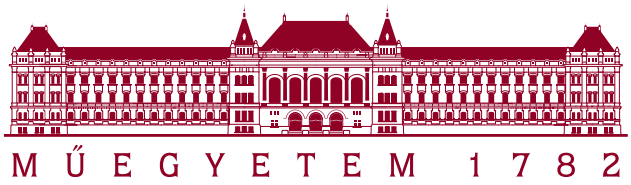


Budapest University of Technology and Economics
Faculty of Transportation Engineering and Vehicle Engineering
Department of Control for Transportation and Vehicle Systems
Budapest, Hungary

Eötvös Loránd Research Network
Institute for Computer Science and Control
Systems and Control Lab
Budapest, Hungary

Application of data-driven methods for improving the peformances of lateral vehicle control systems

Adatvezérelt módszerek alkalmazása laterális járműirányítási rendszerek minőségi kritériumainak javítására

Thesis by

DÁNIEL FÉNYES

In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Supervisor:

Dr. Balázs Németh, Ph.D.

2021.

Declaration

Undersigned, Dániel Fényes, hereby state that this Ph.D. Thesis is my own work wherein I have used only the sources listed in the Bibliography. All parts taken from other works, either in a word for word citation or rewritten keeping the original contents, have been unambiguously marked by a reference to the source.

Nyilatkozat

Alulírott Fényes Dániel kijelentem, hogy ezt a doktori értekezést magam készítettem és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Budapest, 2021. 09. 21.

Fényes Dániel

The reviews of this Ph.D. Thesis and the record of defense will be available later in the Dean Office of the Faculty of Transportation Engineering and Vehicle Engineering of the Budapest University of Technology and Economics.

Az értekezésről készült bírálatok és a jegyzőkönyv a későbbiekben a Budapesti Műszaki és Gazdaságtudományi Egyetem Közlekedésmérnöki és Járműmérnöki Karának Dékáni Hivatalában elérhetőek.

First of all, I would like to express my gratitude to my supervisor Dr. Balázs Németh for his help and patience with which he greatly contributed to my professional development and this dissertation.

In addition, I would like to thank Prof. Péter Gáspár and Prof. József Bokor for the trust placed in me, without which this work would not have been possible.

Furthermore, I would like to thank the closest colleagues at Systems and Control Lab, SZTAKI: Prof. Zoltán Szabó, Dr. Gábor Rödönyi, Tamás Hegedűs, András Mihály, Attila Lelkó and Máté Fazekás for all their help I received from them.

Last but not least, I would like to thank my family for their support.

The researches, presented in this thesis, have been carried out under the project "Talent management in autonomous vehicle control technologies" (EFOP-3.6.3-VEKOP-16-2017-00001) financed by the Hungarian Government and the European Social Fund, and in cooperation with Autonomous Systems National Laboratory sponsored by the Ministry of Innovation and Technology NRDI Office.

ABSTRACT

This thesis proposes data-driven methods for improving the performances of lateral control systems. The resulted methods contribute to the estimation of road-vehicle parameters, to the analysis of nonlinear systems, to the modeling of vehicle systems and to the improvement of vehicle-oriented control design methods. Novel machine-learning-based algorithms are presented for estimating tyre pressure and adhesion coefficient of the tyre-road contact. A data-driven lateral stability analysis is proposed, which can be used to approximate the stability regions of the vehicle during its operation using the measurements of onboard sensors. Then, a possible application of the result of the stability analysis is presented, in which the lateral control system is extended with a longitudinal velocity optimization algorithm. This thesis also deals with the modeling problem of autonomous vehicle systems using data-driven approaches. The proposed modeling methods use different machine-learning-based strategies to overcome the issues posed by the nonlinearities of the vehicle dynamics. The optimized model is ordered into a Linear Parameter-Varying (LPV) form. This formalization allows using the classical control design paradigms, which can guarantee the stability of the closed-loop system during the operation of the vehicle. A lateral control design is also proposed based on the optimized LPV model. The efficiency of this controller is illustrated through a comprehensive simulation example, which is performed in the vehicle dynamics simulation software, CarSim. In this thesis, the modeling and control problem of variable-geometry suspension-based steering system is also addressed. This steering system can also be used to improve the stability of the vehicle. For validating the effectiveness of this steering concept, a testbed has been built in SZTAKI Institute for Computer Science and Control. For this testbed, a data-driven modeling and control strategy is also proposed, which is implemented and tested through a Hardware-In-the-Loop (HiL) simulation. In the simulation, the lateral vehicle dynamics are modeled by the CarMaker simulation software while the dynamics of the variable-geometry suspension system are provided by the testbed.

These contributions have been published in international journals and conference papers.

Keywords: data-driven vehicle control, data-driven estimation, LPV control design, variable-geometry suspension,

ÖSSZEFOGLALÓ

A disszertáció adat-alapú módszerek alkalmazásával foglalkozik oldalirányú járműirányítási rendszerek minőségi jellemzőinek (performancia) javítása céljából. A dolgozat első részében gépi tanulási módszereken alapuló új becslési algoritmusok kerülnek bemutatásra, amelyek az abroncsnyomás és a kerek-talaj kapcsolat közötti tapadási együttható meghatározására alkalmazhatóak. Továbbá, bemutatásra kerül egy adat-alapú stabilitás analízis, amely felhasználható a jármű stabilitási tartományainak meghatározására, kizárólag a jármű fedelezéti szenzorainak a méréseire támaszkodva. Az említett becslési és elemezési módszerek felhasználásra kerülnek oldalirányú járműdinamikai szabályzó vonatkozásában, azaz az analyzis eredményeképpen kapott stabilitási tartományok egy hosszirányú sebesség optimalizálási algoritmusban kerülnek figyelembe vételre. A disszertáció továbbá az autonóm járművek rendszereinek modellezési kérdéseivel is foglalkozik. A javasolt modellezési módszerek különböző gépi tanuláson alapuló stratégiákat alkalmaznak a járműdinamika nemlinearitásaiból adódó problémák leküzdésére. Az eredményül kapott optimalizált modell Lineáris Változó-Paraméterű (LPV) struktúrába kerül felírásra, ily módon a stabilitás és performancia garanciákat biztosító klasszikus irányítástervezési megközelítések alkalmazhatóak. A kapott modell felhasználásra kerül a jármű oldalirányú irányítási algoritmusának kidolgozása során. A szabályozási algoritmus hatékonysága átfogó szimulációs példákon keresztül kerül bemutatásra nagypontosságú járműdinamikai szimulációs szoftverek (CarSim/CarMaker) felhasználásával. A disszertáció továbbá a változtatható-geometriájú futóművön alapuló kormányzási módszer modellezésével és irányítástervezésével is foglalkozik, amely segítségével szintúgy javíthatók a jármű oldalirányú minőségi jellemzői. A kormányzási koncepció validásához építésre került egy tesztpad is a Számítástechnikai és Automatizálási Kutatóintézetben (SZTAKI). A tesztpadhoz kidolgozásra került egy adat-alapú modellezési és irányítási struktúra, amely egy Hardware-In-the-Loop (HiL) szimuláción implementálásra és validálásra.

Az értekezés új tudományos eredményei nemzetközi folyóirat- és konferenciacikkek formájában kerültek a szakmai nyilvánosság elé.

Kulcsszavak: adat-alapú gépjármű irányítás, adatvezérelt becslés, LPV szabályozó tervezés, változtatható geometriájú futómű,

CONTENTS

1. INTRODUCTION AND MOTIVATION

In recent years, the automation of vehicles has become one of the main challenges of the automotive industry. This challenge involves the development of several algorithms and special devices, because the fully automated vehicle must meet numerous strict criteria, such as: recognition of risky/dangerous traffic scenarios and avoidance, efficient path planning, stable and safe motion of the vehicle etc.

Although the first fully automated vehicle is not released, some of its features have already been introduced. These features belong to the group of Advanced Driver Assistance Systems (ADAS), [1]. ADAS includes several technical solutions, which aim to take over the burden of driving in certain traffic scenarios. One of the earliest solutions is the Anti-lock Braking System (ABS), which can adjust the braking pressure to maintain the stability of the vehicle during an emergency braking situation [2]. Another widespread example is the Adaptive Cruise Control (ACC), which can take over the longitudinal control of the vehicle by maintaining a predefined longitudinal velocity or adjusting its speed to the vehicle moving ahead of the controlled vehicle, see e.g. [3]. Electronic Stability Program (ESP) is also an outstanding part of ADAS. ESP aims to improve or maintain the performance of the vehicle when it is not able to perform the cornering maneuver. This is achieved by applying individually controlled braking pressures on both sides of the vehicle [4]. The difference between the pressures generates an additional moment on the vertical axis of the vehicle by which stability of the vehicle can be maintained in understeer and oversteer situations. The Lane Keeping Assist (LKA) is an advanced assistant feature, which can take over the steering of the vehicle in certain conventional driving scenarios, such as cruising on a highway [5]. However, most of the LKA-like algorithms require the attention of the driver, because at any time of driving the algorithm may give back the control of the vehicle to the driver due to unexpected events or dangerous situations. There are other algorithms and technologies, which belong to the ADAS group, a categorization and overview of them can be found in [6].

All of the presented ADAS technologies require advanced control solutions, which can guarantee stable and safe motion of the vehicle during intervention. The mostly used control solutions have been based on the classical model-based control paradigms in the last decades. The classical methods are, in general, the model-based approaches, in which the physical and/or mathematical description of the considered system plays a crucial role.

Brief overview of model-based methods in vehicle control design

Over the last century, several model-based control schemes have been developed, which have been used in vehicle-related applications. One of the oldest and widely used approaches is the Proportional-Derivative-Integral (PID) method, which can be used for controlling the lateral dynamics of the vehicle, see [7]. The main advantage of this method is the simplicity regarding both its design and its implementation as well. However, it has numerous drawbacks. For example, the structure of the controller is linear, with which the motion of the vehicle in its entire operational range cannot be improved. Moreover, the model of the process only during the controller parameter tuning can be considered and thus, it is not a real model-based controller. Therefore, the controller, designed based on this approach, may become less effective in certain dynamical situations. A solution could be the gain-scheduled PID control, in which multiple operating points of the system can be taken into account, see e.g. [8]. In this way, the performances of the overall system can be improved. However, there is an important pitfall of the gain-scheduling control methods, i.e. the stability of the closed-loop system cannot be guaranteed between two operating points.

For this problem, robust controller methods provide a solution, such as $\mathcal{H}_\infty$ and Linear Parameter-Varying-based (LPV) control approaches. Within the LPV framework, the nonlinear dynamics of a system can be approximated by a set of linear systems, in which the connection among the linear systems is given by certain, selected parameters called scheduling variables [9]. Using the LPV framework, the stability of the closed-loop system can be guaranteed even between the operating points. In vehicle control, the LPV-based control is widely used. For example, [10] presents a combined steering-based and braking-based lateral control design using LPV approach. These methods can also be used for improving the roll dynamics of the vehicle, see [11]. Even the control problem of the coupled longitudinal/lateral dynamics can be solved by this method, see [12]. Beside the many advantages of LPV framework, it poses some problems, which must be addressed during the modeling phase of the control design. One of these problems is related to the selection of the scheduling variables. In some cases, the required scheduling parameters of the system are not measurable and neither observable. It results in a less accurate model of the system, which can influence the performances of the controller. Another problem could be that case when the mathematical description of the system is inaccurate due to the uncertainties and unmodeled dynamics.

The Model Predictive Control (MPC) approach has gained significant attention in the automotive industry in the last decade. The basic concept of the MPC approach is to predict the behavior of the system on a predefined prediction horizon. Using this prediction, the optimal control sequence can be computed by minimizing its cost function. The main advantage of this method is its user-friendly design process and its implementation. In the literature, several vehicle-oriented applications can be found. For example, a longitudinal control design for autonomous

vehicles is proposed by [13]. The MPC approach can also be used for trajectory tracking problems, see [14]. The trajectory planning of autonomous vehicles can also be made by using predictive approaches, see [15, 16]. In the original concept of MPC, a general (nonlinear) dynamical model is used to compute the optimal input sequence. These optimization problem using dynamic programming approaches can be solved, see [17, 18]. Due to high computational cost and increased time, these optimization methods in practice cannot be applied. High computational demand can be reduced by using discrete time-grids and Lagrange's reduced gradient algorithms [19]. In practice, the most widely used MPC approach is based on linear models, by which the general optimization problem can be transformed into a quadratic form. Due to the linear model, it also bears some limitations, which can result in reduced performance level. In recent decades, the development of microcontrollers and processors allowed the engineers to consider control approaches with higher computational costs, e.g., Nonlinear Model Predictive Control (NMPC). For example, in [20, 21] NMPC-based lateral stability control strategies are proposed. The main drawback of NMPC is the complexity, which results in high computational time during the optimization process. Thus, in most of the implemented algorithms, only a sub-optimal solution can be reached, which can degrade the performances of the closed-loop system.

