\Delta T_{11}, \Delta T_{12}, \Delta T_{21}, \Delta T_{22})$

(a) Real-time co-simulation signal exchange overview. (b) Hardware-in-the-loop signal exchange overview.

Figure 5.17: Information exchange comparison between real-time co-simulation and hardware-in-the-loop stage.

While the ECU operates within the analog domain, the IPC works with digital signals. In order to handle CAN communication, some sort of translation code needs to be defined to handle switching from analog to digital signals and vice versa. The previously-defined controller slave is now replaced by a so-called CAN slave, an interface module in charge of setting the voltages of the RT machine output signals to a suitable range before entering the ECU, which has got its own interface signal reading block. A prompt to better visualise this is provided in Fig. 5.19: the reading block translates the incoming data using the instructions contained in a specific Data

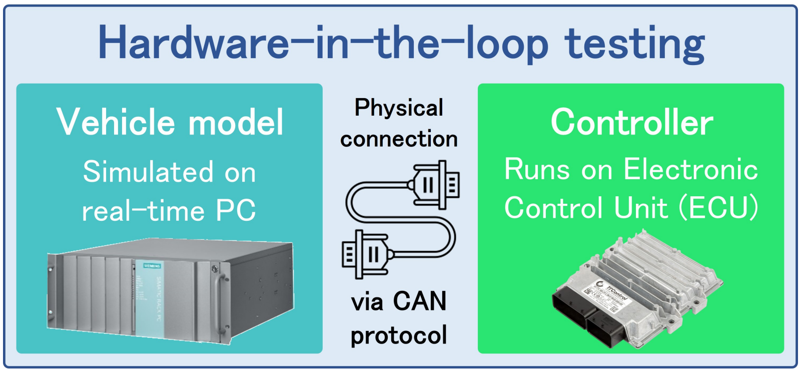

Hardware–in–the–loop testing
Vehicle model
Simulated on real–time PC
Physical connection
via CAN protocol
Controller
Runs on Electronic Control Unit (ECU)

Figure 5.18: Hardware-in-the-loop testing overview.

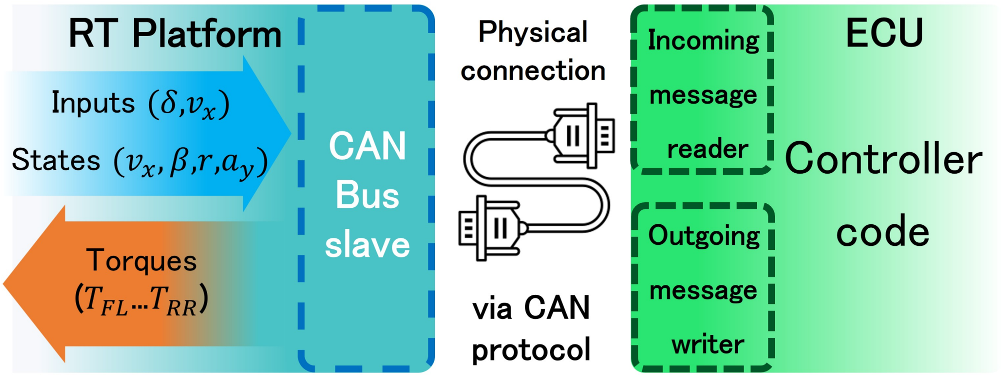


Figure 5.19: Communication flow between ECU and real-time machine.

Base Container (`.dbc`) file, listing the expected type and bit size of all the involved signals. Once the ECU has finished its operations, the output signals will be grouped in messages (whose type and size is also defined in the DBC file), translated again, written on the CAN bus, and read by the RT machine.

For the model stability reasons explained in Section 3.2.2, the vehicle model timestep is left to 1ms, while in hopes of achieving an immediate real-time result, the controller timestep is set to 10ms to relieve the computational burden of the control strategy. Once the setup is complete, meaning the controller code has been flashed on the ECU and the vehicle model is still loaded as a slave module on the real-time platform along with the CAN bus module, a simulation instance can be called.

5.4.1 Results

To avoid redundancy of plots that at a glance look very similar, hardware-in-the-loop validation results are shown in comparison to the ones obtained for the offline and real-time co-simulations. Figure 5.20 depicts the controlled sideslip angle signal in the challenging scenario for all three simulation instances analysed thus far.

Immediately the overall closeness of the three signals can be observed. The zoomed-in portion displays the magnitude discrepancy between the sideslip angle β at all three testing stages, observed at the same time instant, around the order of 10^{-2}. The gap between the offline/real-time co-simulation and the hardware-in-the-loop signal is most likely due to the controller timestep being decreased to 10ms.

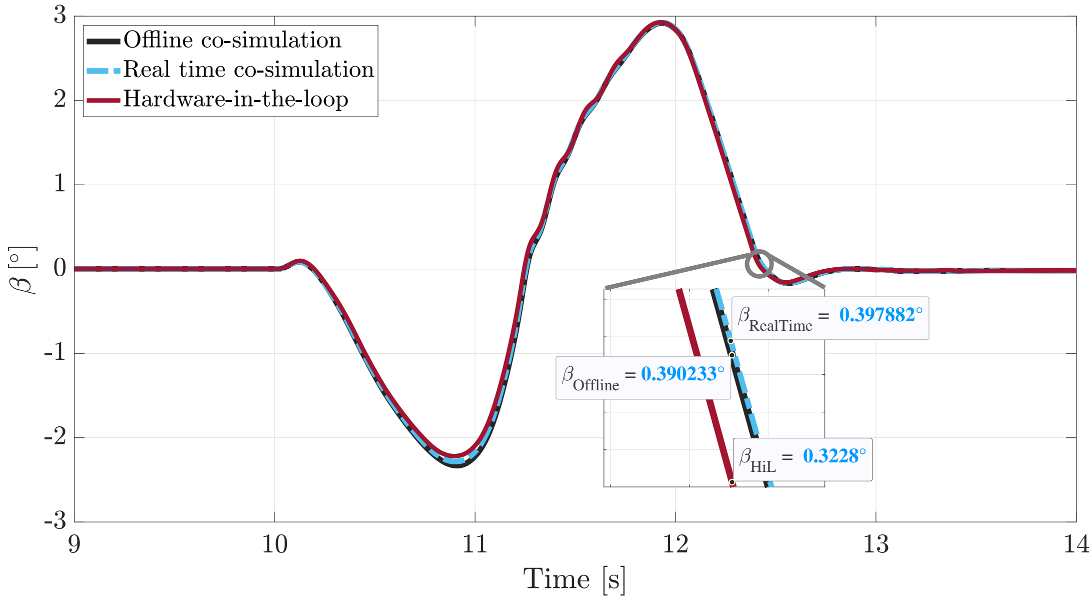


Figure 5.20: Offline co-simulation, real-time co-simulation and hardware-in-the-loop testing sideslip angle response to mild scenario.

As stated earlier in this section, the timestep of the controller was decreased to ensure compliance to the real-time constraints. To guarantee the latter are actually met, a study of the registered execution time is performed, and conclusions on the matter are drawn in the following subsection.

5.4.2 Execution time

Similarly from what was observed from a real-time co-simulation, calling a hardware-in-the-loop test instance also yields insightful information on how long it takes for each slave to complete their tasks at every time step. This time length is hereby labelled execution time.

The sampling time constitutes a rigorous upper bound for the controller to complete a single task. If the latter takes longer than the sampling time to finish, that instance is said to be an overrun (Section 5.3). The absence of overruns is hence definite proof that the controller is able to operate in real time. The ability to check both performance and timing of signals to ensure real time capabilities highlights the value of executing hardware-in-the-loop testing. Satisfying the real-time requirements ultimately means checking that the time needed to execute the controller tasks is lower than the sampling time. The controller execution time should be analysed paired with the co-simulation master execution time, since ultimately their joint computational effort is subject to the real time criteria. For this reason, the overall execution time of interest is hereafter called ECU execution time. The time horizon for completion is 10 ms, as that is the controller sampling time. It should be noted that the execution time may be checked for the vehicle model as well, but that is not relevant as the full-scale application will not rely on a vehicle model but on the vehicle itself: the focus is thus placed solely on the ECU.

The ECU execution time is evaluated over a time horizon corresponding to the controller sampling time. The overall computational effort required from the ECU in the challenging scenario is 0.459 ms. Considering that the available time frame is 10 ms, the controller proves to operate significantly below the predefined threshold. This means that the controller takes less than 5% of the available time frame to complete its task.

The considerable time margin available for the controller prompts two immediate conclusions:

- **Further complexity may be introduced in the controller.** Section 3.3.1 described the high-level controller as a PI controller with gain scheduling. Despite it adapting to the vehicle speed this is still a very simple control strategy, selected to ensure low computational complexity and compliance to real-time constraints. Given that there is still significant time margin available, further complexity may be introduced in the control strategy, e.g. a Linear Quadratic Regulator (LQR) control scheme or implementing Model Predictive Control (MPC). The ECU execution time can be checked throughout the controller progression phases, to ensure real-time constraints are met.

- **The controller is real-time capable.** Provided that thorough testing is performed on a wide variety of scenarios to further validate its performance, it can potentially be deployed on a full-scale vehicle.

Chapter 6

On-vehicle validation: sideslip angle estimation

Despite time limitations within the project hindering a full-vehicle validation for the entire controller, this stage was reached with the interpolation-based sideslip estimator described in Section 4.2 by means of two experimental campaigns, whose details follow.

6.1 Vehicle setup

Experimental tests were conducted on a production sedan car manufactured by Alfa Romeo, more specifically the Alfa Romeo Giulia, equipped with sensors and actuators as specified in the following list:

- **Inertial Measurement Unit (IMU).** The IMU measures the car's acceleration and angular velocity. It provides data to the car's stability control system, allowing it to affect real-time vehicle stability and handling. Here, information coming from the IMU is enriched by means of a gyroscope and a GPS (RMS FES44), providing respectively roll, pitch, and yaw rates/accelerations and insight on the vehicle position.

- **Potentiometers.** They are mounted on the chassis behind the wheel and connected to the wheel hub via extensible metal wires, with the purpose of measuring the elongation of front suspensions.

- **Steering Wheel Torque and Angle measurement (Kistler MSW).** This sensor is mounted on the steering wheel to measure the torque applied on the steering wheel as well as the angle of rotation. The static part is fixed through a windshield sucker mount, while the dynamic part is a dummy steering wheel, replacing the original one.

- **Front Dampers stroke measurement (TE SP2-12).** One stroke measurement device per wheel is present, featuring two tie-rod load cells. They are designed to measure the tensile and compressive force applied to the tie-rod of a car.

- **Sideslip angle optical sensor (Kistler S-motion).** This device is mounted on the front and devoted to the measurement of the sideslip angle of the vehicle. It operates with an acquisition frequency of 250Hz, making it a very powerful tool to look into manoeuvres with fast-changing dynamics. It also contains accelerometers and a GPS.

- **Wheel force transducers (Kistler RoaDyn).** They are encoders whose stator is fixed by a thin beam anchored to the vehicle body through suction cups, while the rotor is located on the wheel itself (Fig. 6.1). Six load cells are equally spaced along the wheel hub, labelled A to F, and their rotations and moments are reported to a single frame of reference located in the very middle of the wheel (i.e., of the rotor). As for the wheel orientation and speed, the axis of reference is the one crossing load cell points A and F.

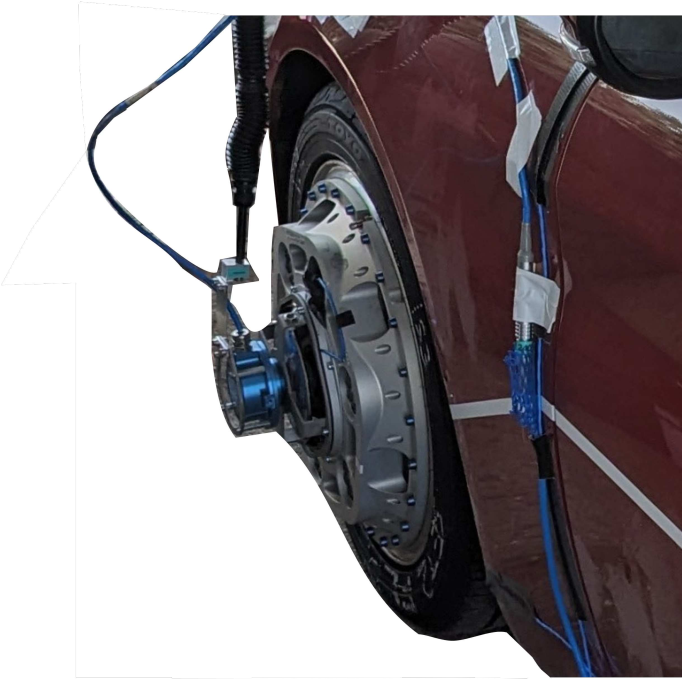


Figure 6.1: Kistler ® RoaDyn wheel force transducers.

Data collection activities demand the presence of a dedicated hardware that can acquire and synchronise both digital and analog signals. The devoted platform for this testing activity was DEWESoft ® DEWE-43A, a Data Acquisition (DAQ) system whose main features follow:

- 8 universal analog inputs, capable of a sampling frequency of 500Hz. This logs data coming from tie-rod load cells, accelerometers, and wire potentiometers. Each used port needs a mention within the configuration file, which is then integrated within the imported `dbc` file.

- 8 digital/counter/encoder inputs.

- 2 high-speed CAN bus inputs. Two additional CAN ports are provided by an Extension logger (DS-CAN 2), connected through a sync cable as slave to the master DEWE-43A. Signals being circulated via CAN are the ones coming from the IMU, GPS, sideslip optical sensor, steering wheel sensor, and wheel force transducers. Another CAN channel is reserved for debug: it logs the On-Board Diagnostics (OBD) data coming from the vehicle CAN.

- Presence of a dedicated software, DewesoftX ®, allowing acquiring and combining of data from additional interfaces like GPS, Flexray, XCP, Ethernet, Serial, PCM telemetry etc. This software is not only the gateway to perform data logging, but it also features a user interface where signals can be displayed and recorded.