All presented control strategies can be used for certain purposes, especially in ADAS applications. However, the mentioned drawbacks of the approaches restrict their application in fully automated, autonomous vehicles since they must operate in any traffic scenarios, when there is no option to take back the control by the driver. A possible way to overcome these problems is to combine model-based control design methods with data-driven approaches. Since the modern vehicles are getting equipped with more and more sensors, large amount of data can be collected, which can be used to design enhanced control systems for vehicles. The main advantage of this method is based on the hypothesis that the collected large datasets may contain hidden information, by which the dynamical behavior of the vehicle can be more accurately described. It is still an open problem how to reveal this information from the measurements. This thesis focuses on this problem and gives some possible ways to reveal this information and to use them in the control design.

Although this topic is relatively new, several papers have been published on data-driven control solutions, which are mainly based on a machine learning algorithms.

Brief overview of vehicle-oriented data-driven control approaches

The machine-learning-based solutions include various methods, such as deep learning through the training of neural networks [22], Support Vector Machine (SVM) or decision logic algorithms [23, 24, 25]. The main advantage of these algorithms is that they can learn from data, thus their models inherently contain the nonlinear behavior of the control plant. Thus, in several control applications better performances comparing to the classical approaches might be achieved. However,

some of these methods, especially the neural networks, have the drawback that it is difficult to find systematic methods to prove the stability and performances of the closed-loop system. Since autonomous vehicles are safety-critic systems, it is recommended to reformulate the learning problem.

There are other solutions, which deal with the identification problem of the dynamical systems. For example, in [26, 27] *SVM-LS*-based *LPV* system identification methods are presented. Another data-driven identification solution can be found in [28], which can ease the tuning of the hyperparameters of the model identification process. A non-parametric and probabilistic approach is proposed by [29] for identifying nonlinear system considering uncertain noises on the measured signals. Particle Bernstein polynomials-based regression method is presented in [30], which is suitable for multivariate regression problems.

In the literature, some solutions can be found, which can provide methods for the reformulation of the learning features in the control problem. For example, [31] proposes a *MPC*-based control solution, in which the terminal cost and set are determined through an iterative process. In this idea the model of the system is based on physical principles with its limitations. Other solution is the Model Free Control (*MFC*), which is proposed in [32, 33]. The *MFC* method does not require a model of the controlled system, but it uses a local model, in which the model in each time step is updated.

The presented algorithms lack the guarantees of the stability, which makes them risky to use in safety critical systems such as autonomous vehicles. However, there are some solutions, which address this issue. For example, [34] presents a safety-set-based control strategy, which can modify the input signal of the system when the output of the machine-learning-based agent may destabilize the system. Another solution is given by [35], in which a Hamilton-Jacobi reachability algorithm is exploited, which can work together with any machine learning-based solutions. [36] presents a combined approach, in which a classical controller is applied to control the linearized system while the machine-learning-based algorithm handles the nonlinearities of the system. The stability proof of this concept is based on Linear Matrix Inequalities (LMIs).

A special field of the machine learning algorithms is the reinforcement learning (RF) [37]. In contrast to other machine learning algorithms, RF does not require a large amount of data for its training process. RF algorithm trains its model by continuously evaluating a predefined cost function, called reward. Several papers deal with the control problem of autonomous vehicles using RF methods, see [38, 39]. Although the RF-based solutions can provide outstanding performances, the stability of the closed-loop system is still an open problem. In order to bridge this gap, [40] proposes an algorithm, which combines the robust control method with RF. In this solution, the RF-based neural network works in parallel with a robust controller. During the training process, the stability of the algorithm is achieved by using uncertainty models.

Although data-driven machine learning-based solutions can provide outstanding

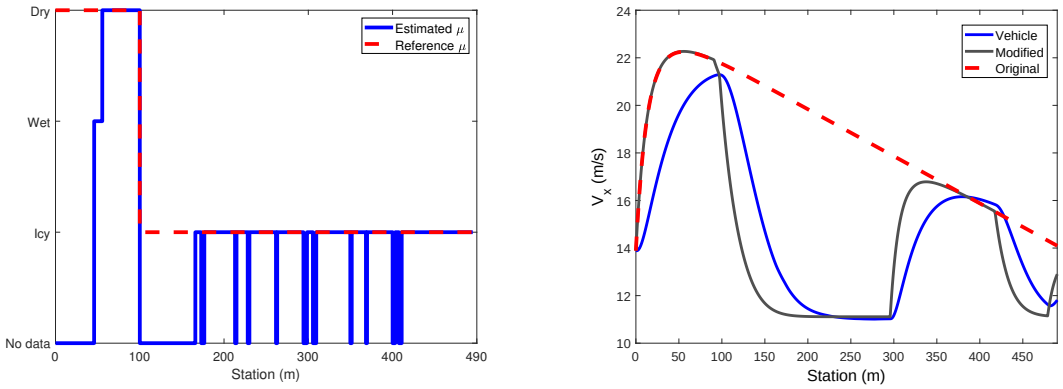
control performances, there exists no elaborate methodology to prove or guarantee the stability for most of them. In this thesis, novel approaches are presented, which combine the advantages of the presented model-based and data-driven methods. The main goal is to improve the performances of the vehicle control system by using machine learning algorithm while preserving the stability guarantees provided by the model-based solutions.

1.1 Motivational example for the combination of model-based and data-driven control

In following, a motivational example is presented for the application of data-driven control of automotive-related problems. For example, it can be used to improve the stability of the vehicle by determining the stable regions of the vehicle, see [FNG18a]. Other application of the data-driven method could be the estimation of the adhesion coefficient, e.g. [FNG21c].

In the following example, the model-based control strategy is extended with the result of data-driven algorithm, i.e. with the learning-based estimation of the road surface adhesion coefficient. The vehicle is driven along a sharp bend twice, see Figure 1.2. In the first run, the vehicle is driven by the nominal LPV controller, represented by the black car. In the second run, the vehicle is controlled by the extended control algorithm, which can modify its longitudinal velocity using the result of the data-driven analyses, represented by the red vehicle. Briefly, the extended algorithm computes the maximal velocity in a predefined prediction horizon by which the stability of the vehicle can be maintained during the turn.

Figure 1.1(a) shows the estimated adhesion coefficient, which is taken into account by the velocity computation algorithm. The modified speed profile can be seen in Figure 1.1(b). The algorithm decreases the original velocity at the beginning of the turn.

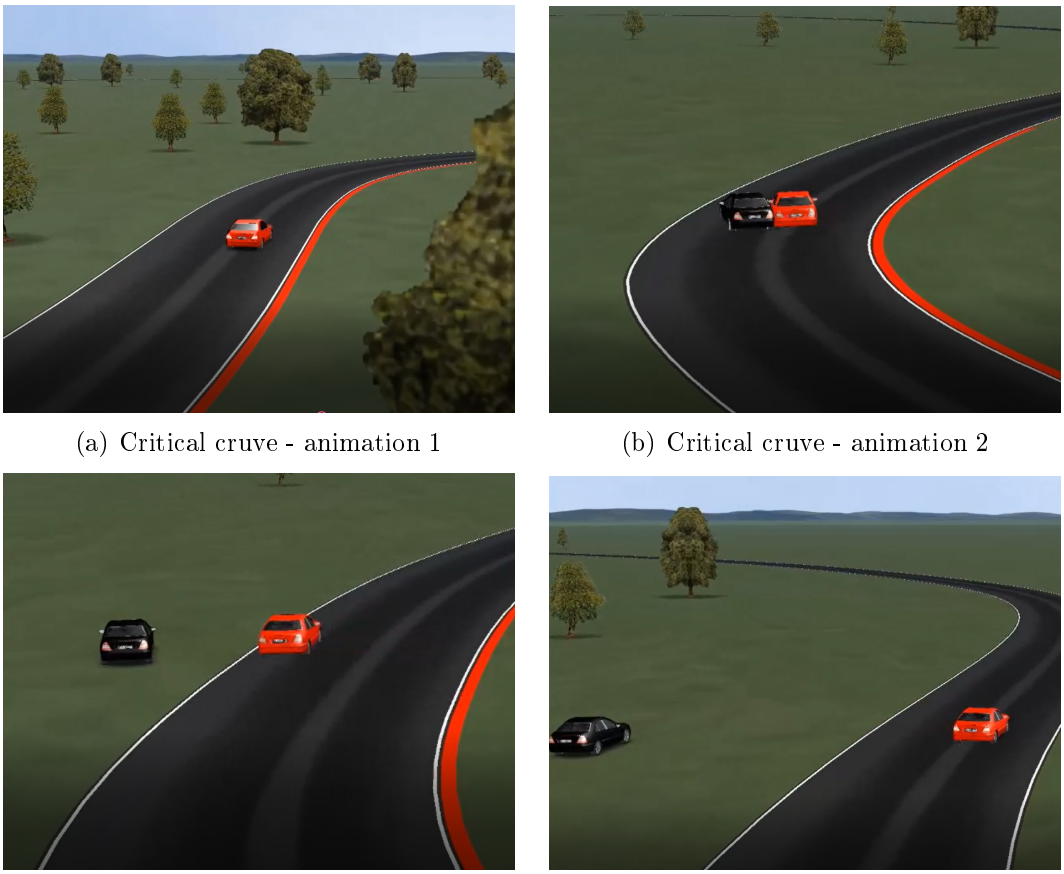


(a) μ estimation (b) Longitudinal velocity

Fig. 1.1: μ estimation and longitudinal velocity

This modification results in the maintained stability of the vehicle as it is shown in Figure 1.2. It can be seen that the black car leaves the road at the bend, while the red one, which is controlled by the data-driven strategy, can follow the road.

Beside this example, there are several ways the data-driven methods can improve the stability and the performances of the vehicle-related control algorithms. In this theses, different algorithms will be presented, which combine the advantages of the model-based solutions and the data-driven methods providing better control performances for the vehicle.



(a) Critical cruve - animation 1

(b) Critical cruve - animation 2

(c) Critical cruve - animation 3

(d) Critical cruve - animation 4

Fig. 1.2: Illustrative example of data-driven control

The stable regions of the vehicle, which is determined by a data-driven algorithm is illustrated in Figures 1.3. The black line shows the trajectory of the data-driven controlled vehicle. As it shows, the trajectory remains inside these regions, which guarantees the stability of the vehicle.

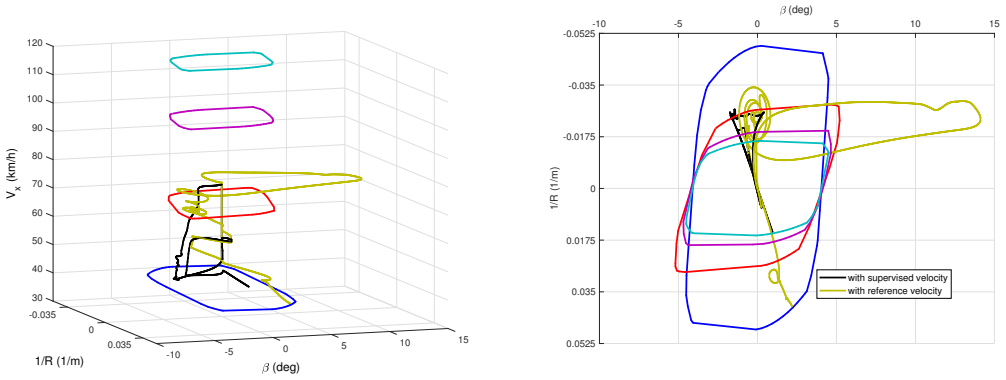


Fig. 1.3: Data-driven stability regions analysis

1.2 Overview of Variable Geometry Suspension Systems

Through the automation of road vehicles, several novel actuators have been built-in. Especially, the small-scaled urban vehicles have challenges in the field of automation, because small and low-cost smart actuators are requested. An innovative solution for steering purposes is the concept of variable-geometry suspension (VGS). VGS systems can be beneficial for several vehicle-oriented problems such as: trajectory tracking or increasing the comfort-related performances. In, general VGS has a simple structure and its energy demand is also low, see [41, 42]. Since two theses of this dissertation deals with the modeling and the control problems of the variable geometry suspension systems, in the following, a brief introduction is given to this component.

More precisely, the term "variable-geometry suspension" does not denote a specific suspension system but a group of it. Therefore, several different structures have been developed, devoted to specific problems and tasks. One are designed to improve the vertical motion of the vehicle. These VGS systems can modify the stiffness of the suspensions, see [43, 44]. [45] presents a solution, in which the roll center of the vehicle can be modified by a VGS system. In this way, the roll angle of the vehicle can be significantly reduced. VGS system can also be used to improve the pitch control of the vehicle, see [46].

Another important group of variable-geometry suspensions focuses on the improvement of the lateral dynamics. In several conceptions the variable-geometry suspension is applied on the rear wheels, and its role is to influence on the toe-angle, see [47, 48]. A front wheel suspension control, in which the functionality of trajectory tracking and roll angle minimization are in the focus is presented in [49],[NGFB17a]. In most of these solutions the positioning of the wheel has a direct influence on the lateral dynamics, which results in the generation of tyre force through the steering or tilting of the wheel. However, there is another novel conception of variable-geometry suspension, in which the actuator intervention is connected to the steering functionality through only its dynamics [50, 51]. The aim of this solution is the modification

of the geometry, which results in a change in the camber angle and in the position of the wheel-road contact. Thus, the scrub radius is also modified. Consequently, a longitudinal force on the wheel creates a moment on the wheel and, thus, it realizes steering angle. The advantages of this suspension are the integration possibility with the driving and the enhanced independent steering functionality with increased maneuverability.

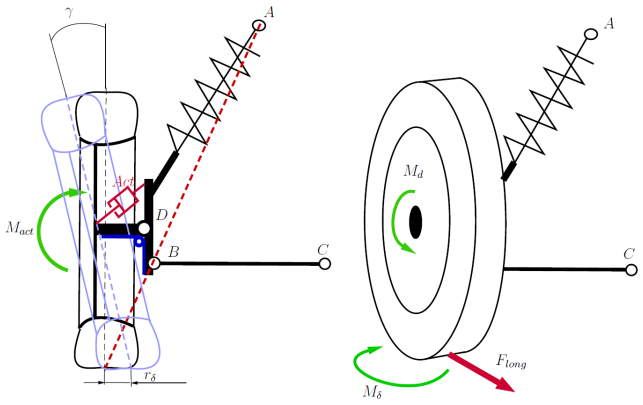


Fig. 1.4: Scheme of the suspension construction

In this thesis a novel conception of variable-geometry suspension is considered, which results in the variation of the steering angle, see Figure 1.4. In the conventional applications the aim of the variable-geometry suspension is to modify the camber angle of the wheel, by which a lateral force between the tyre and the road is generated [52, 53]. Since the impact of the camber angle on the lateral wheel force is limited, the conventional system requires large camber angle, which can be disadvantageous from the aspect of the tyre.

In this application, the main goal is to generate an additional steering angle through the modification of the camber angle. The idea behind this concept is that the longitudinal force on the wheel (F_{long}) can generate torque (M_δ) on the scrub radius (r_δ), by which the suspension can rotate around the axis of steering. Scrub radius can be defined as the intersection of the road surface and the axis of joints A, B. Through the modification of the camber angle (γ), the scrub radius can also be modified, by which torque M_δ can be influenced. In this way, an additional steering angle can be generated in both directions. In this concept, the camber angle is modified by using an actuator (Act) placed on the hub of the suspension.

In this thesis, a modeling a hierarchical control design is presented for variable-geometry suspension-based independently steering system. Furthermore, for validating purposes a suspension testbed has been built in SZTAKI Institute for Computer Science and Control. For controlling the testbed, a data-driven control design method is proposed.

1.3 Structure of the thesis

Figure 1.5 summarizes the main chapters of this thesis and the connections among them. Chapter 2 presents vehicle-oriented estimation algorithms, which are based on different machine learning methods. In chapter 3, a machine learning-based stability analysis is presented for autonomous vehicles. Moreover, based on the result of the stability analysis, and using the estimation algorithm, presented in the previous chapter, a complex lateral control strategy is proposed for autonomous vehicle. Chapter 4 presents two different parameter optimization methods for determining the parameters of the lateral bicycle model. Chapter 5 proposes a modeling and control design of a variable-geometry suspension-based independent steering system. Chapter 6 presents a data-driven modeling and control strategy for a variable-geometry suspension testbed. The presented control algorithm is also implemented and tested on the suspension testbed. Finally, Chapter 7 summarizes the results of this thesis and details the future challenges.

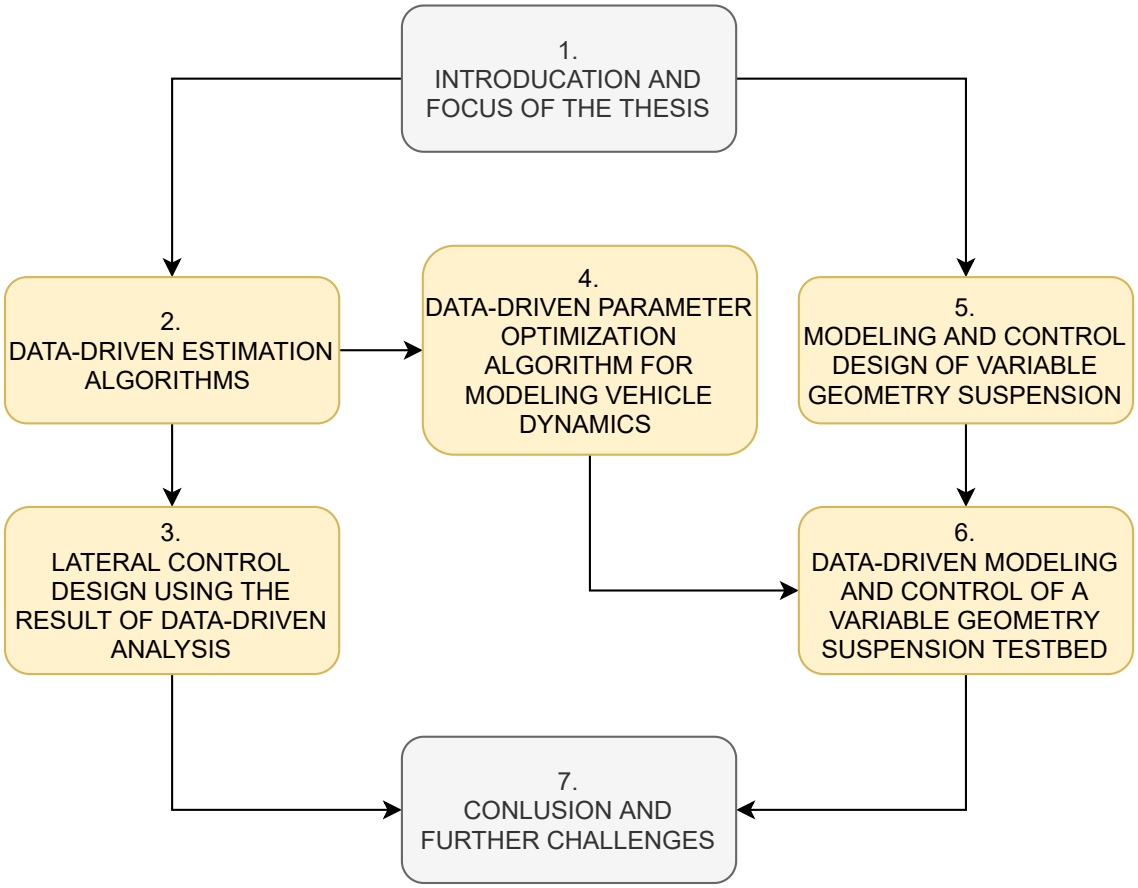


Fig. 1.5: Structure of the thesis

2. VEHICLE-ORIENTED ESTIMATION METHODS USING DATA-DRIVEN APPROACHES

In the era of autonomous vehicle the accurate knowledge of the vehicle's states and its environment is a crucial task in order to guarantee the safe motion of the vehicle. However, some states of the vehicle cannot be measured directly or the measurement process is expensive. Therefore, several estimation algorithms have been developed for this problem. Another problem arises from the nonlinear dynamics of the vehicle, which makes the development of the estimation algorithm difficult and, in general, requires high computational cost. However, the machine-learning-based algorithms are capable of dealing with highly nonlinear problems while providing satisfactory performances. Another aspect, which must be taken into consideration during the design phase, is the signal noises. It this study, filtered signals are investigated, which means that a filtering algorithm (e.g. Kalman-filter) may be required before applying the proposed solutions. In this chapter data-driven estimation methods are presented for autonomous vehicles. More specifically, two different estimation problems are addressed:

- tyre pressure estimation using two methods, pace-regression and neural network.
- Adhesion coefficient estimation based on C4.5 decision tree algorithm.

Furthermore, an application example of the proposed data-driven tyre pressure estimation method is also presented, in which the estimation algorithm is used in a lateral control design in order to enhance the stability and the tracking performance of the closed-loop system.

2.1 Tyre pressure estimation with machine-learning-based algorithms

Firstly, data-driven tyre pressure estimation algorithms are presented. The estimation problem is solved by using two different approaches: pace-regression and neural network. In both cases, the estimation algorithm relies on only those variables, which are accessible from the vehicle's onboard systems. Moreover, the application of the presented estimation model is used in a lateral control design. This section consists of the following main parts:

- *Data acquisition:* In this study, the dataset, which is required by the machine-learning-based algorithms, is provided by the high-fidelity simulation software,

CarMaker. Several simulations have been performed using the simulation environment in order to a wide range of the vehicle dynamics. More details can be found in Subsection 2.1.1.

- *Tyre pressure estimation using pace regression:* In the first case, a pace regression-based estimation algorithm is developed for estimating the change of the tyre pressure of the front left wheel. This method is presented in Subsection 2.1.1.

- *Tyre pressure estimation using neural network:* In the second case, a neural network-based algorithm is proposed for the aforementioned estimation problem. Since the neural network-based algorithm can be used to approximate multiple signals, in this case, the pressures of both front tyres are considered during the estimation process. This algorithm is investigated in Subsection 2.1.2.

- *Application example of the proposed method:* Finally, a possible application of the proposed algorithms are presented through a lateral control example. The estimated tyre pressure is used as a scheduling parameter in an extended lateral model, which is the basis of the control design. In this way, the performances of the control system is enhanced, which is demonstrated through a comprehensive simulation example. The control design and the simulation example are detailed in Subsection 2.2.

2.1.1 Tyre pressure estimation using pace regression

The first step is the acquisition of the appropriate data, with which dataset for the training and validation of machine-learning-based algorithms can be provided. For this purpose several simulations have been performed in the simulation environment, which is the high-fidelity vehicle software CarMaker. During these simulations, the tyre pressure and the longitudinal velocity have been modified.

The longitudinal velocity varies in the interval $11m/s...15m/s$, whilst, the pressures of the front tyres change between $1.0 - 2.5bar$ with the step size $\Delta p = 0.1$ bar. The velocity is set by a built-in PID-based cruise control model in CarMaker, see [54]. During the simulations several parameters are measured and saved such as yaw-rate, accelerations and velocities in various directions, steering angle, tyre forces etc. The sampling time is set to $T_s = 0.01s$. In this way, more than one million distinct instances are collected.

In the first case, the pace regression algorithm is applied to estimate the tyre pressure of the vehicle. Briefly, the pace regression algorithm is an improved version of the classical linear regression, which addresses some of the main drawbacks of it by using cluster analysis. This algorithm was developedy by Yong Wang , Ian H. Witten, see [55]. In the followings, a brief introduction is given to this method.

Brief introduction to pace regression

Consider a dataset with n independent instances, k input variables and an output variable. The instances are written in the form of an $n \times k$ design matrix X. Moreover, let ζ^* be the parameter vector of the true model and then the output vector y can be determined as

$$y = X\zeta^* + \epsilon, \tag{2.1}$$

where ϵ is the noise vector whose elements are sampled from $N(0, \sigma^2)$. It is assumed that σ^2 is known or, at least, it can be estimated ($\hat{\sigma}^2$). $\mathcal{M}(\zeta)$ denotes a fitted, linear model that has an unique parameter vector ζ while the true model is denoted by $\mathcal{M}(\zeta^*)$. The aim of the modeling task is to find a model from the entire model space $\mathbb{M} = \{\mathcal{M}(\zeta) : \zeta \in \mathbb{R}^k\}$ whose predictive accuracy is the greatest on the given dataset. The models can be produced by numerous algorithms such as Ordinary Least Square (OLS). method, OLS subset selection, shrinkage, RIC, CIC methods etc, see [55]. These methods can reduce the dimension of the models by discarding the redundant variables. In order to evaluate a model, the distance between the current model and the true model must be known. This distance can be calculated as

$$\mathcal{D}(\mathcal{M}(\zeta^*), \mathcal{M}(\zeta)) = \frac{||y_{\mathcal{M}(\zeta^*)} - y_{\mathcal{M}(\zeta)}||^2}{\sigma^2}, \tag{2.2}$$

where $||\cdot||$ denotes the $\mathcal{L}_2$ norm and σ^2 can be replaced by its estimated value $\hat{\sigma}^2$. The final task is to determine a model which minimizes this expression.

$$\mathcal{D}(\mathcal{M}^*, \mathcal{M}) = min! \tag{2.3}$$

The basic concept of the OLS subset selection method is to create subset models using various sets of variables. If a dataset has k variables, $k+1$ nested models ($\mathcal{M}_j$) can be created, where $j = 0$ is the null model with zero variables and $j = k$ is the full model with all of the variables. In this case, an estimate of the parameter vector of $\mathcal{M}_j$ can be determined as

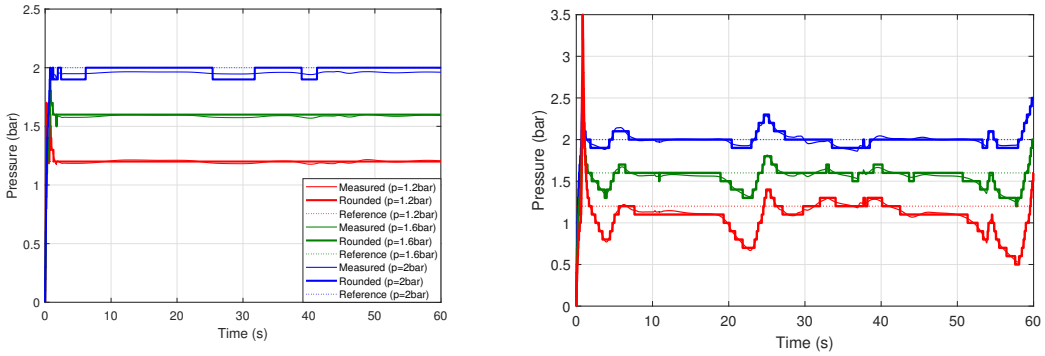
$$\hat{\zeta}_{\mathcal{M}_j} = (X'_{\mathcal{M}j} X_{\mathcal{M}j})^{-1} X'_{\mathcal{M}j} y \tag{2.4}$$

where $X_{\mathcal{M}_j}$ is the $n \times j$ design matrix and let $\mathcal{P}_{\mathcal{M}_j} = X_{\mathcal{M}j}(X'_{\mathcal{M}j} X_{\mathcal{M}j})^{-1} X'_{\mathcal{M}j}$ be an orthogonal projection matrix from the original space (k) onto the reduced space (j). Finally, $\hat{y}_{\mathcal{M}_j} = \mathcal{P}_{\mathcal{M}_j} y$ is the estimate of $y^*_{\mathcal{M}_j} = \mathcal{P}_{\mathcal{M}_j} y^*$.