Additionally, a dedicated platform is needed to deploy and test the interpolation-based estimator as well as acting as ECU. A perfect candidate for the task was found in dSPACE MicroAutoBox II (MAB II) ® [69], a powerful real-time system designed for testing automotive and mechatronic applications. MAB II is equipped with a high-performance multi-core processor and is hence capable of handling complex control tasks with low latencies, making it a suitable candidate to validate real time-capable algorithms. Moreover, MAB II comes with ControlDesk, a companion software enabling online simulation monitoring as well as changing selected (so-called exposed) parameters on the fly, without having to modify and re-flash the MAB II with a new piece of code. Carrying out the control tasks implies that signals of interest for that control action need to be logged to MAB II as well as the DEWESoft logger.

Not all sensors are logged to MAB II, but all of them are logged to DEWESoft logger. Since MAB II only has one input CAN port, in order to input additional sets of sensors, a check on the `.dbc` file was performed to ensure there was available space on that CAN bus to accommodate additional signals. Moreover, CAN splitters were used for the sensors to log both on the DEWESoft logger and to MAB II. The final CAN splitter schematic is shown in Figure 6.2.

The Kistler S-motion ® sensor was mounted on the front tow eye, which means that all measurements will be located in a frame of reference centred at the front of the vehicle. On the other hand, the sideslip angle estimate uses data coming from the IMU, located elsewhere within the vehicle frame. For the measured and estimated signal to be fairly compared, the extra roll and yaw dynamics due to the different locations of measurement devices have to be filtered out of the signals. The lateral and longitudinal velocities located at the front sensor are then corrected using the

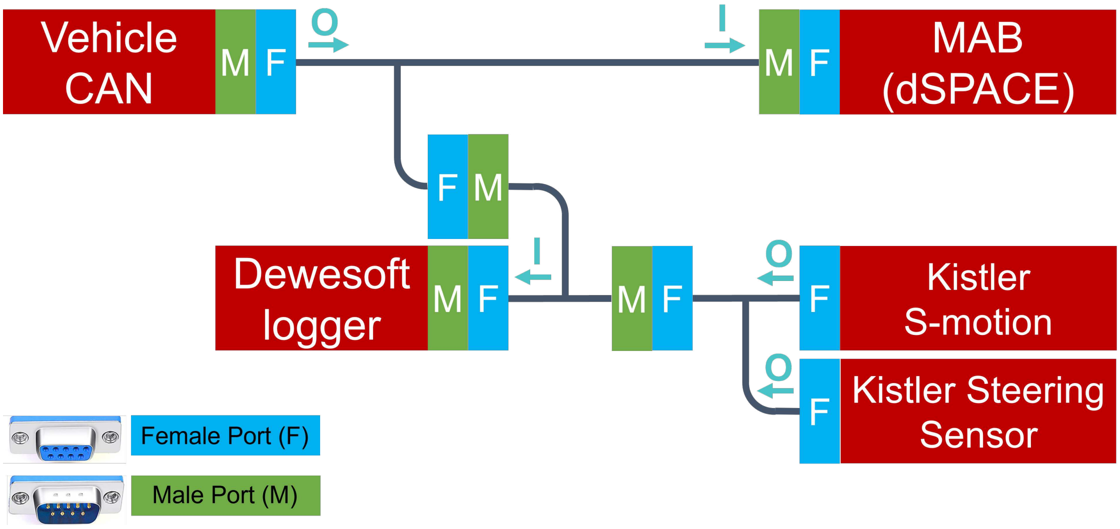


Figure 6.2: CAN splitters schematic to accommodate sensor information both on DEWE-Soft and dSPACE. Outputs are indicated with "O", while inputs are marked as "I".

yaw rate measured at the IMU and the longitudinal and lateral distance between the IMU and the S-motion, respectively $d_{long} = 1.63$ m and $d_{lat} = 0.46$ m. A top view of the location of both sensors is provided in Fig. 6.3, where $v_{x,\mathrm{f}}$ and $v_{y,\mathrm{f}}$ are the longitudinal and lateral velocities centred in the S-motion, while $v_{x,\mathrm{IMU}}$, $v_{y,\mathrm{IMU}}$ and r_{IMU} are respectively the longitudinal and lateral velocity and the yaw rate centred in the IMU.

Recalling Eq. (2.1), expressing the sideslip angle β as approximately equal to the ratio between the lateral and longitudinal velocity v_x and v_y, the reference frame correction can be operated by means of the latter two. Considering the velocity components defined in Fig. 6.3 at both locations of interest, Eq. (6.1) may be written:

$$\begin{aligned} v_{x,\mathrm{IMU}} &= v_{x,\mathrm{f}} - r_{\mathrm{IMU}} d_{lat} \\ v_{y,\mathrm{IMU}} &= v_{y,\mathrm{f}} - r_{\mathrm{IMU}} d_{long} \end{aligned} \tag{6.1}$$

Recalling Eq. (2.1), the measured sideslip angle evaluated at the IMU location may be written:

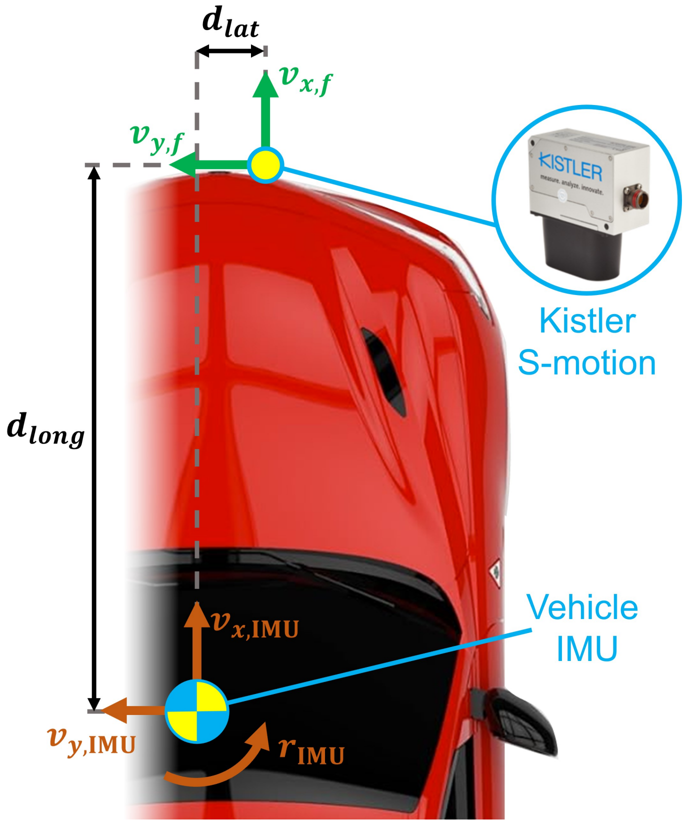


Figure 6.3: Top view of vehicle IMU and Kistler S-motion ® location on test vehicle.

$$\beta_{\mathrm{IMU}} = \frac{v_{y,\mathrm{f}} - r_{\mathrm{IMU}} d_{long}}{v_{x,\mathrm{f}} - r_{\mathrm{IMU}} d_{lat}} \tag{6.2}$$

Notice that the approximately equal symbol in Eq. (2.1) is hereby replaced with regular equal symbol, since the operating framework at hand considers the small-angle assumption to be true.

Estimates only make use of data coming from vehicle CAN, whose sampling frequency is 10Hz: therefore, each considered signal coming from other devices using a higher sampling frequency were downsampled. A fair comparison can then be established between the estimated and measured signal.

6.2 First test session

The first testing campaign was performed in Battipaglia (Italy) at the Circuito del Sele test track (Fig. 6.4) within Seaside Racing, a week-long testing event occurring every year and planned by the MegaRide group collaborating with University of Naples Federico II. The fully-instrumented Alfa Romeo Giulia was provided by Meccanica 42, and performed double lane-change and sine sweep manoeuvres, as well as recording handling data throughout numerous consecutive laps on the test track.

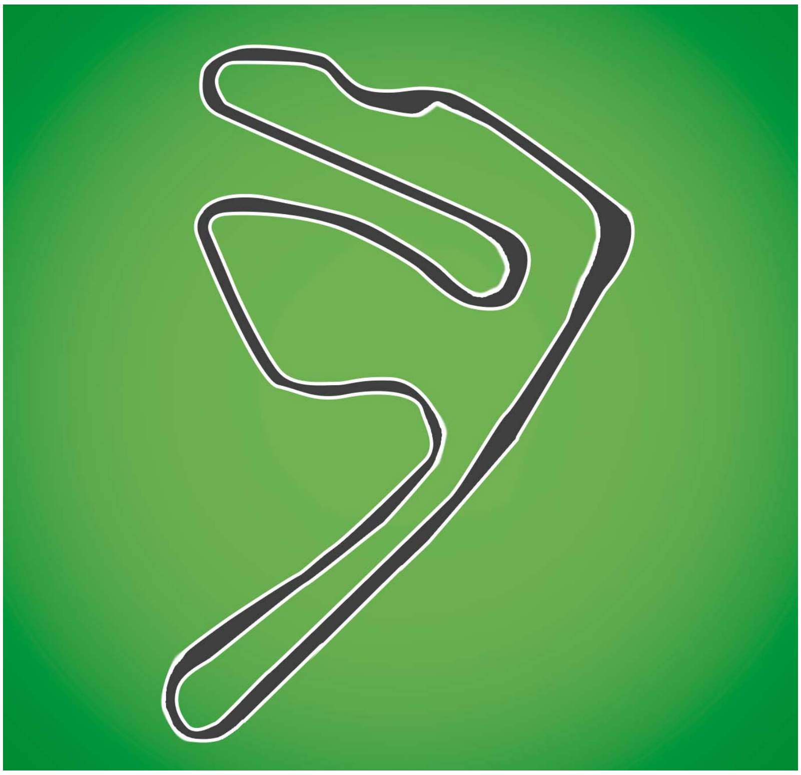


Figure 6.4: Bird's eye view of Circuito del Sele.

The estimator was flashed on the MAB II as exported code from Simulink ®, containing a simulation-based set of initial interpolation coefficients while setting them as exposed parameters to be able to change them on the fly from ControlDesk ®.

The Simulink ® block scheme featured two inputs, the lateral acceleration a_y (m/s^2) and the steering angle δ (°), and one output, which is the sideslip angle β (°). Within the block itself, a_y is used to compute the dynamic sideslip angle β_{dyn} using Eq. (4.9), while the steering angle δ is involved in the computation of the kinematic sideslip angle β_{kin} through Eq. (4.2). The two components then come together as a sum to form the sideslip angle β. The overall estimator architecture is summarised in Fig. 6.5, where inputs are marked in orange while the output is marked in blue.

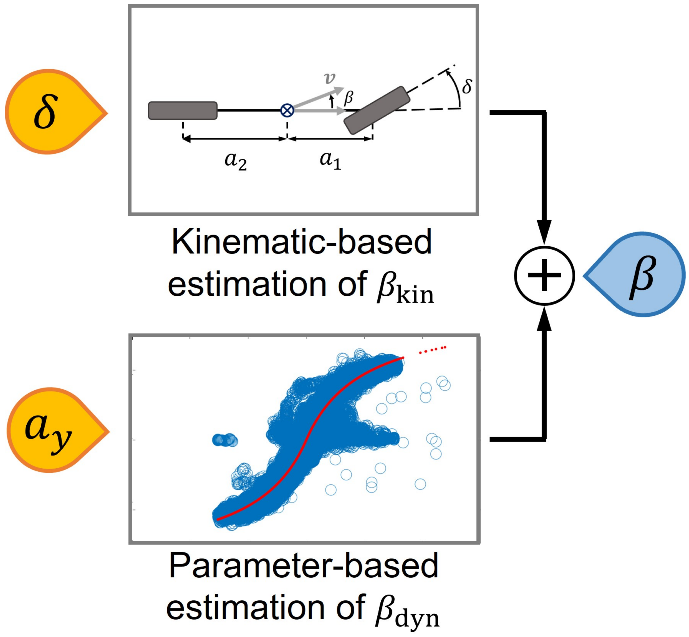

δ
v
β
δ
a_2
a_1
Kinematic-based estimation of β_{kin}
+
β
a_y
Parameter-based estimation of β_{dyn}

Figure 6.5: Estimator overview: orange marks the inputs, while blue marks the output.

6.2.1 Parameter assessment

Eq. (4.9) requires the knowledge of the interpolation parameters $c_{1\,\ldots 3}$: however, no previous experimental data is available for an Alfa Romeo Giulia to sensibly tune the three parameters. A first approximation is then set by using simulation data. The obtained interpolation parameters are reported in the first row of Table 6.1.

As the first set of interpolation coefficients was an unreliable approximation, a re-calibration was performed on the first available dataset, relative to a double lane-change and labelled D0 hereafter. Interpolation parameters coming from this actual run are collected in the second row of Table 6.1. Immediately the huge difference in magnitude for the parameters may be noticed: c_1 is about half of what it was expected to be, and c_2 is approximately three orders of magnitude lower than for the simulation.

At this stage, interest resides in understanding whether using different training instances leads to interpolation coefficients differing to a similar extent than they did in Table 6.1. To that end, a selection of different driving scenarios among the collected data was operated:

1. Handling (D1). Regular driving instance on test track.

Table 6.1: Simulation-obtained interpolation parameters for Alfa Romeo Giulia c_{sim}, compared to those obtained for an actual driving instance using dataset D0.

Training dataset	c_1	c_2	c_3
c_{sim}	-44556	2870.5	10831
D0	-20004	9.8560	10363

Table 6.2: Difference in coefficients when performing interpolation on different datasets.

Training dataset	c_1	c_2	c_3
D1	-18872.030	1199.411	8281.683
D2	-26917.040	1229.671	12626.937
D3	-17763.633	10.036	12296.315

2. Double lane-change (D2). Conducted with an extra load of around 200kg on the backseat.

3. Sine sweep (D3). Chosen for its very high dynamic content and for the challenge of identifying small sideslip angles.

Interpolation coefficients are displayed using each of the three as training dataset in Table 6.2, and they are validated on the remaining two datasets regarded as validation data.