Since this method creates only $k+1$ subset models (instead of 2^k, which is computationally unfeasible at increased k), the predefined order of the variables is a crucial point of this method. There are a set of ranking algorithms, which help determine the best predefined order of the variables, see [56, 57].

Evaluation of the pace regression-based tyre pressure estimation algorithm

In the followings, the results of the pace regression-based estimation using different clusters of the measured attributes are presented. In the first case all of the collected attributes in the regression are used, the machine-learning-based algorithm provides accurate estimation. Figure 2.1 (a) shows the results of the estimation for three different cases, when the pressure of the tyre is set to 1.3 bar, 1.7 bar and 2.1 bar, respectively, and the velocity is set to 14 m/s. The elements of X is selected as the actual values of various measured signals, i.e., longitudinal and lateral accelerations a_x, a_y, longitudinal and lateral velocities v_x, v_y, yaw-rate $\dot{\psi}$, angular velocity of wheels W_{ij}, steering angle of front wheels δ_i, side-slip angle β, roll rate $\dot{\phi}$. The calculated pressures values are rounded to eliminate fluctuation in the estimation.



(a) Estimation of the pressure at constant velocity

(b) Estimation of the pressure CarMaker Driver

Fig. 2.1: Estimation of the tyre pressure at varying velocity

Although the previous estimation seems to be accurate enough, the velocity has been set to constant, which is unrealistic in various traffic scenarios. Thus, the velocity of the vehicle varies by using the CarMaker Driver, which is the in-built model of the simulator. The estimated pressure is illustrated in Figure 2.1 (b).

As the velocity varies during the simulation, the estimation becomes less accurate. Therefore, a new dataset is generated, in which the reference velocity of the vehicle varies randomly during the simulation. However, in this case the pace regression provides worse results using only the actual values of the attributes. Therefore, the past values of the variables must be also used. The time interval between two consecutive points is set to $T = 1/6Hz = 0.15\ s$, which is a suitable value for all onboard sensors.

Moreover, in practice some of the attributes cannot be measured directly, e.g. the forces on the tyres or the side slip angles. Therefore, only the signals, which are available from the on-board system (velocity, steering angle, wheel speeds, yaw rate and accelerations) are selected in the generation of the regression model. Using the selected data, a new estimation model is built up. The following table shows correlation coefficients of the generated models and the mean errors.

All data	Past data	Const. velocity	Accuracy	Av. error
✓	✗	✓	97.9	0.0063
✗	✗	✓	69.19	0.072
✓	✓	✓	96.38	0.0194
✗	✓	✓	90.1	0.047
✓	✓	✗	19.5	0.129
✗	✓	✗	9.8	0.29
✗	✗	✗	22.3	0.91

Tab. 2.1: Table of the accuracy

The result of the above simulations has shown that using all of the attributes in the estimation results in the accuracy of the estimation is the highest. When only the measurable signals are used, the accuracy is reduced. However, using also the past values of attributes, the accuracy of the pressure estimation increases. In both cases, the estimation was inaccurate when the velocity varies during the simulation. Thus, new dataset is generated, in which the reference value of the velocity varies randomly. The results obtained by using the new dataset are shown in the Table 2.1, but in this case only the results of the relevant cases are shown. In the table, mark '✓' denotes that the given attribute or setting is used during the training process while '✗' indicates that the given attribute is excluded from the training set.

All data	Past data	Const. velocity	Accuracy	Av. error
✗	✗	✗	26.7	0.101
✗	✓	✗	57.4	0.14

Tab. 2.2: The accuracy of the estimation using the new dataset

As Table 2.2 shows, in the second case, the accuracy is high close to 60% with a reasonable mean error of 0.14. This means that the algorithm is able to estimate the tyre pressure well.

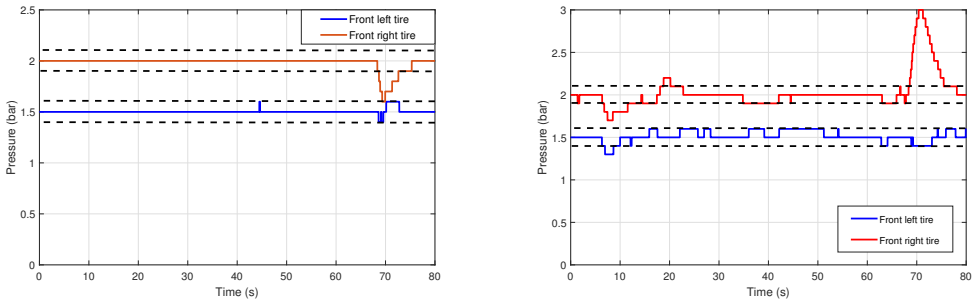
2.1.2 Neural network-based tyre pressure estimation

In the followings, another solution is presented for the tyre pressure estimation problem. In this case, a neural network is applied using different sets of attributes similarly to the previous, pace regression-based solution. Firstly, the accuracy and the performances of the applied neural networks are detailed. Secondly, a possible application example is shown, in which the resulted neural network is used in a lateral control design to improve the performances of the control system by taking into account the change in the tyre pressure.

The neural network, used for the tyre pressure estimation consists of one input, one output and 3 hidden layers. The hidden layers contain 55-45-55 neurons. The numbers of the hidden layers and the neurons are determined by using the so-called

k-field cross validation technique. Initially, this method divides the dataset into two subsets. The first subset is the training set, which used for training the network. The other subset is the test set which is used for evaluating the neural network. Moreover, another crucial part of the network is the used activation functions. As mentioned, there is a lot of functions that can be used in the training process. In this case, the rectified linear unit (ReLU) and the log-sigmoid functions are used since they can be easily adjusted to nonlinear problems. For training the network, the Levenberg-Marquardt algorithm is used, which is a well-known optimization method in the field of machine-learning techniques [58].

In the followings, the results of the deep learning based estimation are presented and illustrated through examples. Several neural networks have been created using different attributes to find the best applicable estimation model. For example, when all of the collected attributes are used, the neural network provides accurate estimation. In Figure 2.2 (a), the estimated pressure is illustrated, when the velocity of the vehicle is controlled by CarMaker Driver, which is the in-built model of the simulation software. As in the previous case, the output of the neural network is



(a) Estimation of the pressure CarMaker Driver

(b) Without past data

Fig. 2.2: Estimation of the pressure CarMaker Driver

rounded to 0.1. Since the measurements are exposed to noises and the disturbances, a $0.2bar$ wide interval is determined and inside this interval the estimation is considered to be acceptable. The dashed lines illustrate the borders of the interval. Apart from a short section at the end of the simulation, the estimated pressure values are accurate, because the reference values of the tyres are set to $1.5bar$ and $2.0bar$ respectively. Since not all of the collected attributes can be directly measured on the car, new neural networks are built up using solely the variables that are available from the on-board system, such as: wheel speeds, accelerations, longitudinal velocity, yaw-rate and steering angle.

The result of the new neural network can be seen in Figure 2.2 (b). As the figure shows, this network provides worse estimation than the previous one, but its result is still accurate enough to be used in the control system. In order to improve the estimation, the past values of the measurements are also used in the following neural

network. The same time interval has been used $T = 1/6Hz = 0.15\ s$ as in the case of pace regression.

Using the extended training set, a new estimation model with the past values is built up. The result of the new network is shown in Figure 2.3. Although it can be seen that this network provides more fluctuating output, the peak value of error signal decreased significantly.

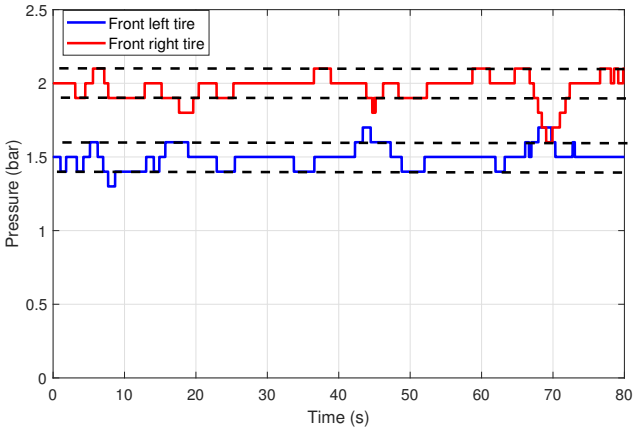


Fig. 2.3: With past data

In the Table 2.3, the comparison of the presented neural networks is detailed, where the accuracy and the averaged errors of the networks are shown. The accuracy is calculated using the aforementioned intervals. As the figures have shown, the best result is given by that network, which uses all of the attributes (100% and 96.1%). However, not all of the attributes can be measured directly from the onboard system, therefore this network is not applicable in the control system. The second best estimation is provided by that network, which uses the available variables and their past values. Although the accuracy of this network is worse (90.3% and 94.4%), its averaged error is small (0.0055 and 0.0554). It means that the output of this network is close enough to the actual pressures of the tyres.

Tyre (L/R)	All data	Past data	Accuracy	Av. error
Left	✓	✗	100%	0.00
Right	✓	✗	96.1%	0.0072
Left	✗	✓	90.3%	0.01522
Right	✗	✓	94.4%	0.00552
Left	✗	✗	83.3%	0.0554
Right	✗	✗	92.2%	0.01

Tab. 2.3: Table of the accuracy using the new dataset

2.2 LPV-based control design using the tyre pressures

In the followings, an application example of the proposed tyre pressure estimation algorithm is presented for trajectory tracking problem. The output of the tyre pressure estimation is used in the control system as a scheduling parameter. The control algorithm has another scheduling parameter, which is the longitudinal velocity of the vehicle. The control system has two control inputs, such as the steering angle and the differential driving, which compensates the loss of lateral force due to the change of the tyre pressure. The entire structure of the control system is depicted in Figure 2.4 including the training process of the machine learning algorithm. The algorithm is divided into three layers which are the followings: *Simulation environment*, *tyre pressure estimation* and *Control system*. The *Simulation environment* serves to validate the algorithm, in which *CarMaker* simulation software is used. The *Control System* consists of the main steps of the control design, which can be divided into two sub-layers. The upper sub-layer generates the reference trajectory, and the lower one is responsible for the control of the vehicle. In this section the *tyre pressure estimation* layer is presented in detail.

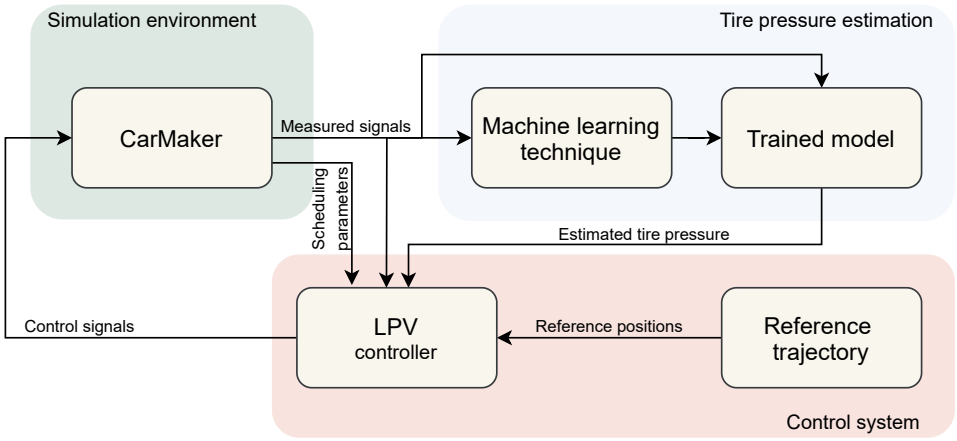


Fig. 2.4: Structure of the control system

Modeling of the vehicle dynamics

Throughout this dissertation, the lateral dynamics of the vehicle is modeled by the single-track bicycle model. The basic idea behind this model is to replace the wheels on the front and rear axles by one-one virtual wheel placed on the axis of symmetry of the vehicle as illustrated in Figure 2.5. The model consists of two main equations, see [59]. The first one describes the lateral acceleration of the vehicle. The second equation describes the yaw motion of the vehicle:

$$ma_y = F_{yf}(\alpha_f) + F_{yr}(\alpha_r), \tag{2.5a}$$

$$\ddot{\psi} I_z = F_{yf}(\alpha_f) l_1 - F_{yr}(\alpha_r) l_2, \tag{2.5b}$$

where $F_{y,i}$ denote the lateral forces on the front and rear wheels, m represents the mass of the vehicle, $\dot{\psi}$ is the yaw-rate, I_z denotes the yaw-inertia, α_i are the slip angles of the front and rear wheels, l_i are the distances between the axles of the center of gravity (CoG).