Entries in Table 6.2 are particularly interesting, as they provide insight on the expected accuracy of coefficients during their retrieval. While c_1 and c_3 entries are both similar to each other and to the experimental one (D0, Table 6.1), c_2 is very similar to the experimental one for D3, but quite far from it for both D1 and D2, which instead are more similar to c_2 for c_{sim} (Table 6.1).

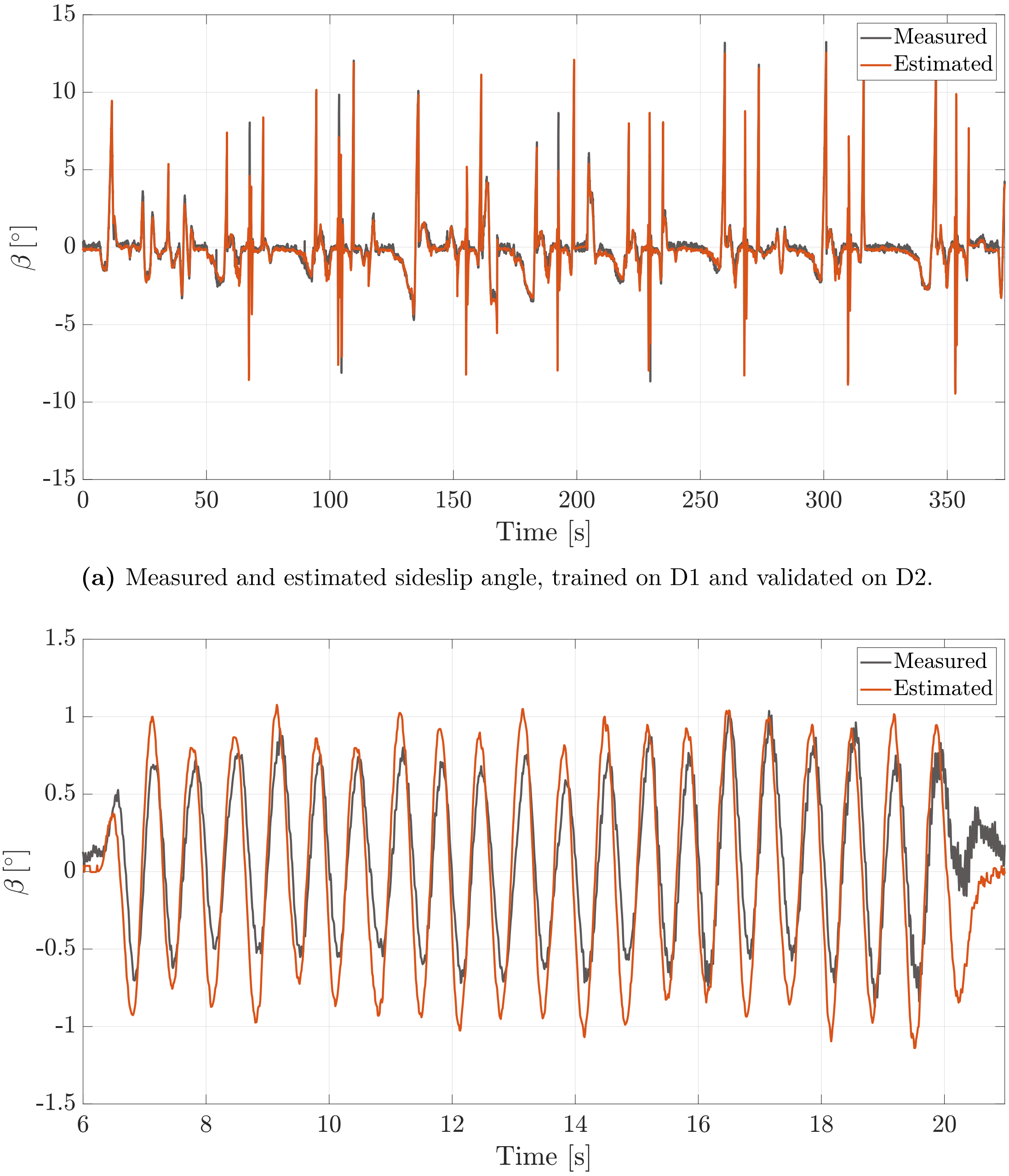
Table 6.3: Validation outcomes for different interpolation coefficients.

Training dataset	Validation dataset	RMSE (°)
D1	D2	0.586
	D3	0.338
D2	D1	0.578
	D3	0.168
D3	D1	0.649
	D2	0.617

Despite a re-tuning of the parameters being necessary to perform a thorough assessment of the estimation strategy, time-related issues within testing prevented that from happening. Consequently, results will be presented in terms of offline signal comparison. Moreover, albeit flawed in its assessment, the interpolation-generated estimate will still be valuable to determine real-time feasibility.

Before looking at data logs coming from MAB II, offline validation was performed using data recorded from the DEWESoft ® DAQ. This was useful to check what order of magnitude of discrepancy could be expected between the measured and estimated signal, if the training coefficients were those listed in the rows of Table 6.2 and they were validated against the other two remaining datasets. RMSEs coming from training on all three datasets are compared in Table 6.3, showing a comparatively small discrepancy between the measured and estimated signal. Given the flawed nature of the RMSE as a performance indicator, comparative plots for all six cases listed in Table 6.2 are presented respectively when training on D1 (Fig. 6.6), on D2 (Fig. 6.7) and D3 (Fig. 6.8).

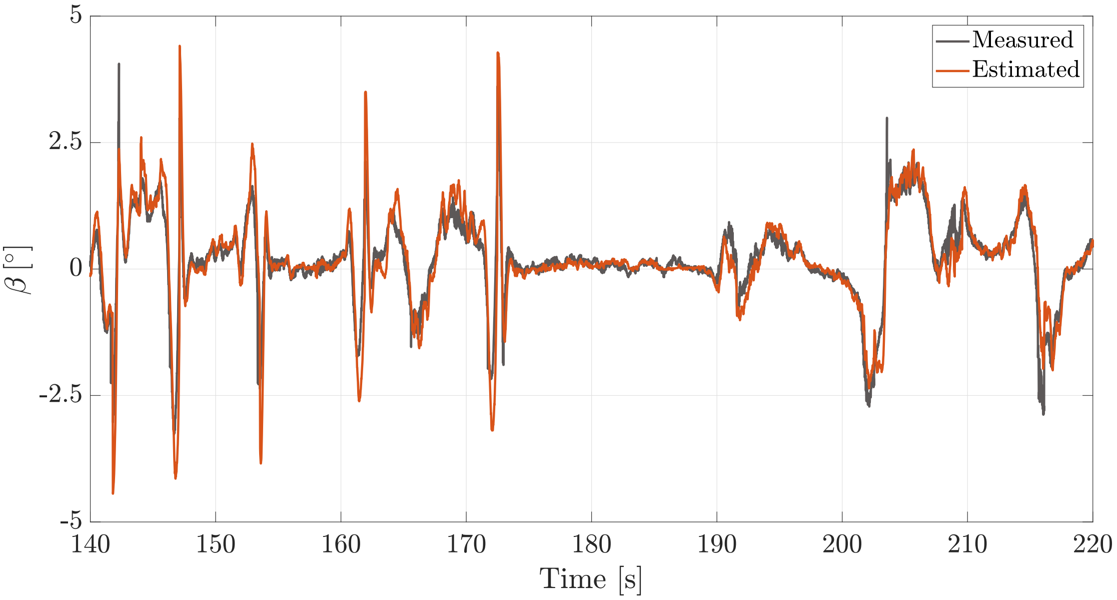
Both datasets D1 and D2, referring to a regular driving instance and to a double lane-change manoeuvre respectively, produce satisfactory estimates. They both



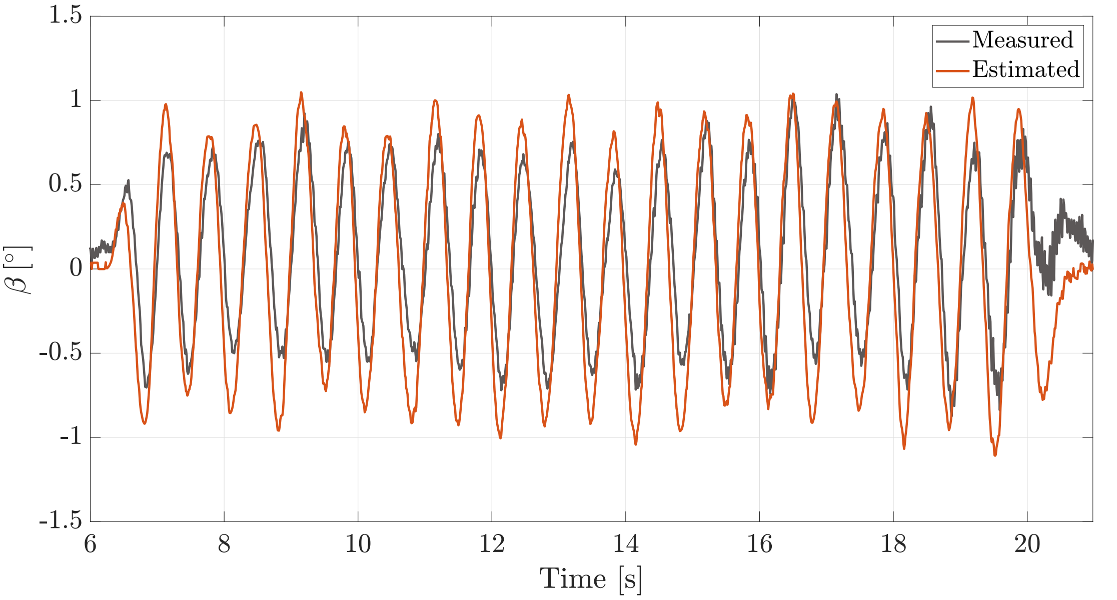
(a) Measured and estimated sideslip angle, trained on D1 and validated on D2.

(b) Measured and estimated sideslip angle, trained on D1 and validated on D3.

Figure 6.6: Validation outcomes when training on dataset D1.

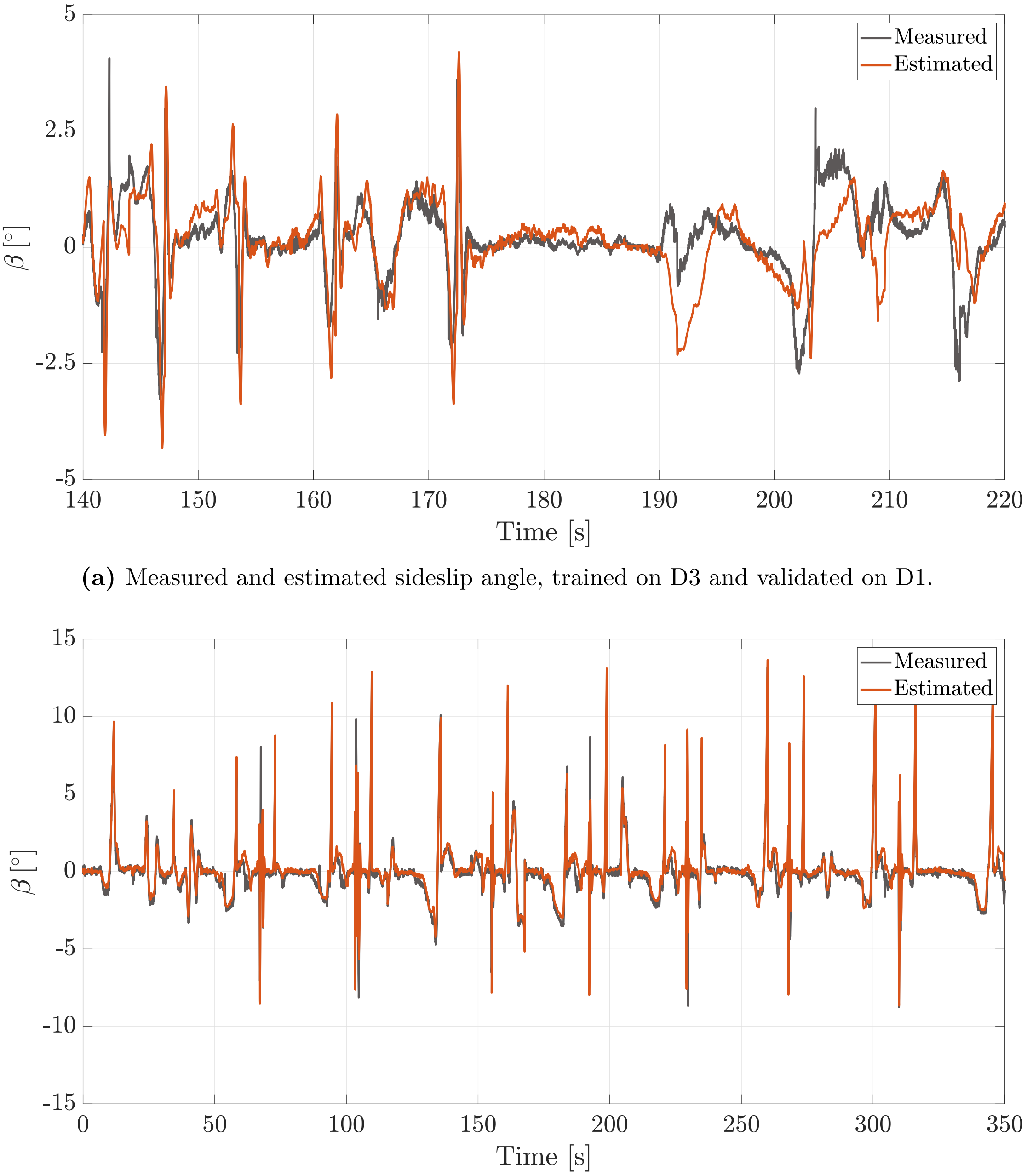


(a) Measured and estimated sideslip angle, trained on D2 and validated on D1.



(b) Measured and estimated sideslip angle, trained on D2 and validated on D3.

Figure 6.7: Validation outcomes when training on dataset D2.



(a) Measured and estimated sideslip angle, trained on D3 and validated on D1.

(b) Measured and estimated sideslip angle, trained on D3 and validated on D2.