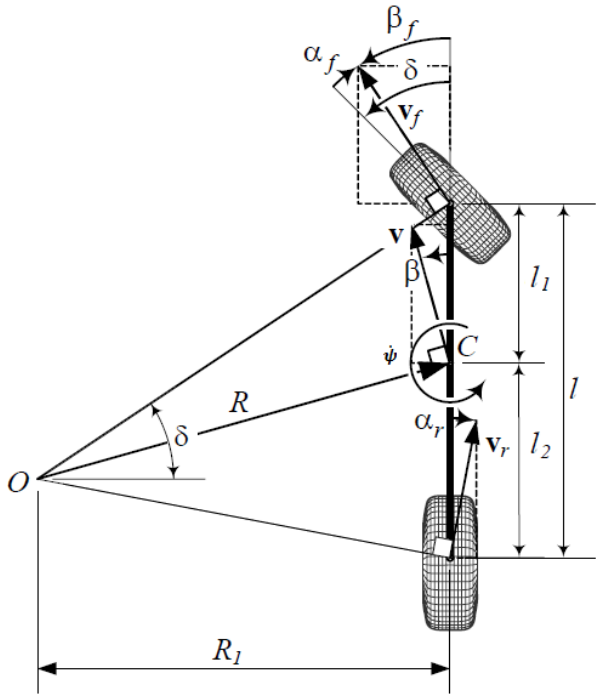


Fig. 2.5: Single-track lateral vehicle model [59]

The lateral acceleration a_y consists of two main components:

$$a_y = \dot{v}_y + \dot{\psi} v_x(t), \tag{2.6}$$

The lateral acceleration consists of two main components:

$$F_{yi}(\alpha_i) = C_{\alpha i}\alpha_i, \tag{2.7}$$

where $v_x(t)$ is the longitudinal velocity, which is an external variable of the model, and v_y denotes the lateral velocity.

There are two main sources of the nonlinearities in this model. The first one is caused by the longitudinal velocity, while the second one is induced by the relationship between the lateral force (F_{yi}) and the corresponding side-slip α_i. As Figure 2.6 shows, this relationship can be illustrated by a set of functions depending on several other factors, such as adhesion coefficient, vertical load.

A general assumption is to consider this function only for small slip angles (α_i) and for constant vertical load and adhesion coefficient. In this way, the relationship between the lateral force and slip angles can be described as a linear function using

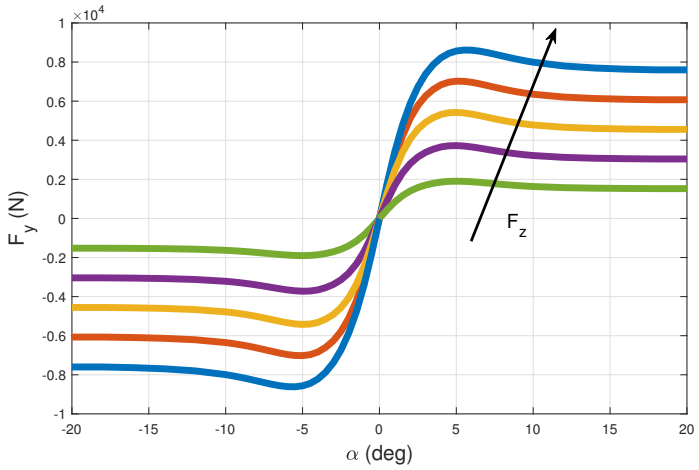


Fig. 2.6: Lateral force and side-slip

the cornering stiffness C_{α_i}. Then, considering only front wheel steering, the slip angle of the front and rear axles can be computed as:

$$\alpha_f = \frac{1}{v_x(t)}(v_y + l_1\dot{\psi}) - \delta = \beta + \frac{l_1\dot{\psi}}{v_x(t)} - \delta \tag{2.8a}$$

$$\alpha_r = \frac{1}{v_x(t)}(v_y - l_2\dot{\psi}) = \beta - \frac{l_2\dot{\psi}}{v_x(t)} \tag{2.8b}$$

where $\beta = \frac{v_y}{v_x(t)}$ is the side-slip of the CoG of the vehicle and δ is the front wheel steering angle.

Using equations (2.5a) and (2.5b), the original model can be rewritten into the following form:

1. The lateral motion equation of the vehicle:

$$m\dot{v}_y + m\dot{\psi}v_x(t) = \left(-\frac{l_1}{v_x(t)}C_{\alpha,f} + \frac{l_2}{v_x(t)}C_{\alpha,r}\right)\dot{\psi} - (C_{\alpha,f} + C_{\alpha,r})\frac{v_y}{v_x(t)} + C_{\alpha,f}\delta \tag{2.9}$$

2. The yaw-motion of the vehicle:

$$\ddot{\psi}I_z = \left(-\frac{l_1^2}{v_x(t)}C_{\alpha,f} - \frac{l_2^2}{v_x(t)}C_{\alpha,r}\right)\dot{\psi} - (l_1C_{\alpha,f} - l_2C_{\alpha,r})\frac{v_y}{v_x(t)} + l_1C_{\alpha,f}\delta \tag{2.10}$$

Note that the presented lateral model is an underactuated system since it has only one control input (δ) while it has two state-variable ($\dot{\psi}, v_y$). Furthermore, this model is suitable only for high longitudinal velocities, i.e., $v_x > 5\frac{m}{S}$. Under this longitudinal velocity, a kinematic model is applicable to describe the lateral motion of the vehicle, see [59].

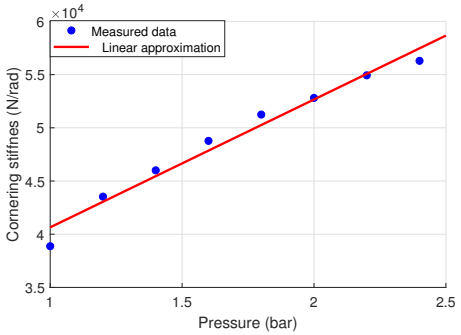


Fig. 2.7: Relationship between pressure and cornering stiffness

Modeling the effect of the tyre pressure on the lateral force

The lateral dynamics of the vehicle is modeled by the two-wheeled bicycle model (2.5). The original model is extended with two additional terms,$\Delta F(p_1)$ and M_d:

$$I\ddot{\psi} = F_{f,y}(\alpha_f)l_1 - F_{r,y}(\alpha_r)l_2 + M_d + \Delta F(p_1)l_1, \tag{2.11}$$

$$mv_x(\dot{\psi} + \dot{\beta}) = F_{f,y}(\alpha_f) + F_{r,y}(\alpha_r) + \Delta F(p_1), \tag{2.12}$$

where M_d is the differential torque, which is used as a second control signal for steering the vehicle. p_1 represents the mean pressure of the front tyres. $\Delta F(p_1)$ is determined from the estimated tyre pressure. Although the pressure p_1 does not appear in the equations, this variable highly correlates to the cornering stiffness, which is determined as [60]:

$$F_{f,y}(p_1) = C_{\alpha,f}(p_1)\alpha. \tag{2.13}$$

The relationship between the pressure and the cornering stiffness is assumed to be linear based on the simulations on CarMaker. The maximum value of the the cornering stiffness is considered to be $57000N/rad$, while the minimum is $39000N/rad$, as illustrated in Figure 2.7. The presented lateral vehicle model can be transformed into a parameter-varying state-space representation (see Appendix .1), which is formed as:

$$\dot{x}_v = A_v(v_x, p_1)x_v + B_v(v_x, p_1)u_v, \tag{2.14}$$

$$A_v = \begin{bmatrix} -\frac{l_1^2 C_{\alpha,f}(p_1) + l_2^2 C_{\alpha,r}}{I_z v_x} & -\frac{l_1 C_{\alpha,f}(p_1) - l_2 C_{\alpha,r}}{I_z v_x} & 0 \\ \frac{-l_1 C_{\alpha,f}(p_1) + l_2 C_{\alpha,r}}{m v_x} - v_x & -\frac{C_{\alpha,f}(p_1) + C_{\alpha,r}}{m v_x} & 0 \\ 0 & 1 & 0 \end{bmatrix}, \tag{2.15}$$

$$B_v = \begin{bmatrix} \frac{l_1 C_{\alpha,f}(p_1)}{I_z} \\ \frac{C_{\alpha,f}(p_1)}{m} \\ 0 \end{bmatrix}. \tag{2.16}$$

The state vector x_v consists of the signals $x_v = [\dot{\psi}, v_y, y]^T$. The control inputs of the system are $u_v = [\delta, M_d]$, where δ denotes front wheel steering angle. The signals v_x, p_1 are scheduling variables of the system. Since u_v contains two control input signals, the under-actuated characteristics of the system is eliminated.

Modeling the dynamics of the steering system

The dynamics of the steering system can have a significant impact on the performances of the lateral control system. Therefore, it must be taken into account during the control design of the vehicle. The dynamics of the steering system is described by the following state-space representation:

$$\dot{x}_s = A_s x_s + B_s u_s, \quad y_s = c_s^T x_s, \tag{2.17}$$

where u_s is the angle of the steering wheel, y_s is the steering angle of the front wheels, A_s, B_s and c_s^T are matrices. The state vector x_s consists of the states of the steering system.

In practice, the determination of the parameters in the state-space representation (2.17) from physical relations of the steering dynamics can be difficult. Therefore, an identification process to compute the required parameters is performed. In [61], a second-order form is proposed for modeling the dynamics of the steering system, which can be given by the following transfer function:

$$G_s(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^2 + a_1 s + a_0}, \tag{2.18}$$

where b_i and a_i are the parameters, which must be determined through an identification process.

In the followings the model formulation of auto-regressive with exogenous input (ARX) identification structure is used for the determination of the system parameters. The ARX structure is formed as [62]:

$$j(t) + a_1 j(t-1) + ... + a_{n_a} j(t-n_a) = b_1 k(t-1) + ... + b_{n_b} k(t-n_b) + e(t), \tag{2.19}$$

where j denotes the output of the system, in this case, the steering angle of the front wheels. Moreover, k denotes the input of the system, which is the angle of the steering wheel. $e(t)$ is the error function. The parameters can be written into a parameter vector:

$$\sigma = [a_0 \quad a_1 \quad ... \quad a_{t-n_a} \quad b_0 \quad b_1 \quad ... \quad b_{t-n_b}]^T. \tag{2.20}$$

By using shift operator q^{-1}, the equation (2.19) can be divided into two equations:

$$A(q) = 1 + a_1 q^{-1} + ... + a_{n_a} q^{-n_a}, \tag{2.21}$$

$$B(q) = b_1 q^{-1} + ... + b_{n_b} q^{-n_b}. \tag{2.22}$$

Finally, the transfer function of the identified system can be calculated as:

$$G(q,\sigma) = \frac{B(q)}{A(q)}. \tag{2.23}$$

The resulted transfer function is a discrete-time system, which means that the resulted system must be transformed into a continuous form to achieve state-space representation (2.17). For the transformation $T_s = 0.01s$ sample time and a zero-order hold element are used.

Design method of the LPV-based controller

The presented two state-space representations (2.14),(2.17) are combined and written into an extended representation:

$$\dot{x_e} = A_e(v_x, p_1)x_e + B_e(v_x, p_1)u_e, \tag{2.24}$$

where u_e=$[u_s\ M_d]^T$ and $x_e = [x_s\ x_v]^T$, while the matrices are:

$$A_e(v_x, p_1) = \left[\begin{array}{c|c} A_s & 0_{2\times4} \\ \hline B_{v,1}(v_x, p_1)C_s^T & A_v(v_x, p_1) \end{array}\right], \tag{2.25a}$$

$$B_e(v_x, p_1) = \left[\begin{array}{c|c} B_s & 0_{2\times1} \\ \hline 0_{4\times1} & B_{v,2}(v_x, p_1) \end{array}\right], \tag{2.25b}$$

where $B_{v,i}(v_x, p_1)$ denotes the i^{th} column of $B_v(v_x, p_1)$.

The control system is responsible for guaranteeing the trajectory tracking of the vehicle and minimizing the interventions. Therefore, the following four performances are defined.

- *Minimization of the lateral error*
  In order to reach appropriate tracking performance, the control system has to minimize the lateral error between the road y_{ref} and the lateral position of the vehicle y :

$$z_2 = y_{ref} - y, \qquad |z_1| \to min, \tag{2.26}$$

- *Minimization of the yaw-rate error*
  Beside the lateral error, the controller has to reduce the error between the reference $\dot{\psi}_{ref}$ and the measured yaw-rate $\dot{\psi}$ in order to reach accurate and smooth tracking.

$$z_1 = \dot{\psi}_{ref} - \dot{\psi}, \qquad |z_1| \to min, \tag{2.27}$$

  where y_{ref} is considered to be given.