Figure 6.8: Validation outcomes when training on dataset D3.

perform very well when validated against each other: Fig. 6.6a displays the estimator ability to track highly-dynamic peaks, and a similar result is demonstrated in Fig. 6.7a. As for their performance against D3, there is some visible inaccuracy in tracking the peaks, which is also due to the D3 signal not presenting a very high magnitude. Frequency-wise, the performance is still satisfactory.

Dataset D3 appears to be the least appropriate to train on. The poorly-performing outcome could have been predicted from the very low magnitude of the corresponding sideslip angle signal, observed in grey in Fig. 6.6b and Fig. 6.7b. A lack of proper dynamic variety within the signal prevents the code from finding adequate interpolation parameters when training.

6.2.2 Real-time feasibility check

Upon completion of offline checks, the collected data are analysed to assess whether the proposed strategy is able to handle real-time constraints.

The only incoming source of signals MAB II deals with is the vehicle CAN, as mentioned in Section 6.1. As the sampling frequency for those signals is 10Hz, this gives MAB II an available execution time of 0.1s. To make sure that the resulting estimation algorithm did not incur in any overruns, the overrun strategy on MAB II was left to its default action, which is “Stop simulation”. If a longer time than the allocated one is required for the controller to finish the assigned task, the next instance will not begin but rather cause the simulation to stop abruptly. Data was smoothly recorded on MAB II with no interruptions, which indicates the algorithm ran in real time. A further sensible check is to look at the variable `overrunCount`, devoted to counting the number of overruns: the displayed value was zero for all recorded instances. This leads to the firm belief that this estimation strategy is feasible in real time.

Reasonably, the signal comparison between the measured and the and estimated sideslip angle is reported in Fig. 6.9, with data recorded during run instance D1.

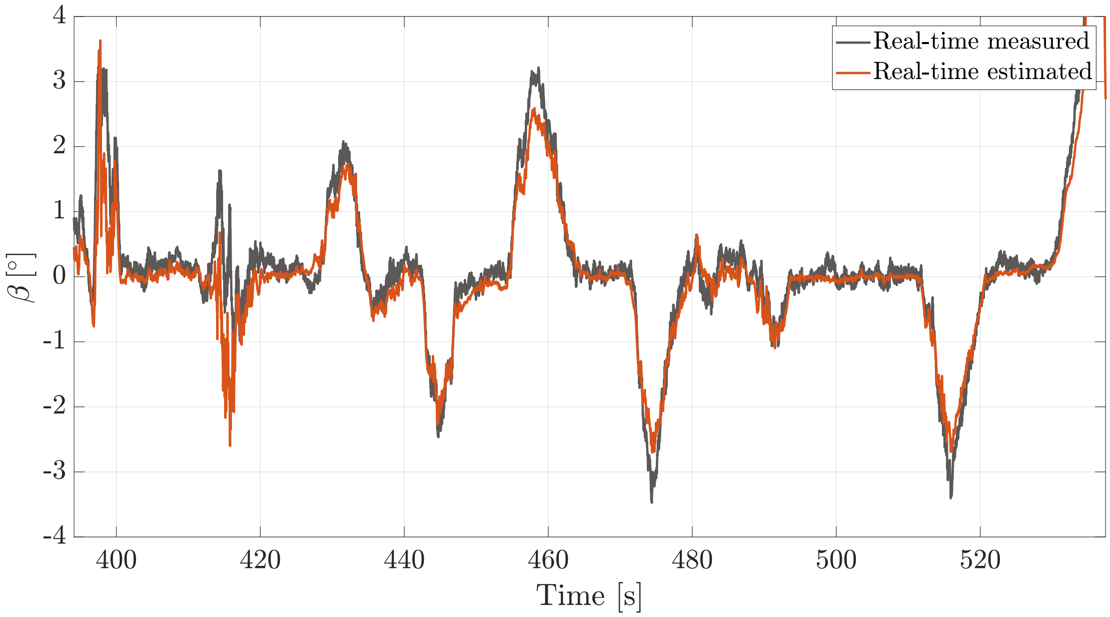


Figure 6.9: Real-time interpolation outcome for handling dataset.

The estimate quality appears to be satisfactory, especially considering that it was retrieved by using solely information coming from the vehicle IMU, operating at a much lower frequency than the one used by higher-end sensors. The cheap yet reasonably reliable nature of the analysed estimator may be an incentive to implement this simple strategy to gain sideslip angle knowledge without requiring additional, expensive sensors. Moreover, the interaction with the overall controller framework is guaranteed to be successful because of its real-time applicability.

6.3 Second test session

A second test session took place at a dedicated testing space within the renowned Mugello Circuit (Italy), using the same fully-instrumented Alfa Romeo as the first experimental campaign. The overall estimator schematic and upload procedure remains the same as the one described in Section 6.2.

6.3.1 Parameter assessment

In preparation to the experimental campaign, a few changes were made to the estimator to ensure a smoother and more insightful testing. Firstly, the interpolating function was studied to infer boundary conditions on its parameters, aiming to bring further value to the proposed estimation methodology. A new formulation was then conjured for the estimator, reducing the number of parameters from three to two, aiming to facilitate function analysis. To do so, a simple algebraic manipulation is performed: the denominator parameters c_2, c_3 in Eq. (4.8) are divided by the numerator parameter c_1. Eq. (6.3) is then obtained, where $p_1 = c_2/c_1$ and $p_2 = c_3/c_1$.

$$a_y = -\frac{\beta_{dyn}}{p_1|\beta_{dyn}| + p_2} \tag{6.3}$$

In the denominator of Eq. (6.3), $\sqrt{(\beta_{dyn})^2}$ was substituted with $|\beta_{dyn}|$, as the two are mathematically equivalent. This ensures that the expected behaviour is maintained, and provides a more tractable formulation for function analysis. The expected behaviour for the interpolating function is shown in Fig. 6.10.

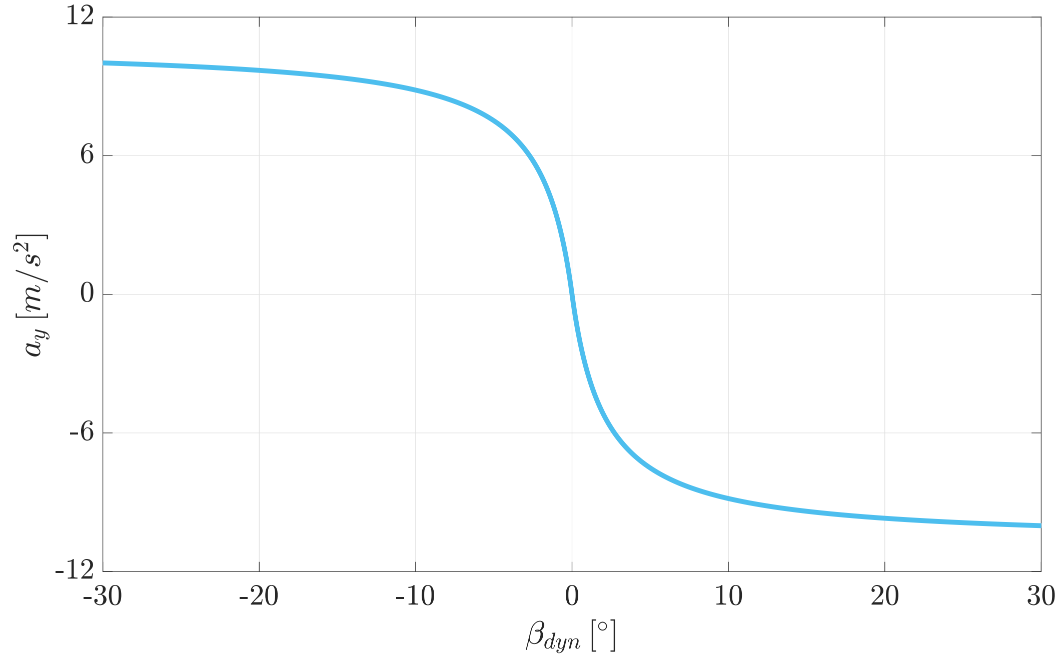


Figure 6.10: Expected behaviour between the dynamic sideslip angle β_{dyn} and the lateral acceleration a_y, according to Eq. (6.3).

Firstly, the horizontal asymptotes are looked for: Eq. (6.4) - (6.5) show the computation of the function limits as the independent variable β_{dyn} approaches positive and negative infinity, respectively.

$$\lim_{\beta_{dyn}\to+\infty} -\frac{\beta_{dyn}}{p_1|\beta_{dyn}|+p_2} = -\frac{1}{p_1} \tag{6.4}$$

$$\lim_{\beta_{dyn}\to-\infty} -\frac{\beta_{dyn}}{p_1|\beta_{dyn}|+p_2} = \frac{1}{p_1} \tag{6.5}$$

For the mathematical results to match what is visually expected from Fig. 6.10, the horizontal asymptote must be negative for β_{dyn} approaching positive infinity and positive for β_{dyn} approaching negative infinity: this calls for p_1 to be positive.

Moving forward, more can be inferred by computing the first and second derivative of the function at hand (Eq. 6.6 and Eq. 6.7 respectively).

$$\begin{aligned}
a_y'(\beta_{dyn}) &= \frac{-(p_1|\beta_{dyn}|+p_2)+\beta_{dyn}(p_1\mathrm{sgn}(\beta_{dyn}))}{(p_1|\beta_{dyn}|+p_2)^2} = \\
&= \frac{\cancel{-p_1|\beta_{dyn}|}-p_2+\cancel{p_1|\beta_{dyn}|}}{(p_1|\beta_{dyn}|+p_2)^2} = \\
&= -\frac{p_2}{(p_1|\beta_{dyn}|+p_2)^2} \qquad (6.6)\\
a_y''(\beta_{dyn}) &= -p_2\cdot\left(-\frac{2}{(p_1|\beta_{dyn}|+p_2)^3}\right)\cdot p_1\mathrm{sgn}(\beta_{dyn}) = \\
&= \frac{2p_1p_2\mathrm{sgn}(\beta_{dyn})}{(p_1|\beta_{dyn}|+p_2)^3} \qquad (6.7)
\end{aligned}$$

The second derivative gives direct insight in the concavity of the function, which is expected to be positive for a positive dynamic sideslip angle β_{dyn}, negative otherwise. It should be noted that the second derivative is not defined in $\beta_{dyn}=0$ due to the presence of the sign function at the numerator. The study of the sign of the second derivative was consequently performed.

- Numerator:

$$2p_1p_2\text{sgn}(\beta_{dyn}) > 0 \Longrightarrow$$
$$\Longrightarrow \begin{cases} p_1p_2 > 0 & \text{if} \quad \beta_{dyn} > 0 \\ p_1p_2 > 0 & \text{if} \quad \beta_{dyn} < 0 \end{cases} \tag{6.8}$$

- Denominator:

$$(p_1\,|\beta| + p_2)^3 > 0 \Longrightarrow$$
$$\Longrightarrow \begin{cases} \beta_{dyn} > -\dfrac{p_2}{p_1} & \text{if} \quad \beta_{dyn} > 0 \\ \beta_{dyn} < \dfrac{p_2}{p_1} & \text{if} \quad \beta_{dyn} < 0 \end{cases} \tag{6.9}$$

According to the expected behaviour in Fig. 6.10, the function $a_y(\beta_{dyn})$ must have positive concavity for $\beta_{dyn} > 0$, and negative concavity for $\beta_{dyn} < 0$. Two cases are consequently distinguished:

1. $a''_y(\beta_{dyn}) > 0$.

   Eq. (6.8) - (6.9) reveal that for $\beta_{dyn} > 0$, p_1 and p_2 must have the same sign to ensure positive concavity. The evaluation of the sign of the concavity is performed in Fig. 6.11a.

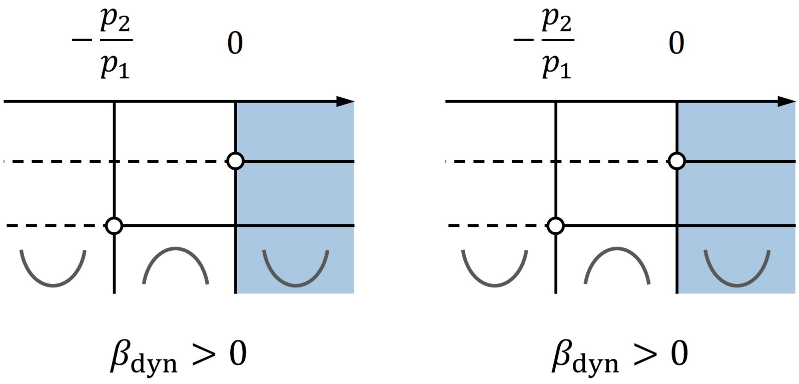
2. $a''_y(\beta_{dyn}) < 0$.

   For $\beta_{dyn} < 0$, p_1 and p_2 must have the same sign to ensure positive concavity, otherwise an undesired inflection point would appear. The evaluation of the sign of the concavity is performed in Fig. 6.11b.

To summarise, the following can be gathered from studying the interpolation function:

- p_1 must be positive for the asymptotes to match their expected sign.
- p_1 and p_2 must have the same sign for the second derivative to match 5.3.1the expected concavity.

This leads to the conclusion that both p_1 and p_2 must be positive numbers.



(a) Case 1.

(b) Case 2.

Figure 6.11: Concavity assessment for both examined cases.

On a more practical note, the obtained boundary conditions were used to solve a nonlinear curve fitting problem, whose outcome was the parameter set (p_1, p_2) yielding the best fit to the measured data points mapping the dynamic sideslip angle signal to the measured lateral acceleration. The curve fitting is performed by solving a nonlinear least-squares problem, using the Levenberg-Marquardt algorithm on MATLAB ®. For the dynamic sideslip angle to be retrieved using the best fit parameters and the measured lateral acceleration, Eq. (6.3) must be inverted. Using the expected behaviour in Fig. 6.10 as reference, it is easy to infer that a positive β_{dyn} is related to a negative a_y. Conversely, a negative β_{dyn} is related to a positive a_y. The inverse function may then be computed as in Eq. (6.10), by means of a piecewise formulation distinguishing $\beta_{dyn} > 0$ (hence $a_y < 0$) and $\beta_{dyn} \leq 0$ (hence $a_y \geq 0$). Notice that the possibility of the function being zero is hereby included, as the function itself is continuous in zero.

$$\beta_{dyn} = \begin{cases} -\dfrac{p_2 a_y}{1 + p_1 a_y} & \text{if} \quad a_y < 0 \\ -\dfrac{p_2 a_y}{1 - p_1 a_y} & \text{if} \quad a_y \geq 0 \end{cases} \tag{6.10}$$

It is worth remarking that the interpolation is performed offline using a sample dataset, recorded on the vehicle of interest, with a dynamic range wide enough that the interpolation may be considered representative of the vehicle lateral behaviour

Table 6.4: Experimental interpolation parameters.

Parameter	p_1	p_2
Value	0.0748	0.2502

(e.g.: a ramp steer manoeuvre). The resulting interpolation parameters p_1, p_2 from such trial runs are reported in Table 6.4.

Finally, as in the first test campaign, the kinematic component of the sideslip angle was computed using Eq. (4.7) and the overall sideslip angle β was reconstructed as sum of the dynamic and kinematic signal.

6.3.2 Results

Results are shown for a list of selected manoeuvres, namely:

- Circle manoeuvre, 16.7 m/s;
- Sine with dwell manoeuvre, 11 m/s;
- Double lane-change manoeuvre, 11 m/s;
- Slalom manoeuvre, 11 m/s;
- Slalom manoeuvre, 16.7 m/s;
- Ramp steer manoeuvre, 8.3 m/s;

The main quantities framing each manoeuvre are shown, along with a comparison between the measured and the total real-time estimated sideslip angle, both reported to the IMU using Eq. (6.2). The kinematic and dynamic components are also presented.

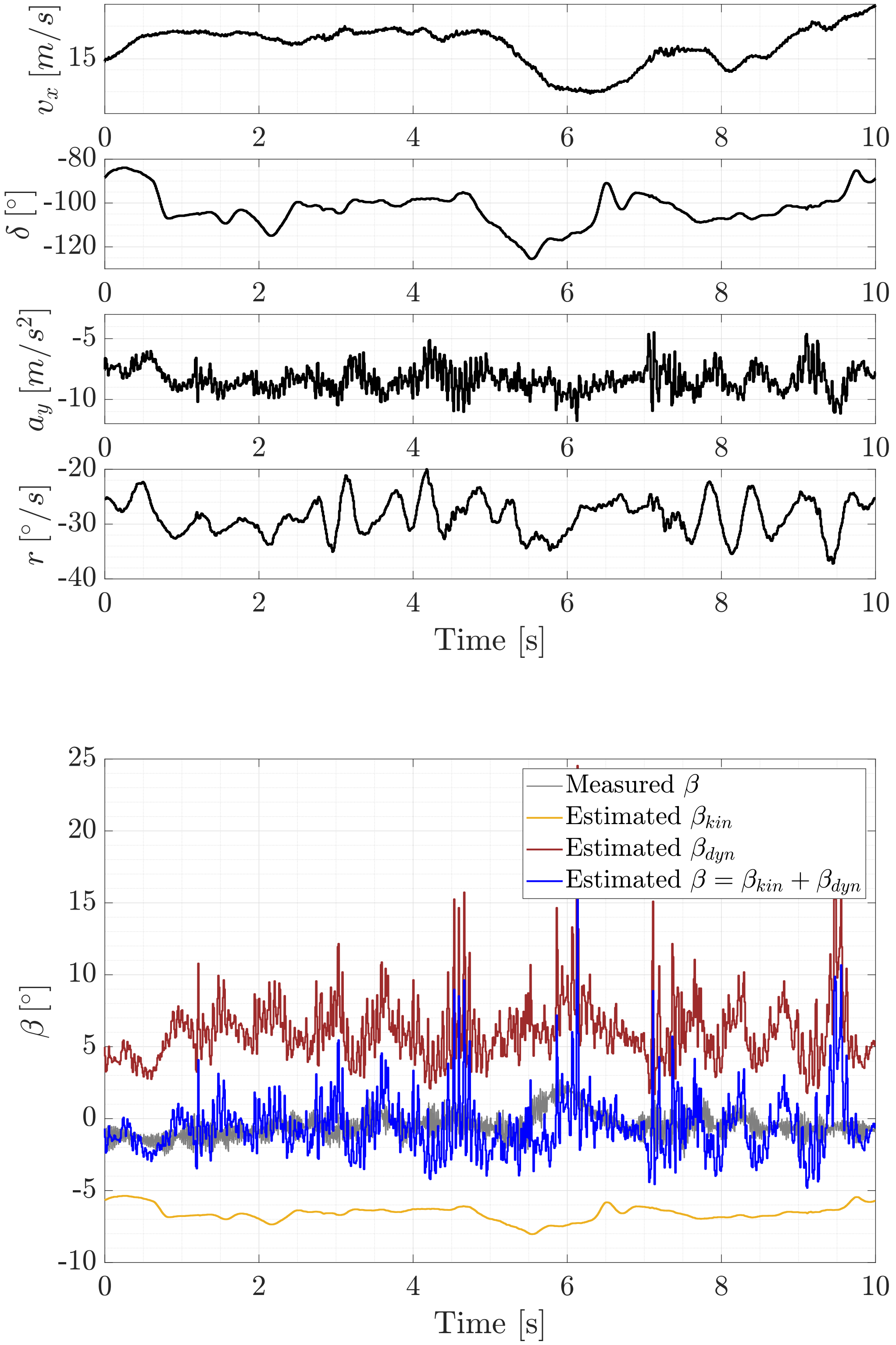


Figure 6.12: Circle manoeuvre, 16.7 m/s: (top) main quantities; (bottom) measured and estimated β.

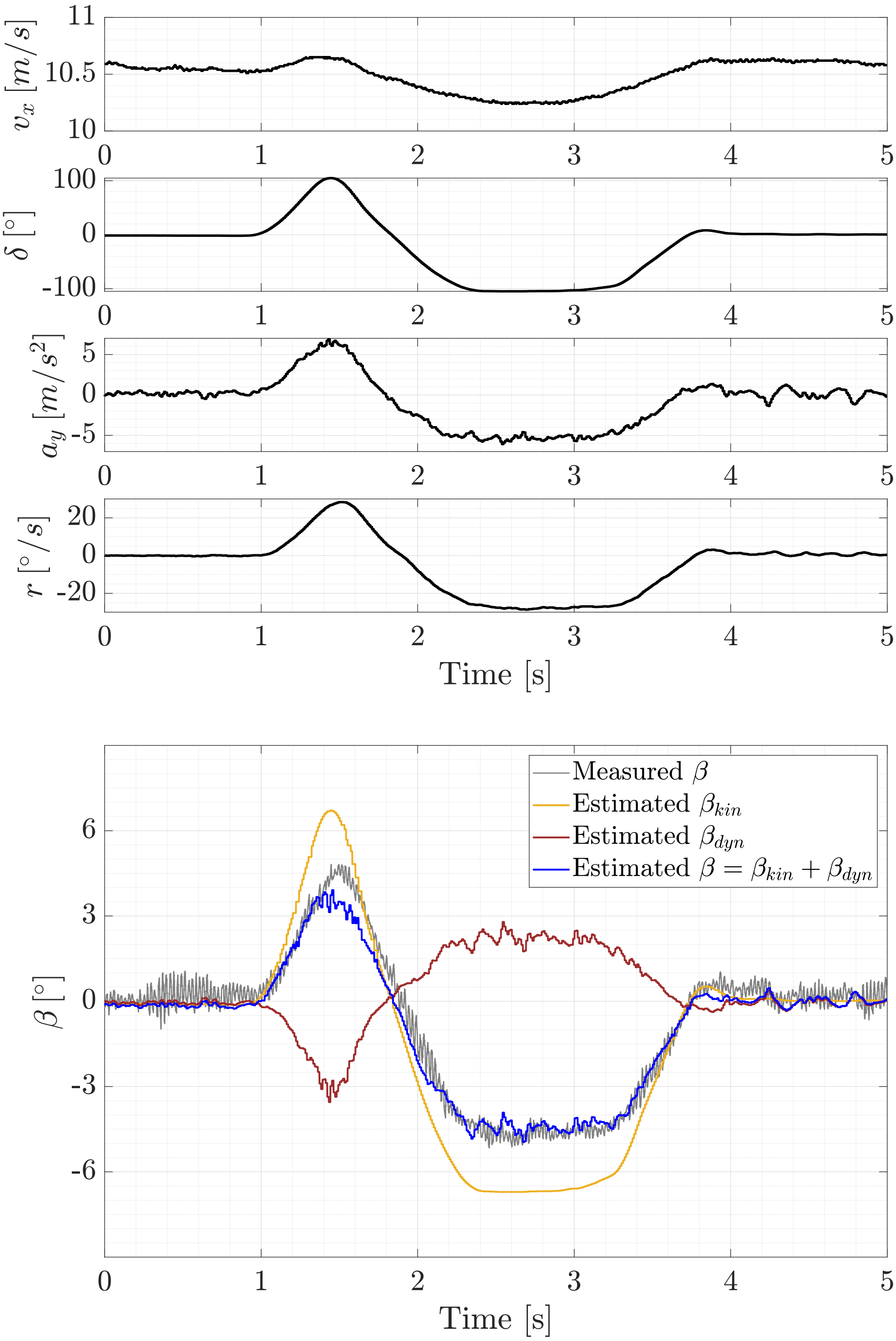


Figure 6.13: Sine with dwell manoeuvre, 11 m/s: (top) main quantities; (bottom) measured and estimated β.

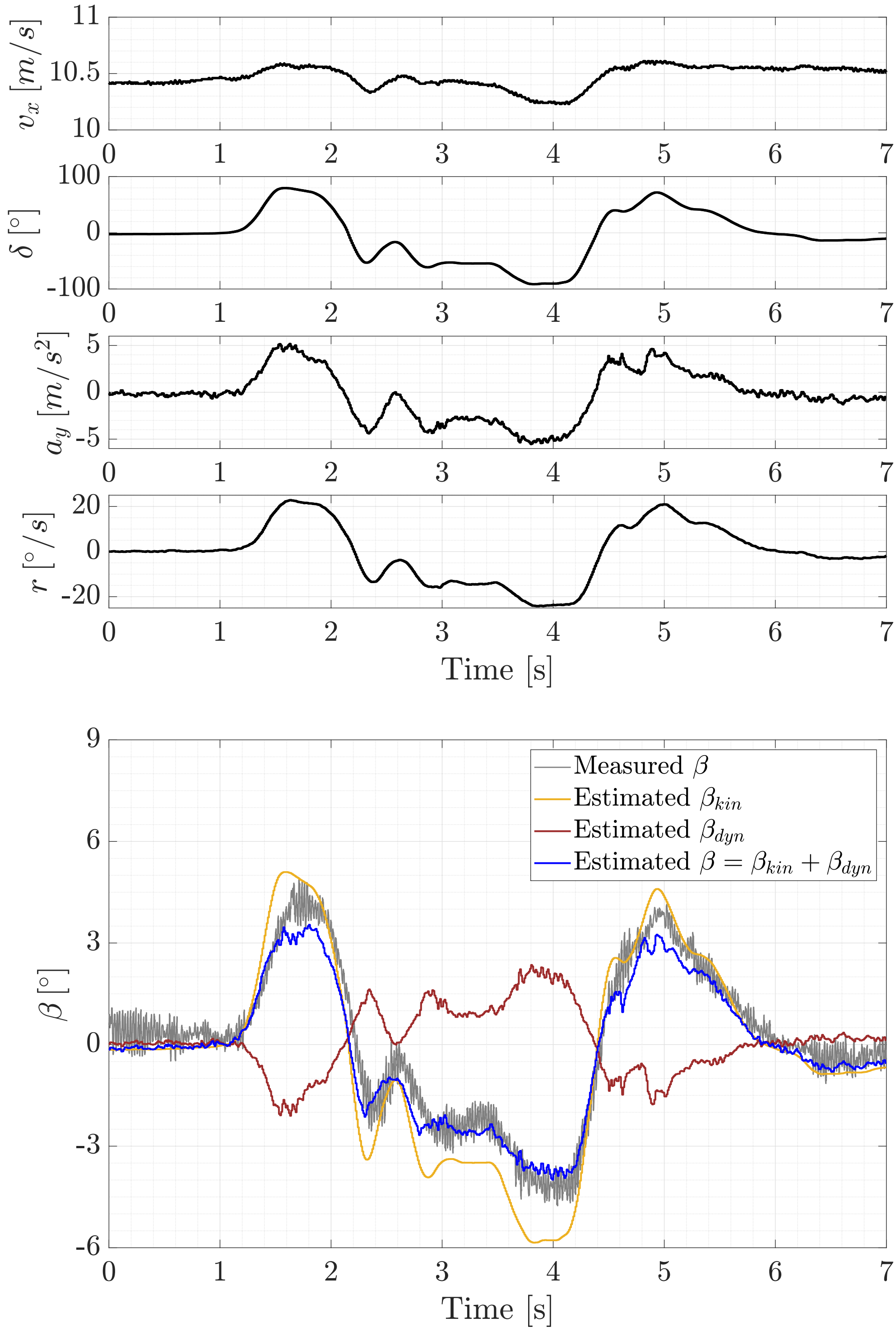


Figure 6.14: Double lane-change manoeuvre, 11 m/s: (top) main quantities; (bottom) measured and estimated β.