- *Minimization of the steering angle*
  The control system has to minimize its interventions to reduce the energy consumption, which means the minimization of the steering angle.

$$z_2 = \delta, \qquad |z_2| \rightarrow min. \tag{2.28}$$

- *Minimization of the differential drive*
  Similarly to the third performance, the controller has to minimize the differential torque as well as the steering angle.

$$z_3 = M_d, \qquad |z_3| \rightarrow min. \tag{2.29}$$

The presented performances are summarized in the following vector $z = \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T$, which leads to the performance equation

$$z = C_1 x_e + D_{11} r + D_{12} u_e, \tag{2.30}$$

where C_1, D_{11}, D_{12} are matrices and r contains the signal y_{ref}. In the LPV control design, the presented extended state-space model is employed. In the control design several transfer functions are used to scale the measured signals and to reach the specific performances. In this manner, the required behavior of the system can be induced. The weighting functions and the augmented plant are illustrated in Figure 2.8.

The weighting functions $W_{ref,1}$ and $W_{ref,2}$ are to scale the reference signals y_{ref} and $\dot{\psi}_{ref}$. They are formed as:

$$W_{ref,1} = \frac{0.1}{100s + 1}, \tag{2.31}$$

$$W_{ref,2} = \frac{0.01}{100s + 1}. \tag{2.32}$$

Furthermore the goals of functions $W_{z,1}$ and $W_{z,2}$ are to guarantee the accurate trajectory tracking of the vehicle.

$$W_{z,1} = \frac{s + 1}{s^2 + 2s + 1}, \tag{2.33}$$

$$W_{z,2} = \frac{1}{s + 1}. \tag{2.34}$$

The following two weighting functions weight the performances of the actuations and ensure the balance between them.

$$W_{z,3} = \left(\frac{p_{max}}{p_{est}}\right)^2 \frac{5s + 5}{0.1s + 1} 10^{-2}, \tag{2.35}$$

$$W_{z,4} = \left(\frac{p_{est}}{p_{max}}\right)^6 \frac{1s}{2s + 1} 10^{-1}. \tag{2.36}$$

The last three functions weight the noises on the measured signals.

$$W_{w,1} = 0.002, \tag{2.37}$$
$$W_{w,2} = 0.001, \tag{2.38}$$
$$W_{w,3} = 0.05. \tag{2.39}$$

The reason of the scaling is that the reachable lateral force decreases together with tyre pressure. Therefore, at low pressures, the differential drive compensates the steering. Finally, the roles of the weighting functions $W_{w,1}$, $W_{w,2}$ and $W_{w,3}$ are to scale the noises of the measured signals.

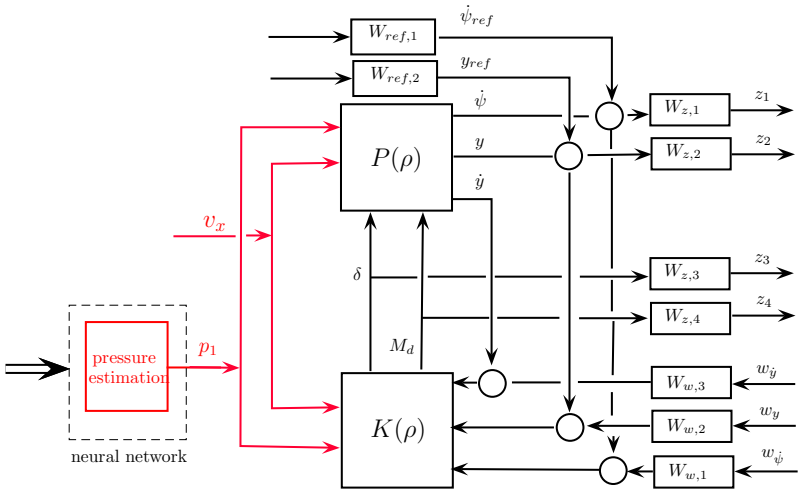


Fig. 2.8: Structure of LPV controller

The quadratic LPV performance problem is to choose the parameter-varying controller $K(v_x, p_1)$ in such a way that the resulting closed-loop system is quadratically stable and the induced $\mathcal{L}_2$ norm from the disturbance and the performances is less than the value γ. The minimization task is the following:

$$\inf_{K(v_x,p_1)} \sup_{v_x,p_1 \in F_\rho} \sup_{\|w\|_2 \neq 0, w \in \mathcal{L}_2} \frac{\|z\|_2}{\|w\|_2}, \tag{2.40}$$

where F_ρ bounds the scheduling variables. The yielded controller $K(v_x, p_1)$ is formed as

$$\dot{x}_K = A_K(v_x, p_1)x_K + B_K(v_x, p_1)y_K, \tag{2.41a}$$
$$u = C_K(v_x, p_1)x_K + D_K(v_x, p_1)y_K, \tag{2.41b}$$

where $A_K(v_x, p_1), B_K(v_x, p_1)$ and $C_K(v_x, p_1), D_K(v_x, p_1)$ are scheduling variable dependent matrices.

Calculation of the reference signals

The calculation of the reference signal is a crucial point in any control system. The capability, delay and other tracking properties of the designed controller must be taken into account. Therefore, not only the actual errors (position and yaw-rate) are used but the predictions of the errors are also calculated. The prediction of the motion of the vehicle is based on the following simple model:

$$x(t+T) = x(t) + v_x \cdot T, \tag{2.42}$$

$$y(t+T) = y(t) + v_y \cdot T, \tag{2.43}$$

$$\dot{\psi}(t+T) = \dot{\psi}(t), \tag{2.44}$$

where x and y are the coordinates of the vehicle.
During the prediction the effects of the accelerations are considered to be infinitesimally small. Therefore, they are neglected in the prediction. In the reference calculation, the prediction is calculated for three time steps: $T_0 = 0.0$, $T_1 = 0.5$ and $T_2 = 0.75$. Then, the summed error signals are formed as:

$$e_{\dot{\psi},p}(t) = 0.5e_{\dot{\psi}}(t) + 0.3e_{\dot{\psi}}(t+T_1) + 0.2e_{\dot{\psi}}(t+T_2) \tag{2.45}$$

$$e_{y,p}(t) = 0.5e_y(t) + 0.3e_y(t+T_1) + 0.2e_y(t+T_2) \tag{2.46}$$

These error signals are used in the proposed control system.

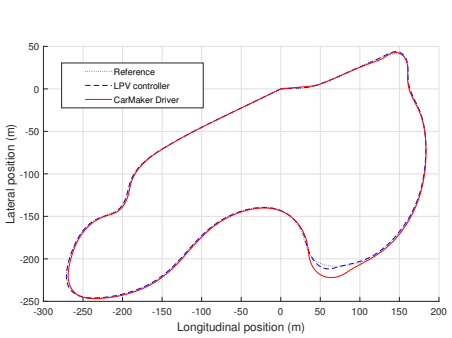
Simulation results

In the rest of this section, a comprehensive simulation is presented to show the efficiency of the proposed control system. In the simulation, the vehicle is driven along the shrunken Melbourne Formula 1 track. Since in the CarMaker environment the pressure of the tyre cannot be modified during the simulation, several runs have been performed using different tyre pressures. In the nominal case, the pressures of the tyres are set to 1 bar and the car is controlled by the CarMaker built-in driver.

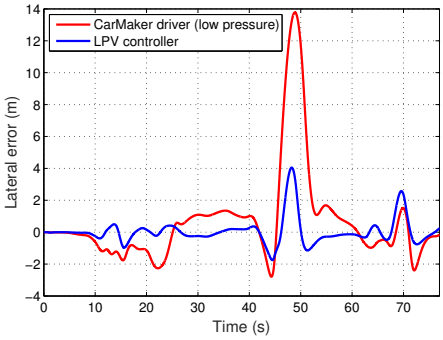
The path of the vehicle is illustrated in Figure 2.9(a). It can be seen that the car is not able to follow the track, it leaves the road at a sharp bend, which is highlighted in Figure 2.9(c). In the second run, the pressure of the tyre is to set to the same value and the car is controlled by the proposed control system. The path of the vehicle is shown in the same figure. As it shows, in contrast to the previous case, the vehicle is able to follow the road using the presented control system. The lateral errors are shown in Figure 2.9. It can be seen that the LPV controller provides smaller error throughout the whole simulation. The reduction in the lateral error is conspicuous between $45 - 50s$, which time interval belongs to the sharp bend.

Figure 2.10 shows the results of the neural network. Since the simulation presents a critical driving situation, the longitudinal and side slips of the vehicle are high, which result in the inaccurate estimation of the pressure. In order to avoid this, the estimation is executed only if the following criterion is satisfied:

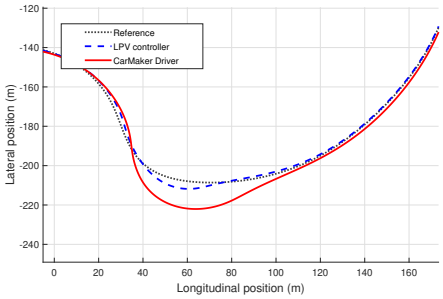
$$|\beta| < 0.075rad \text{ and } |\dot{\psi}| < 0.5rad. \tag{2.47}$$



(a) Whole simulation



(b) Lateral errors



(c) The bend where the driver leaves the road

Fig. 2.9: Positions of the vehicles during the simulations

The blue solid line shows the cases when the mentioned condition is satisfied. The red dashed line represents the cases when the criterion is not satisfied. In this way, the estimation is accurate, its mean error is below < 0.03.

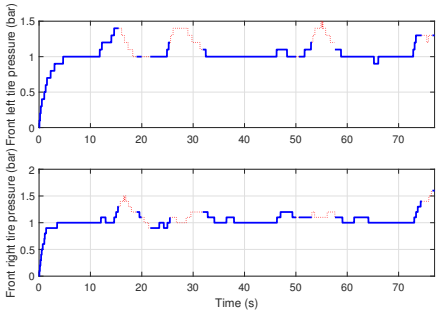
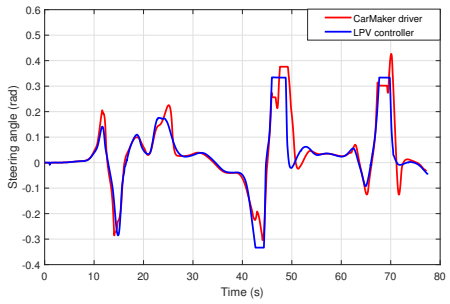


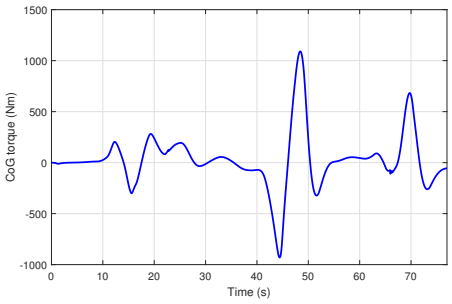
Fig. 2.10: Result of the neural network

Using the actual value of the tyre pressure and the velocity information, the LPV control computes the steering angle and the differential torque. The calculated steering angles are shown in Figure 2.11(a) in both cases. It can been seen that the LPV system provides lower values than the CarMaker Driver, which results in the

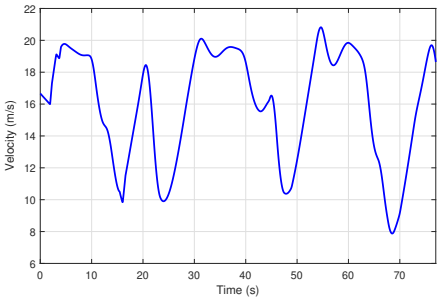
stable motion of the vehicle. Furthermore, the lower steering values are compensated by the differential torque as shown in Figure 2.11(b).



(a) Steering angle



(b) Desired torque actuation



(c) Velocity of the vehicle

Fig. 2.11: Control inputs of the system

Finally, Figure 2.11(c) depicts the predefined velocity profile of the vehicle. It can be seen that the LPV control system guarantees the trajectory tracking of the vehicle at different velocities.

2.3 Estimation of the adhesion coefficient

In this section, a data-driven road surface estimation is presented. In this solution, three different road type are considered: dry, wet and icy. All types cover a predefined range of the adhesion coefficient, as shown in Table 2.4. As a first step, the acquisition of the used dataset is performed, which is detailed in Subsection 2.3.1. In the second step a selection criterion is applied, which aims to select those instances, where the current adhesion coefficient is close to its maximal value. This a crucial step since at low adhesion coefficient there is no significant difference among the measurements. Therefore, the classification algorithm may provide poor result. In the third step, C4.5 decision generation algorithm for classification is used, see Subsection 2.3.3. Finally, a simulation is presented to show the efficiency of the proposed road surface estimation algorithm. The process of the estimation is illustrated in Figure 2.12.