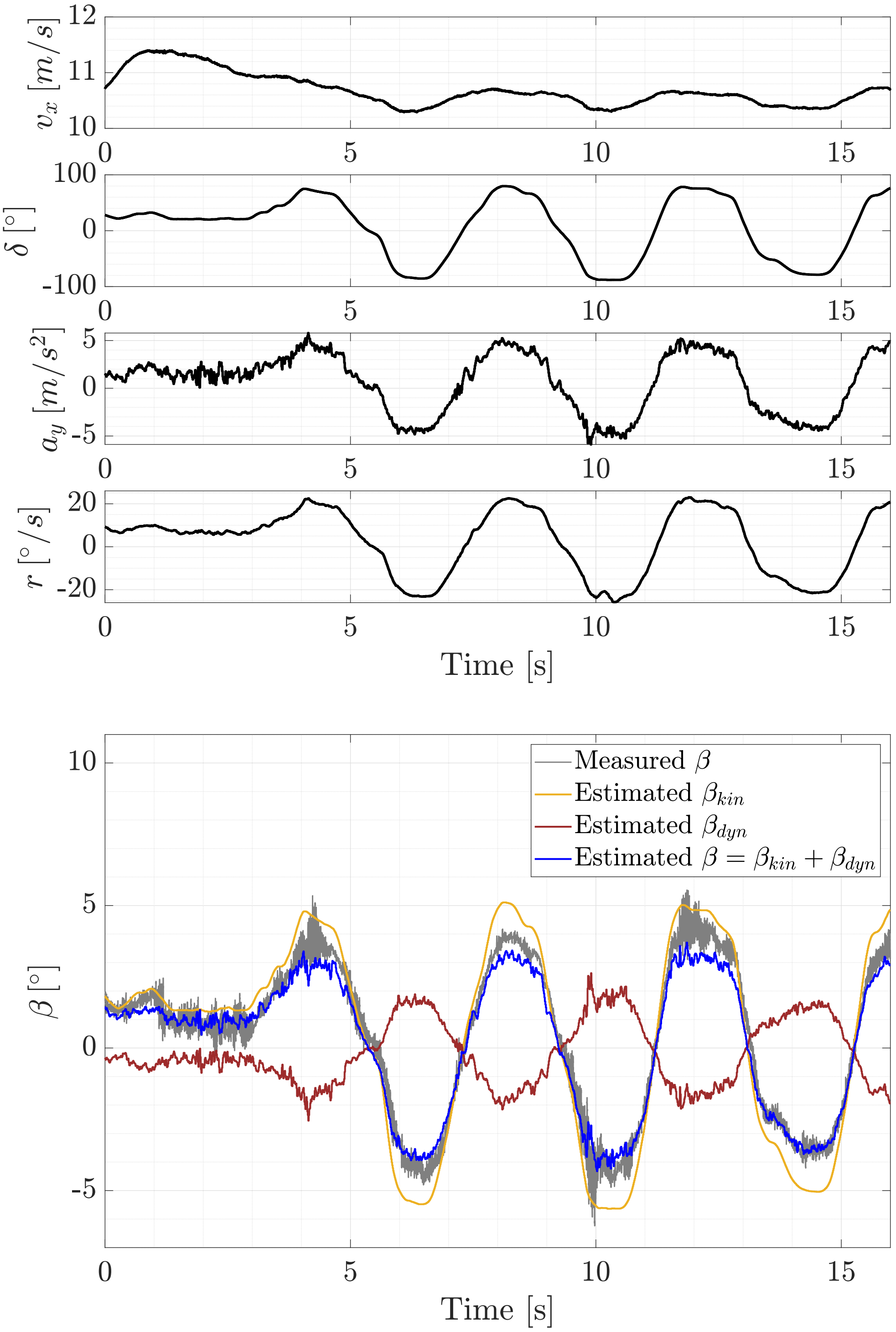


Figure 6.15: Slalom manoeuvre, 11 m/s: (top) main quantities; (bottom) measured and estimated β.

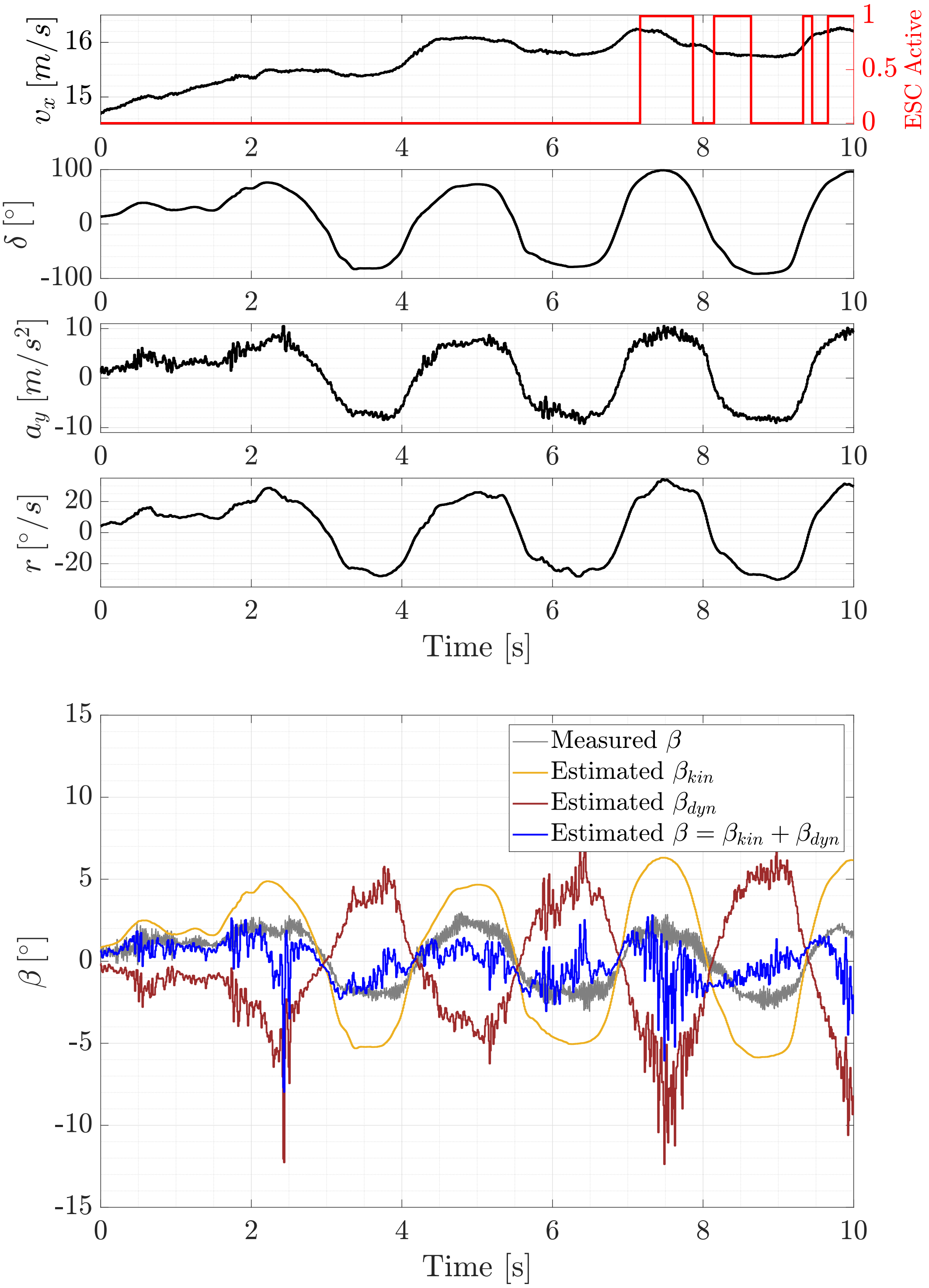


Figure 6.16: Slalom manoeuvre, 16.6 m/s: (top) main quantities; (bottom) measured and estimated β.

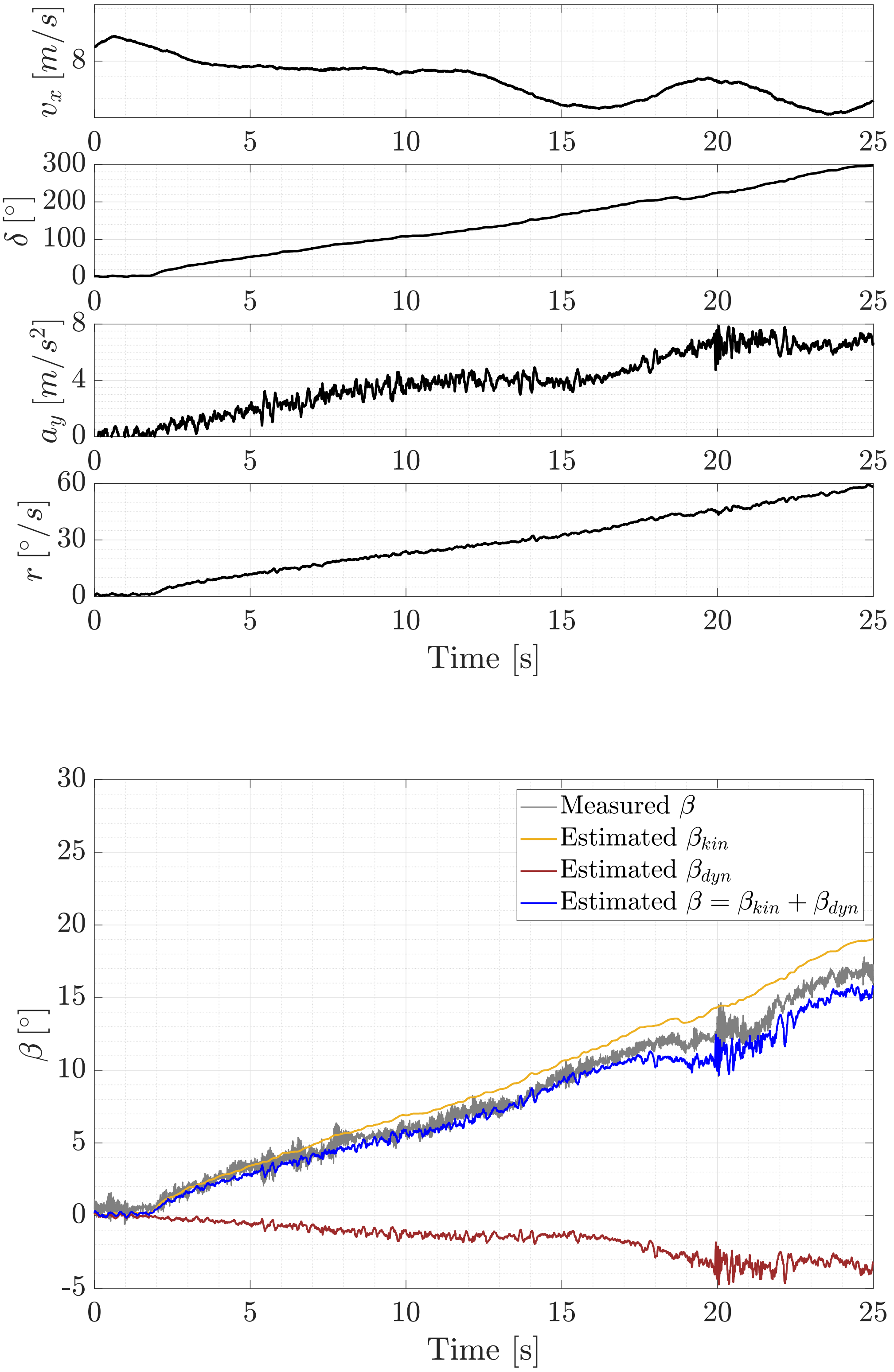


Figure 6.17: Ramp steer manoeuvre, 8.3 m/s: (top) main quantities; (bottom) measured and estimated β.

The proposed estimator yields a good-quality estimate, while also abiding by real-time constraints. Some interesting observations arise:

- During experimental tests, high yaw rate values have been observed to trigger the ESC, as shown in Fig. 6.16 (top). The noise on the lateral acceleration signal namely around 7s due to the activation of the ESC cause the estimate of the dynamic sideslip angle β_{dyn} to fluctuate in turn, as in Fig. 6.16 (bottom).
- Performance visibly deteriorates as the lateral acceleration increases over 10 m/s^2. This is due to the range of interpolation points being insufficient for an accurate portrayal of the relation existing between the lateral acceleration a_y and the dynamic sideslip angle β_{dyn} in such a wide dynamic range.

6.3.3 Comparative analysis

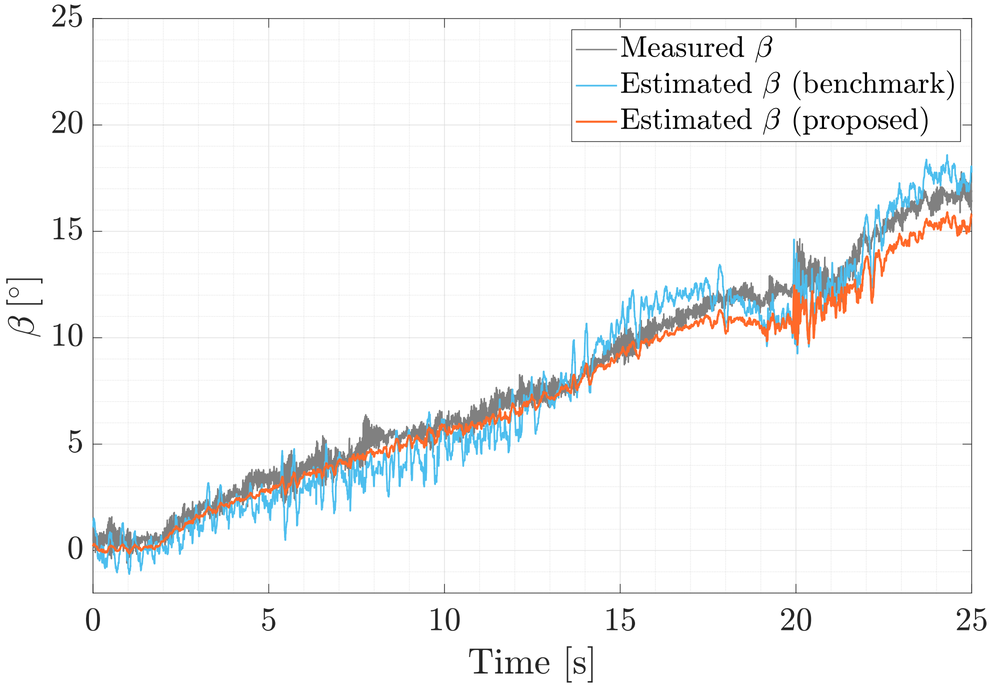
A performance comparison with the adaptive sideslip estimator presented in Van Aalst et al. [70] is conducted for further validation. This estimator is especially chosen as a reliable benchmark because it is capable of achieving stable estimation on the full vehicle range, by adapting the cornering stiffnesses to match the tire operating conditions. Outcomes are presented in terms of root-mean-square error (RMSE) in Table 6.5. The measured and estimated sideslip angle for both approaches are shown in Fig. 6.18, for a ramp steer manoeuvre (top) and a slalom manoeuvre (bottom). The difference between the measured and estimated sideslip angle is reported in Fig. 6.19 in terms of sideslip estimation error e_β, for both the proposed and the benchmark approach, on a ramp steer manoeuvre (top) and a slalom manoeuvre (bottom). These manoeuvres have been specifically selected to highlight that the adaptive and the proposed estimator perform comparatively well in manoeuvres where dynamics are building slowly (ramp steer in Fig. 6.18, top), while the proposed estimator appears to outperform the benchmark estimator when faster dynamics are observed (slalom in Fig. 6.18, bottom). Other than the performance, a bigger advantage is noted in terms of setup: the adaptive estimator is based on a Kalman filter, whose covariances need thorough tuning. Moreover, measurements of wheel speeds and vehicle yaw rate are required for the adaptive estimator, other than the

Table 6.5: Comparison of root-mean-square error (RMSE) between the proposed estimation approach and literature-based adaptive Kalman filter.

Manoeuvre	$\mathrm{RMSE}_{\mathrm{Kalman}}$ (°)	$\mathrm{RMSE}_{\mathrm{Proposed}}$ (°)
Ramp steer, 8.3 m/s	1.538	0.906
Sine with dwell, 11 m/s	1.096	0.488
Double lane-change, 11 m/s	1.920	0.570
Slalom, 11 m/s	1.248	0.562
Slalom, 16.7 m/s	1.587	1.449
Circle, 16.7 m/s	2.160	1.901

lateral acceleration signal: the proposed estimator only needs the lateral acceleration to estimate the dynamic sideslip angle component.

To summarise, this chapter presented the results of two test campaigns aiming to prove the real-time applicability of the proposed estimator. The estimator was deployed on a full-scale vehicle and proved to effectively retrieve the target signal. Performance was then compared to a benchmark adaptive Kalman-based estimator to assess its validity. This represents a further remarkable milestone within the proposed validation journey, as it sets the ground for the last envisioned step: the deployment of the proposed controller on the full-scale vehicle.



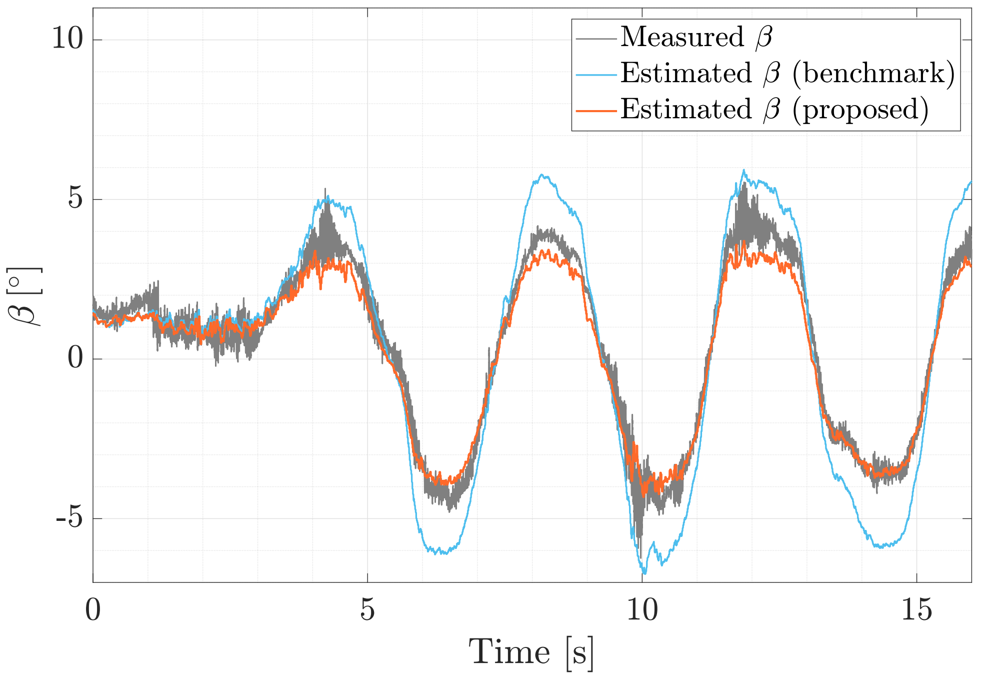
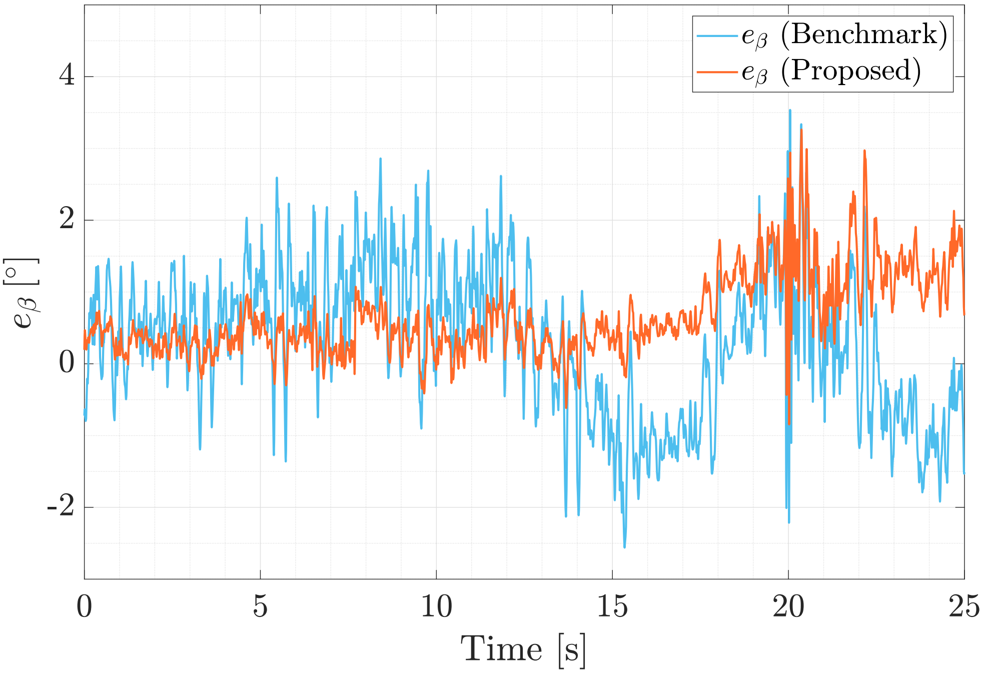


Figure 6.18: Estimate quality comparison with Kalman filter: (top) Ramp steer manoeuvre, 8.3 m/s; (bottom) Slalom manoeuvre, 11 m/s.



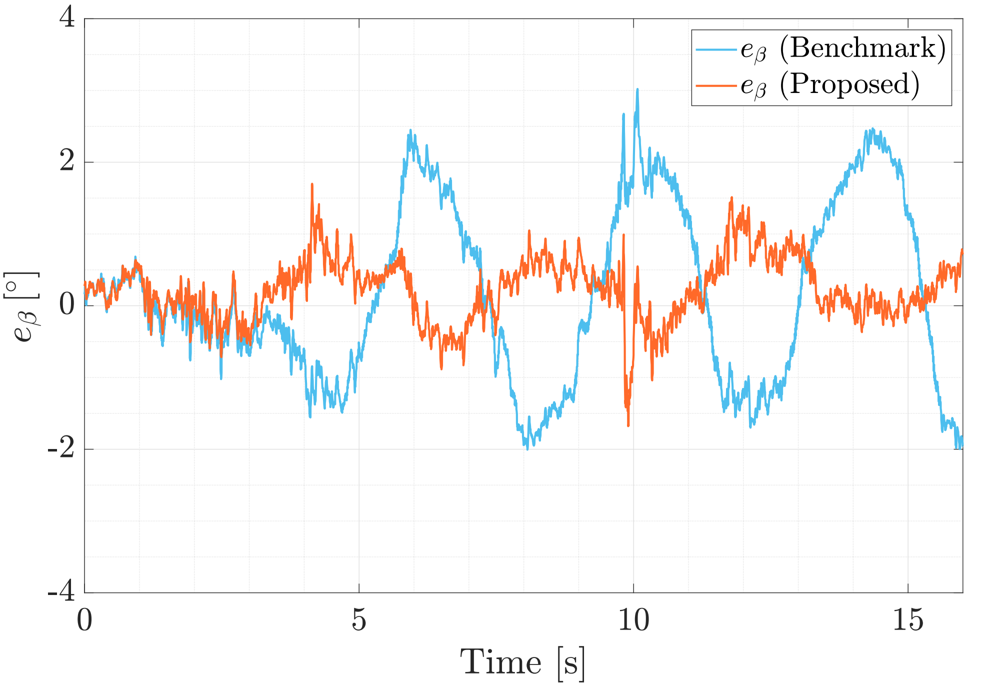


Figure 6.19: Error-based comparison with Kalman filter: (top) Ramp steer manoeuvre, 8.3 m/s; (bottom) Slalom manoeuvre, 11 m/s.

Chapter 7

Conclusions and Future Work

This body of work aims to improve the safety and reliability of driver assistance systems, proposing a straightforward and easy-to-replicate methodology to progressively test a torque vectoring-based lateral stability controller.

At each time step, the presented controller assesses the vehicle lateral stability and interprets the vehicle states to ultimately generate, if required, an appropriate control action. The latter is carried out by allocating braking torques to the individual wheels, hence constituting a suitable response to the danger of stability loss. The introduced validation journey for the controller is a variation of the “Vee” chart [10], where a linear workflow is envisioned to highlight the continuous sequence of events within the said framework.

The main idea is broken down into manageable steps for what would be otherwise an overwhelmingly complicated task. Starting from the design phase, where its performance requirements are stated, the controller goes through many simulation instances, running on platforms that are increasingly nearer both in complexity and computational capabilities to the one destined to run on a full-scale vehicle. In each simulation instance, extensive testing is performed to guarantee compliance of the controller response to the standards set in the design phase.

An additional set of contributions to existing knowledge were envisioned within the main mission. A collection of corresponding milestones relative to such objectives follows:

- **Novel reference behaviour design strategy** (Section 3.4.2). The target vehicle behaviour is established through the use of MAPs, overcoming the limits of the handling diagram and offering full capability to design the desired vehicle cornering response. A full design procedure is proposed, as well as a numerical comparison against two literature-renowned referencing methods.

- **Stability indicator estimation for control purposes** (Section 4.1, 4.2). Acknowledging the flaws within model-based approaches, two successful attempts of abstraction from the latter were made with the purpose of estimating the sideslip angle, one of the most crucial stability indicators. A neural network-based approach showed the potential of enriching data with physics knowledge, to estimate the target quantity without the need for a detrimentally high number of inputs.

  Moreover, an interpolation-based parametric procedure is designed to estimate the sideslip angle. The real-time capabilities of this estimator were successfully assessed in a real-time scenario. Using a full-scale test vehicle and appropriate hardware, promising results are obtained using vehicle-embedded sensors, which would eliminate the need for very expensive additional sensors during testing instances.

- **Analytic assessment of the vehicle stability region** (Section 3.4). An analytical method is presented to promptly determine the operating condition of the vehicle, for the lateral vehicle stability controller to possibly and timely intervene. The proposed methodology is fast and reliable, and comparisons to numerical and graphical approaches presented in the literature encourage application in real-time scenarios.

As a result of the listed novel contributions, new created knowledge may be evidenced. More specifically:

- MAPs offer a multifaceted picture of the handling behaviour of the vehicle. With respect to previous approaches, the MAPs-based reference contribution incorporates further complexity into the desired vehicle response, while retaining the simplicity of the reference being stored in a lookup table.

- Limitations in model-based approaches for estimating the sideslip angle are overcome. The proposed NN-based approach reduces the need for extensive input data, while the interpolation-based parametric approach is an original technique to reduce reliance on expensive additional sensors.
- The vehicle lateral stability is assessed for the first time using an analytical approach, enabling controller intervention in real time.

The testing journey was successfully carried up to the hardware-in-the-loop validation step showing promising performance. Furthermore, the interpolation-based sideslip angle estimator achieved very good results during real-time testing on a full-scale vehicle.

To establish a solid basis for future implementations, limitations of this work are addressed:

- Throughout this body of work, the friction coefficient is considered to be constant and equal to 1, and its possible variation is neglected. However, incorporating case studies with varying friction conditions is crucial for a comprehensive understanding of vehicle dynamics in different environments. This aspect should be thoroughly addressed to enhance the accuracy and applicability of the study.
- The proposed validation scenarios variety is limited to a mild and a challenging single lane change. To bring the controller towards full validation, a larger number of scenarios should be investigated. This includes standardised lateral stability-challenging manoeuvres, e.g. sine with dwell [71].
- The Dugoff model has been chosen for its simplicity and due to its curve shape being an invertible function, which allows to study the tyre slip angle and in turn the sideslip angle by means of simple algebraic manipulations. However, in the absence of an adhesion peak, this model does not offer a precise

representation of the entire operative range of the tyre. Looking to expand this work towards limit adhesion conditions like drifting, this gap should be addressed for a thorough study of the tyre behaviour.

In light of the discussed limitations and achievements, numerous opportunities for further investigation are envisioned. Building on the results discussed in the previous chapters, future work could target the following:

- **Expand the number and variety of controller test scenarios**. This thesis analysed two single lane-change scenarios to assess the controller performance within a milder or more severe stability condition. Further studies could focus on validating the controller using additional manoeuvres, namely sine with dwell, in accordance with the global technical regulation on ESC systems issued by the European Commission [72].

- **Implement a more elaborate control scheme**. The employed PI controller ensures low computational complexity while complying to real-time constraints. A study of the required computational effort revealed sufficient margin to introduce further complexity in the control algorithm, aiming to enhance its robustness. Linear Quadratic Regulator (LQR) control scheme or Model Predictive Control (MPC) are examples of potential control techniques to be implemented.

- **Investigate novel torque vectoring strategies**. At the present state, individual wheels receive the required torque from the high-level controller and equally split it between front and rear wheel of the left or right side. A more in-depth investigation on the most efficient way to split the required torque amount would lead to a more performing controller. For instance, studying the tyre behaviour under combined-load operating conditions could lead to an optimisation of the lateral force distribution [21].