Tab. 2.4: Road surfaces and approximated μ values

Road surface	App. μ
dry	0.8-1.0
wet	0.5-0.8
icy	0.0-0.5

Fig. 2.12: Scheme of the algorithm

2.3.1 Data acquisition and analysis

All machine-learning-based algorithms require a lot of data to provide appropriate results. In this case, the data is provided by the high-fidelity simulation software, CarSim. Several simulations have been performed through CarSim at varying longitudinal velocity (v_x) and on different road surfaces (dry, wet, icy). In this way, more than 10 million instances have been collected. An instance consists of the following measured attributes (sensor signals):

1. Longitudinal acceleration a_x
2. Lateral acceleration a_y
3. Longitudinal velocity v_x
4. Lateral velocity v_y

5. Yaw rate $\dot{\psi}$
6. Angular velocity of wheels
   W_{ij} $(i \in \{front, rear\},\ j \in \{left, right\}$
7. Steering angle of front wheels
   δ_i, $(i \in \{left, right\})$
8. Angle of steering wheel δ_s
9. Side-slip angle of the vehicleβ
10. Slip angles of the wheels
    α_{ij} $(i \in \{front, rear\},\ j \in \{left, right\}$
11. Torques of the wheels
    M_{ij} $(i \in \{front, rear\},\ j \in \{left, right\}$
12. Roll rate $\dot{\varphi}$

2.3.2 Selection of the acquired dataset

Since not all variables are necessary for estimating the road surface, the required ones will be selected by the machine learning algorithm. The used decision tree algorithm is inherently able to select the most relevant attribute in each iteration step, more details are given in the next section. Moreover, not all instances can be used for estimating the road surface since the estimation is solely accurate, if the adhesion coefficient is close to its peak value. This indicates that the vehicle must be excited (steered, accelerated) enough for the accurate estimation. The selection of the appropriate instances is not a trivial problem. In the literature, several criteria can been found, e.g. in [63] yaw-rate, steering speed, lateral acceleration, longitudinal speed with some restrictions are used to choose the instances, in which the friction coefficient is close to its peak value. In this case, a new criterion is used, which is based on a stability condition, which was presented by [FNG18a]. The original stability condition can be written as:

$$-\varepsilon < \frac{|1 + \alpha_f|}{|1 + \delta - \beta - \frac{l_1 \dot{\psi}}{v_x}|} - 1 \leq \varepsilon, \tag{2.48}$$

where ε is an experimentally defined parameter, l_1 is the distance between the CG of the vehicle and the front axle and α_f is the mean value of the slips of the front wheels. The basic idea of this criterion relies on deviation of the dynamical behaviors of the real vehicle and the linearized model. This deviation is large when the vehicle has high values of speed and steering angle, which indicates that the vehicle gets

close to the peak value of μ. Therefore, after some changes, this condition can be used as a selection criterion for the current case:

$$\varepsilon < \frac{|1+\alpha_f|}{|1+\delta-\beta-\frac{l_1\dot{\psi}}{v_x}|} - 1 \text{ or } \frac{|1+\alpha_f|}{|1+\delta-\beta-\frac{l_1\dot{\psi}}{v_x}|} - 1 < -\varepsilon, \tag{2.49}$$

In this case, the parameter ε should be as small as possible in order to not exclude too many instances.

2.3.3 Classification algorithm for road surface estimation

C4.5 is a widely used machine learning algorithm, which generates decision trees for the classification of large amounts of data. The original algorithm was developed in 1960 by [64]. Over the past decades, the original method has been significantly improved, see e.g. [65, 66]. In the following the basic concept of C4.5 method is presented.

The initial step of the algorithm is the collection of data from varying instances. In general, an instance has several types of values called attributes $\mathcal{A} = \mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_k$. An attribute can be an independent variable or a dependent variable called class. The values of an independent variable can be continuous (numeric) or discrete (nominal). A dependent, class variable $\mathcal{C}$ is always discrete with a predefined set of values $\mathcal{C} = \mathcal{C}_1, \mathcal{C}_2...\mathcal{C}_m$ with m members. The collected data are divided into two parts:

1. a training set, which is used for teaching the algorithm,
2. a test set, which is used for evaluating the results.

The aim of the algorithm is to create a function ($\mathbb{F}$) based on the training set which is able to classify the instances by the selected class

$$\mathbb{F}(\mathcal{A}_1, \mathcal{A}_2...\mathcal{A}_k) \rightarrow \mathcal{C} \tag{2.50}$$

The created function is ordered into a tree structure, as illustrated in an example, see Figure 2.13. A tree consists of nodes and leaves. A node is associated with an attribute and a condition, and has at least two outcomes, which depend on the current value of the attribute. A leaf determines the value of the class for the current instance. The size of the resulting tree is a crucial part of the algorithm, since a large and complex tree makes it difficult to understand and use the results. Thus, C4.5 algorithm uses the greedy search method to produce the decision tree. Moreover, C4.5 algorithm considers the information gain and gain ratio criteria in the generation of the decision tree.

In the method, the information content $\mathbb{I}(S)$ of a training set is determined as

$$\mathbb{I}(\mathcal{S}) = -\sum_{j=1}^{m} \mathbb{RF} = (\mathcal{C}_j, \mathcal{S}) log(\mathbb{RF}((\mathcal{C}_j, \mathcal{S})), \tag{2.51}$$

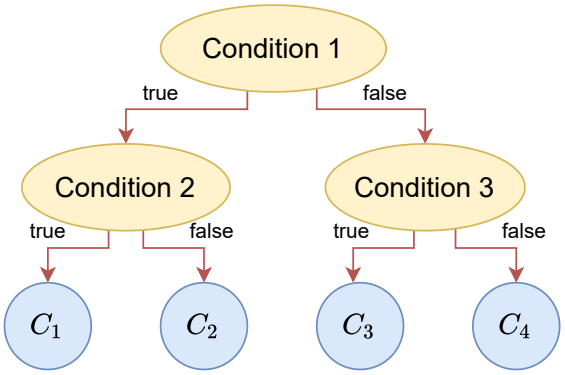


Fig. 2.13: Decision tree

where $\mathcal{S}$ is a training set that belongs to $\mathcal{C}_j$ and $\mathbb{RF}(\mathcal{C}_j, \mathcal{S})$ denotes the relative frequency of the instances. Let $\mathcal{B}$ be a test that divides $\mathcal{S}$ into subsets $\mathcal{S}_1, \mathcal{S}_2..\mathcal{S}_t$. Then the information gain $G(\mathcal{S}, B)$ can be calculated in the following form:

$$\mathbb{G}(\mathcal{S}, \mathcal{B}) = \mathbb{I}(\mathcal{S}) - \sum_{i=1}^{t} \frac{|\mathcal{S}_i|}{|\mathcal{S}|} \mathbb{I}(\mathcal{S}_i) \tag{2.52}$$

The purpose of the gain criterion is to select the best test $\mathcal{B}$ that maximizes $\mathbb{G}(\mathcal{S}, \mathcal{B})$. However, this criterion may cause problems. The reason is that the maximization of $\mathbb{G}(\mathcal{S}, \mathcal{B})$ leads to a large number of outcomes in test $\mathcal{B}$. This can be avoided by taking into consideration the potential information $\mathbb{P}(\mathcal{S}, \mathcal{B})$, such as

$$\mathbb{P}(\mathcal{S}, \mathcal{B}) = -\sum_{i=1}^{t} \frac{|\mathcal{S}_i|}{|\mathcal{S}|} log \frac{|\mathcal{S}_i|}{|\mathcal{S}|} \tag{2.53}$$

The ratio of $\mathbb{G}(\mathcal{S}, \mathcal{B})$ and $\mathbb{P}(\mathcal{S}, \mathcal{B})$ must be maximized by a test $\mathcal{B}$:

$$max\left(\frac{\mathbb{G}(\mathcal{S}, \mathcal{B})}{\mathbb{P}(\mathcal{S}, \mathcal{B})}\right) \tag{2.54}$$

Finally, C4.5 algorithm builds up the appropriate decision tree using the optimized test $\mathcal{B}$. Further details about the generation of the decision tree are found e.g. in [65].

2.3.4 Generation of the decision tree

The data collection and the evaluation through (2.49) results in data set S, which is used for the generation of the decision tree T. The inputs of the decision tree are the measured attributes of the vehicle, as presented in the previous section, while the output is the estimated type of the road surface. In this case, the generation is based on the C4.5 machine learning algorithm presented in 2.3.3.The result of the optimization is the decision tree T.

Prepocess The purpose of this step is to prepare the data for classification. This step consists of two subtasks. First task is the averaging of the values of the measured attributes over a period of time T.

$$\hat{A}_{i,t} = \sum_{n=t-T}^{t} \frac{A_{i,n}}{T} \tag{2.55}$$

where $A_{i,t}$ is the t^{th} instance of i^{th} attribute. In this way, the noises of the measurements can be reduced, and whereby better classification results can be achieved. Second task is the selection of the instances by the presented criterion 2.49.

Classification As mentioned, the goal of the classification is to create a model that is able to determine the road surface using only the measured attributes. Moreover, the presented decision tree algorithm C4.5 is used for creating the classification model. Several models have been created with different parameters, such as minimal number of instances per leaf and the selection parameter ε, the results can be found in the tables below. Table 2.5 consists of three columns, such as: minimal objects

Tab. 2.5: Relationship between the tree size and the object number

Min. Objects Num.	Corr. Class. Inst	Size
2	99.6172%	153
10	99.1009%	115
50	96.0477%	77
100	93.4307%	61
200	87.9740%	35
500	82.6954%	21

number per leaf, correctly classified instances and size of the tree. The table shows the impact of the minimal number of instances per leaf (MIL). It can be seen that as the value of MIL decreases, the size of the tree becomes greater and moreover, the percentage of the correctly classified instances gets closer to 100. Although the bigger tree provides better result, it loses its generality and fits only the test data well. Therefore, a reasonable balance should be found. In the followings, the tree with 100 MIL is used, since it still provides good classification results, while its size is small enough. Table 2.6 shows the effect of parameter ε on the resulted trees at a fixed MIL value, 100. It can be seen that if the parameter ε is too small ($< 1\%$), then the produced tree has a low classification ability. Since low ε value indicates weak selection criterion (2.49), the training set contains a lot of instances, where the adhesion coefficient was not close to its peak value. Furthermore, over 2%, the percentage of the correctly classified instances starts to decline, since the vast majority of the instances in the training set is strongly unstable, which results in the chaotic motion of the vehicle. Overall, it can be concluded that the best result is given by $\varepsilon = 2\%$.

Tab. 2.6: Relationship between the tree size and the object number

Parameter ε	Corr. Class. Inst.	Size
0.1%	58.5124%	217
1 %	88.0485%	97
2 %	93.4307%	61
5 %	92.92611%	49
10 %	90.7918%	39

The confusion matrix which belongs to this decision tree is illustrated in Table 2.7. It can be seen that the percentage of highest misclassification is 4.4%, where the 'icy' surface is classified as 'wet' by the decision tree. Table 2.7 also illustrates that there is no misclassification between the categories 'icy' and 'dry'. It means that the value of the misclassification is limited to one category.

Tab. 2.7: Confusion matrix

$\hat{dry}$	$\hat{wet}$	$\hat{icy}$	-
35.6%	0.7%	0%	dry
3.6%	29%	3.4%	wet
0%	4.4%	23.3%	icy

Resulted model The resulted model can be used in a reconfiguration control strategy. The model calculates the possible road surface using the measured data of an individual vehicle and using this information, the reconfiguration system computes the best strategy to safely control the vehicle. The development of the reconfiguration strategy and the control system is a future challenge for the authors.

2.3.5 An illustrative example of the proposed method

In the rest of this section, a simulation example of the road surface estimation algorithm is presented. During the simulation a D-class passenger car is driven along a section of Melbourne formula circuit. The track is illustrated on Figure 2.14. The track is divided into two sections, where the surface of the road is different, such as dry and icy. The dry part is depicted with a green dashed line, while the icy segment is illustrated with a red dashed line.

Furthermore, the longitudinal velocity of the vehicle is shown in Figure 2.15 (a). Basically, the speed profile consists of the different values. The first one ($\approx 100km/h$) belongs to the dry part, whilst the second one ($\approx 60km/h$) belongs to the icy part.

The classified estimated and reference road surfaces can be found in Figure 2.15 (b). It can be seen that during the simulation, the estimation algorithm frequently

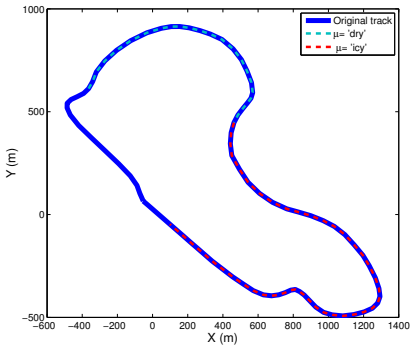
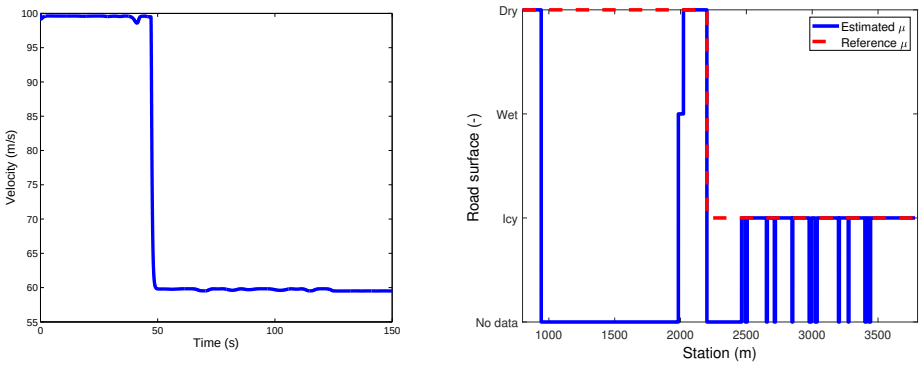


Fig. 2.14: Melbourne track



(a) Longitudinal velocity (b) Road surface estimation

Fig. 2.15: Longitudinal velocity and road surface

yields 'No data' category. The reason of it is that the vehicle does not reach the peak value of μ, since the current instance does not satisfy the inequality (2.49). Therefore, the estimation cannot be executed. Apart from the 'no data' sections, the estimation is accurate on dry and icy surfaces as well.