- **Full-vehicle deployment**. Once the extensive testing phase is completed, the proposed control strategy can be deployed on a full-scale vehicle. At this stage, further tests can be performed to assess controller performance, considering the influence of hardware-related communication delays.

References

[1] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, “Three decades of driver assistance systems: Review and future perspectives,” *IEEE Intelligent transportation systems magazine*, vol. 6, no. 4, pp. 6–22, 2014.

[2] A. T. Van Zanten, R. Erhardt, and G. Pfaff, “Vdc, the vehicle dynamics control system of bosch,” *SAE transactions*, pp. 1419–1436, 1995.

[3] L. Zhai, T. Sun, and J. Wang, “Electronic stability control based on motor driving and braking torque distribution for a four in-wheel motor drive electric vehicle,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 4726–4739, 2016.

[4] A. Gimondi, M. Corno, and S. M. Savaresi, “A yaw rate based stability control for under-actuated vehicles,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, 2020.

[5] J. Guo, L. Chu, H. Liu, M. Shang, and Y. Fang, “Integrated control of active front steering and electronic stability program,” in *2010 2nd International Conference on Advanced Computer Control*, vol. 4, pp. 449–453, IEEE, 2010.

[6] B. Lenzo, A. Sorniotti, P. Gruber, and K. Sannen, “On the experimental analysis of single input single output control of yaw rate and sideslip angle,” *International Journal of Automotive Technology*, vol. 18, no. 5, pp. 799–811, 2017.

[7] M. Vignati, E. Sabbioni, and D. Tarsitano, “Torque vectoring control for iwm vehicles,” *International Journal of Vehicle Performance*, vol. 2, no. 3, pp. 302–324, 2016.

[8] “Road traffic injuries, world health organization, 2022..” https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, 2022. Accessed: 2023-09-20.

[9] I. I. for Highway Safety, “Fatality facts 2019.” http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm, 2019. Accessed: 2021-11-19.

[10] K. Forsberg and H. Mooz, “The relationship of system engineering to the project cycle,” in *INCOSE international symposium*, vol. 1, pp. 57–65, Wiley Online Library, 1991.

[11] N. B. Ruparelia, “Software development lifecycle models,” *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 3, pp. 8–13, 2010.

[12] M. Tristano, B. Lenzo, X. Xu, B. Forrier, T. D'hondt, E. Risaliti, and E. Wilhelm, "Real-time implementation of yaw rate and sideslip control through individual wheel torques," in *2022 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–6, IEEE, 2022.

[13] K. Etschberger, R. Hofmann, J. Stolberg, C. Schlegel, and S. Weiher, *Controller area network: basics, protocols, chips and applications.* IXXAT Automation, 2001.

[14] M. Tristano, B. Lenzo, X. Xu, B. Forrier, T. D'hondt, E. Risaliti, and E. Wilhelm, "Hardware-in-the-loop real-time implementation of a vehicle stability control through individual wheel torques," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 4, pp. 4683–4693, 2024.

[15] M. Tristano and B. Lenzo, "Design of the vehicle cornering response based on the map of achievable performance," *27th IAVSD Symposium on Dynamics of Vehicles on Roads and Tracks*, 2021.

[16] M. Tristano, B. Lenzo, X. Xu, H. Saxton, and X. Zhang, "Physics-infused neural network-driven investigation of vehicle sideslip angle," *28th IAVSD Symposium on Dynamics of Vehicles on Roads and Tracks*, 2023.

[17] M. Tristano and B. Lenzo, "A parametric interpolation-based approach to sideslip angle estimation (accepted)," in *AVEC International Symposium on Advanced Vehicle Control*, 2024.

[18] M. Tristano and B. Lenzo, "Estimating vehicle sideslip angle through kinematic and dynamic contributions: theory and experimental results (accepted for publication)," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 2024.

[19] M. Tristano and B. Lenzo, "Analytical solution of phase plane equilibria harnessing the root-rational tire model (under review)," *Vehicle system dynamics*, 2024.

[20] B. ISO, "8855: 2011. road vehicles—vehicle dynamics and road-holding ability—vocabulary," tech. rep., Tech rep., ISO, 2011.

[21] M. Guiggiani, "The science of vehicle dynamics," *Pisa, Italy: Springer Netherlands*, p. 15, 2014.

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

[23] B.-C. Chen and F.-C. Hsieh, "Sideslip angle estimation using extended kalman filter," *Vehicle System Dynamics*, vol. 46, no. S1, pp. 353–364, 2008.

[24] F. Cheli, E. Sabbioni, M. Pesce, and S. Melzi, "A methodology for vehicle sideslip angle identification: comparison with experimental data," *Vehicle System Dynamics*, vol. 45, no. 6, pp. 549–563, 2007.

[25] 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, no. 3, p. 355, 2018.

[26] H. B. Pacejka, *Tire and Vehicle Dynamics.* Oxford: Butterworth-Heinemann, 3rd ed., 2012. ISBN: 978-0-08-097016-5.

[27] B. Li, X. Yang, and J. Yang, "Tire model application and parameter identification-a literature review," *SAE International Journal of Passenger Cars-Mechanical Systems*, vol. 7, no. 2014-01-0872, pp. 231–243, 2014.

[28] H. B. Pacejka and E. Bakker, "The magic formula tyre model," *Vehicle system dynamics*, vol. 21, no. S1, pp. 1–18, 1992.

[29] H. Dugoff, P. S. Fancher, and L. Segel, "Tire performance characteristics affecting vehicle response to steering and braking control inputs," tech. rep., National Bureau of Standards, 1969.

[30] M. Bian, L. Chen, Y. Luo, and K. Li, "A dynamic model for tire/road friction estimation under combined longitudinal/lateral slip situation," tech. rep., SAE Technical Paper, 2014.

[31] H. B. Pacejka, "Simplified analysis of steady-state turning behaviour of motor vehicles. part 1. handling diagrams of simple systems.," *Vehicle System Dynamics*, vol. 2, no. 3, pp. 161–172, 1973.

[32] L. De Novellis, A. Sorniotti, and P. Gruber, "Driving modes for designing the cornering response of fully electric vehicles with multiple motors," *Mechanical Systems and Signal Processing*, vol. 64, pp. 1–15, 2015.

[33] Z. Hao, L. Xian-sheng, S. Shu-ming, L. Hong-fei, G. Rachel, and L. Li, "Phase plane analysis for vehicle handling and stability," *International journal of computational intelligence Systems*, vol. 4, no. 6, pp. 1179–1186, 2011.

[34] C. G. Bobier-Tiu, C. E. Beal, J. C. Kegelman, R. Y. Hindiyeh, and J. C. Gerdes, "Vehicle control synthesis using phase portraits of planar dynamics," *Vehicle System Dynamics*, 2018.

[35] M. A. Selby, *Intelligent vehicle motion control.* PhD thesis, University of Leeds, 2003.

[36] S. Inagaki, I. Kushiro, and M. Yamamoto, "Analysis on vehicle stability in critical cornering using phase-plane method," *Jsae Review*, vol. 2, no. 16, p. 216, 1995.

[37] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems [j]," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[38] S. Haykin, *Kalman filtering and neural networks.* John Wiley & Sons, 2004.

[39] F. Naets, S. van Aalst, B. Boulkroune, N. El Ghouti, and W. Desmet, "Design and experimental validation of a stable two-stage estimator for automotive sideslip angle and tire parameters," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 9727–9742, 2017.

[40] F. Di Biase, B. Lenzo, and F. Timpone, "Vehicle sideslip angle estimation for a heavy-duty vehicle via extended kalman filter using a rational tyre model," *IEEE Access*, vol. 8, pp. 142120–142130, 2020.

[41] L. De Novellis, A. Sorniotti, P. Gruber, L. Shead, V. Ivanov, and K. Hoepping, "Torque vectoring for electric vehicles with individually controlled motors: State-of-the-art and future developments," *World Electric Vehicle Journal*, vol. 5, no. 2, pp. 617–628, 2012.

[42] A. Mangia, B. Lenzo, and E. Sabbioni, "An integrated torque-vectoring control framework for electric vehicles featuring multiple handling and energy-efficiency modes selectable by the driver," *Meccanica*, vol. 56, no. 5, pp. 991–1010, 2021.

[43] M. Vignati, E. Sabbioni, and F. Cheli, "A torque vectoring control for enhancing vehicle performance in drifting," *Electronics*, vol. 7, no. 12, p. 394, 2018.

[44] I. O. for Standardization, "Iso 3888-1: 2018 passenger cars—test track for a severe lane-change manoeuvre—part 1: Double lane-change," 2018.

[45] "Simrod experience: Model-based system testing in practice." https://blogs.sw.siemens.com/simcenter/simrod-experience-model-based-system-testing/, 2021. Accessed: 2023-09-20.

[46] J. C. Butcher, *Numerical methods for ordinary differential equations.* John Wiley & Sons, 2016.

[47] W. H. Press, *Numerical recipes 3rd edition: The art of scientific computing.* Cambridge university press, 2007.

[48] K. Ogata, *Modern Control Engineering.* Upper Saddle River, NJ: Prentice Hall, 5th edition ed., 2010.

[49] B. Lenzo, M. Zanchetta, A. Sorniotti, P. Gruber, and W. De Nijs, "Yaw rate and sideslip angle control through single input single output direct yaw moment control," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 1, pp. 124–139, 2020.

[50] S. Sastry, *Nonlinear systems: analysis, stability, and control*, vol. 10. Springer Science & Business Media, 2013.

[51] I. Zubov, I. Afanasyev, A. Gabdullin, R. Mustafin, and I. Shimchik, "Autonomous drifting control in 3d car racing simulator," in *2018 International Conference on Intelligent Systems (IS)*, pp. 235–241, IEEE, 2018.

[52] P. Ruffini, *Teoria generale delle equazioni: in cui si dimostra impossibile la soluzione algebraica delle equazioni generali di grado superiore al quarto.* Nella stamperia di S. Tommaso d'Aquino, 1799.

[53] H. B. Pacejka, "Simplified analysis of steady-state turning behaviour of motor vehicles part 2: Stability of the steady-state turn," *Vehicle System Dynamics*, vol. 2, no. 4, pp. 173–183, 1973.

[54] H. B. Pacejka, "Simplified analysis of steady-state turning behaviour of motor vehicles part 3: More elaborate systems," *Vehicle System Dynamics*, vol. 2, no. 4, pp. 185–204, 1973.

[55] E. Esmailzadeh, A. Goodarzi, and G. Vossoughi, "Optimal yaw moment control law for improved vehicle handling," *Mechatronics*, vol. 13, no. 7, pp. 659–675, 2003.

[56] N. Guo, X. Zhang, Y. Zou, B. Lenzo, G. Du, and T. Zhanga, “A supervisory control strategy of distributed drive electric vehicles for coordinating handling, lateral stability, and energy efficiency,” *IEEE Transactions on Transportation Electrification*, 2021.

[57] W. Wei, B. Shaoyi, Z. Lanchun, Z. Kai, W. Yongzhi, H. Weixing, *et al.*, “Vehicle sideslip angle estimation based on general regression neural network,” *Mathematical Problems in Engineering*, vol. 2016, 2016.

[58] A. Bonfitto, S. Feraco, A. Tonoli, and N. Amati, “Combined regression and classification artificial neural networks for sideslip angle estimation and road condition identification,” *Vehicle system dynamics*, vol. 58, no. 11, pp. 1766–1787, 2020.

[59] T. Gräber, S. Lupberger, M. Unterreiner, and D. Schramm, “A hybrid approach to side-slip angle estimation with recurrent neural networks and kinematic vehicle models,” *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 1, pp. 39–47, 2018.

[60] S. Haykin, *Neural networks and learning machines, 3/E.* Pearson Education India, 2009.

[61] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[62] B. Lenzo, “Torque vectoring control for enhancing vehicle safety and energy efficiency,” in *Vehicle Dynamics: Fundamentals and Ultimate Trends*, pp. 193–233, Springer, 2021.

[63] J. C. Kegelman, L. K. Harbott, and J. C. Gerdes, “2013 targa sixty-six. stanford digital repository..” http://purl.stanford.edu/yf219gg2055, 2016.

[64] J. C. Kegelman, L. K. Harbott, and J. C. Gerdes, “2014 targa sixty-six. stanford digital repository..” http://purl.stanford.edu/hd122pw0365, 2016.

[65] F. Farroni, “Trick-tire/road interaction characterization & knowledge-a tool for the evaluation of tire and vehicle performances in outdoor test sessions,” *Mechanical Systems and Signal Processing*, vol. 72, pp. 808–831, 2016.

[66] S. I. Mall, “Simatic ipc647e..” https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10346533?activeTab=productinformation&regionUrl=WW, 2023. Accessed: 2023-11-02.

[67] T. Dhondt, Y. Mollet, A. J. Joos, L. Cecconi, M. Sarrazin, and J. Gyselinck, “Scalable electric-motor-in-the-loop testing for vehicle powertrains.,” in *ICINCO*, pp. 594–603, 2020.

[68] “Ttc 580 | hydac.” https://www.hydac.com/shop/en/ttc-580-1000427891, 2023. Accessed: 2023-11-02.

[69] dSPACE, “Microautobox ii - compact and robust prototyping system for in-vehicle applications.” https://www.dspace.com/en/inc/home/products/hw/micautob/microautobox2.cfm, 2022. Accessed: 2023-11-06.

[70] S. van Aalst, F. Naets, B. Boulkroune, W. De Nijs, and W. Desmet, “An adaptive vehicle sideslip estimator for reliable estimation in low and high excitation driving,” *IFAC-PapersOnLine*, vol. 51, no. 9, pp. 243–248, 2018.

[71] I. O. for Standardization, “Iso 19365: Passenger cars — validation of vehicle dynamic simulation — sine with dwell stability control testing,” 2016.

[72] E. Commission, “Mobility & transport - road safety.” https://road-safety.transport.ec.europa.eu/european-road-safety-observatory/statistics-and-analysis-archive/esafety/electronic-stability-control_en, -. Accessed: 2023-10-06.