Thesis 1 I have developed new data-driven estimation methods for vehicle control applications using different machine-learning-based algorithms. I have shown that the proposed algorithms can be used for estimating the tyre pressure and for approximating the adhesion coefficient between the road and the tyre. For the estimation of the adhesion coefficient, I have developed a method based on the C4.5 decision tree algorithm, while the estimation of the tyre pressure has been based on the pace regression and the neural network approaches. Furthermore, I have also designed a Linear Parameter-Varying (LPV) based trajectory tracking controller, which can incorporate in the result of the tyre pressure estimation algorithm in order to increase the performances and to guarantee the stable motion of the vehicle.

Related publications: [FNG20a, FNG21c, HFNG20, FNGA19, FNGS19, FNG19a, FNG$^+$19e, FNG19c, FNG19b]

3. APPROXIMATION OF THE LATERAL STABILITY REGIONS AND ITS APPLICATION METHODS FOR VEHICLE CONTROL

In this chapter, a method for computing the lateral stability regions of passenger vehicles using data-driven approaches is presented. The motivation of stability analysis is to provide safe motion for automated vehicles through an appropriate control system. Therefore, a stability criterion for analysis and synthesis purposes is introduced, which can be used to separate the stable and unstable instances from the recorded dataset. Then, C4.5 decision tree algorithm is applied to create a decision-tree-based classification model, which is able to categorize the measurements during the operation of the vehicle. This decision tree-based model uses only the onboard signals, which are available on a conventional intelligent passenger car.

Furthermore, the proposed classification model is used in a lateral control design, which applies also the road surface estimation algorithm presented in the previous section. The lateral control design is extended with a longitudinal velocity optimization process, which computes the maximal longitudinal velocity of the vehicle for a predefined prediction horizon. In this way, the stability of the vehicle can be guaranteed, while the traveling time is minimized. This chapter is organized as follows:

- Firstly, a novel method is presented for approximating the lateral stability sets of the vehicles using a data-driven approach.
- Secondly, a LPV-based lateral control design is proposed, in which the adhesion coefficient is handled as a scheduling parameter provided by the estimation algorithm presented in Subsection 2.3.
- Thirdly, a longitudinal velocity optimization algorithm is presented, which can calculate the optimal velocity profile for the vehicle using the data-driven stability sets, presented in this chapter.
- Finally, a simulation example is given to illustrate the operation and the efficiency of the proposed control algorithm through a comprehensive scenario.

3.1 Approximation of lateral stability regions for passenger cars

Stable motion is the most important requirement against vehicles with autonomous control functionality. Nevertheless, in vehicle control context, considering

nonlinear effects of the vehicle, the definition of stability can be difficult. Unstable motion of the vehicle by various factors can be caused, e.g., sudden change in the adhesion coefficient, insufficient lateral force at large lateral slip. These factors must be taken into account during the control design in order to guarantee the motion stability of the vehicle even in risky traffic situations. From control engineering aspect, the criterion of stable vehicle motion is in relation with the performance of the vehicle on the motion, as it is illustrated by the following examples.

- It can difficult to determine the unstable motion of the vehicle. Due to nonlinearities in the lateral dynamics, the stability regions are generally computed on constant longitudinal velocities, see e.g. [67, 68]. Nevertheless, sometimes the unstable motion of the vehicle leads to a significant reduction in the velocity, at which the stability of the vehicle is restored. Thus, if the stability of the vehicle is evaluated depending on the velocity, the scenario is only locally unstable, but globally stable.

- Moreover, the increasing error in the path tracking of the autonomous vehicle is caused by deficiencies in the lateral control. In this case the error can be handled as a performance problem. Performance level depend also on the designed controller, not only on the vehicle dynamics itself.

These examples illustrate that it may be difficult to find an appropriate criterion to evaluate stability in the context of vehicle dynamics with nonlinearities. The stability of motion is described by the original work of Lyapunov, see [69]. It motivates that stability is recommended to interpret as a character of the motion. Thus, in this section the motion of the vehicle based on collected data is analyzed.

The viewpoint of analyzing vehicle motion with data-driven tools yields a decision tree based on collected data. The data is provided by the sensors of the autonomous vehicles, such as inertial and gyro sensors, GPS velocity measurement and wheel speed sensors. Moreover, the data-driven analysis contains scenarios which are considered to be acceptable or unacceptable from the viewpoint of the path tracking of the autonomous vehicle. These scenarios are called good or bad instances in the dataset. The purpose of the decision tree generation is to find the set of relations with which the a current scenario can be classified as acceptable or unacceptable. Each scenario requires a definition of a criterion to be acceptable.

The approach of this thesis is based on the idea that the motion of the vehicle is generally acceptable for the human passengers in the linear region of the tyre force characteristics. In this case the side-slip angle of the axles can provide information about the characteristics of the motion. Thus, the defined criterion expresses the similarity between the current side-slip of the front axle $(1 + \alpha_1)$ and the expected side-slip based on the linear formulation of the vehicle [FNG18a]. The selection criterion is similar to that, which has been used to road surface estimation (2.49). This criterion scales the nonlinear behavior of vehicle by comparing the current

measurement to a linear model:

$$-\varepsilon_1 < \frac{|1+\alpha_f|}{|1+\delta-\beta-\frac{l_f\dot{\psi}}{v_x}|} - 1 \leq \varepsilon_1, \tag{3.1}$$

where ε_1 is an experimentally defined parameter, l_1 is the distance between the center of gravity of the vehicle and the front axle. $\dot{\psi}$ denotes the yaw rate, β is the vehicle side-slip v_x is longitudinal velocity and δ is the steering angle. This condition expresses that an instance is said to be 'acceptable', if (3.1) is fulfilled. It approximates the stable states of the vehicle. Otherwise, the instance is classified as 'unacceptable', which is related to the unstable state of the vehicle. Thus, condition (3.1) results in the dataset S_1 with the classification of 'acceptable' or 'unacceptable' for the stability set approximation.

In the followings, a machine learning-based algorithm is proposed for approximating the lateral stability regions of the vehicle. This method is a pure data-driven approach, which means it uses only the available measured signals from the onboard system. The main advantage of this approach is that it is more suitable for the nonlinear dynamics of the vehicle, which appears at high velocities.

In the following a brief description of the results of the stability set approximation is presented using the previously introduced C4.5 method, see Subsection 2.3.3. The approximation is based on the assumption that criterion (3.1) is able to distinguish the 'acceptable' and 'unacceptable' states, which provides information about the stability of the vehicle. However, the defined condition is not identical with a vehicle stability criterion, as presented above. Although the class of 'acceptable' instances contains only stable states, it can provide a conservative inner approximation of the stability set. The details of the set computation are found in [FNG18a].

Illustration of the analysis results

The purpose of this section is to demonstrate the reachability sets of the lateral vehicle model, which are computed through the machine learning algorithm. In the analysis the data-mining WEKA software is used, in which C4.5 algorithm has been implemented [70].

The attributes of the instances are α_f, α_r slip at the front and rear wheels, β side slip of the vehicle, $\dot{\psi}$ yaw-rate, v_x longitudinal velocity, μ adhesion coefficient and C class. The C class has two values, i.e., *acceptable* and *unacceptable*, and the instances are classified by the algorithm. During the analysis the training set contains approximately 1.2 million instances, while the test set for the validation has 2 million members. In the example a mid-size passenger car is used.

The generated trees are evaluated by the cross-validation technique, the results can be found in Table 3.1. The first column in Table 3.1 shows the minimum number of instances which are contained in a leaf. The second column illustrates the percentage of the correctly classified instances. The sizes of the produced trees are in the last column. Note that the increasing number of the minimum objects

decreases both the percentage of the correctly classified instances and the sizes of the trees.

Tab. 3.1: Relationship between the tree size and the object number

Min. Objects	Correctly Classified Inst.	Size of Tree
2	99.7343%	2431
10	99.6426%	1339
100	99.2948%	315
500	98.9136%	97
1000	98.7037%	61
5000	98.1892%	17

In the following, the classification with minimum 500 objects is used, because this object number has a reasonable percentage of correctly classified instances. Moreover, the generated tree is sufficiently small to be used for further analyses. In Figure 3.1 the instances in the test set which are classified as 'good' are illustrated. Note that the results of the decision tree appropriately cover the test set.

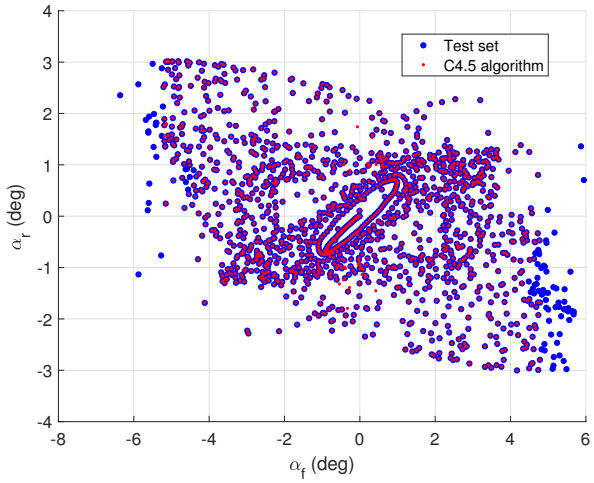


Fig. 3.1: Results of the decision tree

Figure 3.2 shows the results of the decision tree and the classified test sets in the plane of α_f and α_r at different velocities. The two slip attributes have high impacts on the resulting decision tree, which shows that the calculated sets fit well. Note that the sizes of the sets become larger with increasing velocity. It means that the vehicle can reach larger regions of slips at high velocities. This tendency is confirmed by the experience in vehicle dynamics.

Figure 3.3 illustrates the results of the classification and the test sets in the plane of yaw rate and side slip at different velocities, in which the sets also fit well. These attributes have lower impacts on the logic relations in the decision tree. The regions

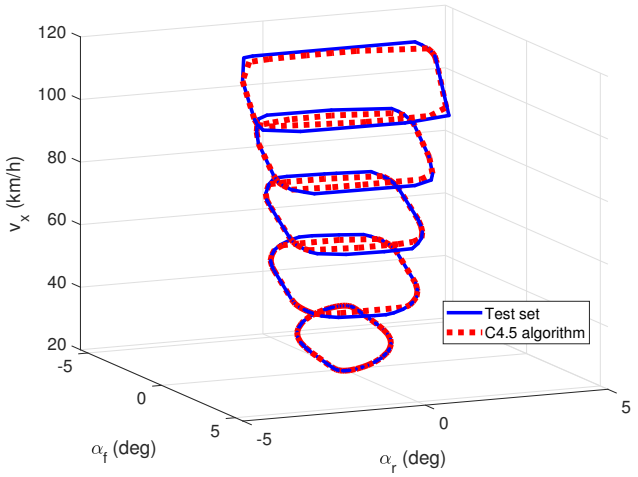


Fig. 3.2: α_f and α_r sets depending on velocity v_x

of reachable β and $\dot{\psi}$ increase depending on the longitudinal velocity, similarly to the tendency at the lateral slips α_f, α_r, see Figure 3.2.

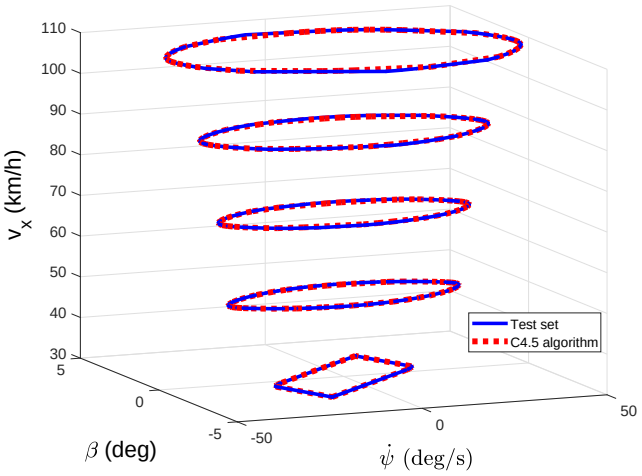


Fig. 3.3: $\dot{\psi}$ and β sets depending on velocity v_x

In the following, the impact of the adhesion coefficient on the reachability regions of the vehicle model is illustrated. Figure 3.4 shows the regions of the slips at different adhesion coefficients, in which the velocity of the vehicle is fixed at $v_x = 90$ km/h. Note that the illustrated sets become smaller at high adhesion coefficients. The reason for this behavior is that the adhesion coefficient highly influences the lateral forces of the wheels. At high μ the small slip angle generates high lateral force, while at small μ the higher slip angle induces high lateral force.

Figure 3.5 shows the sets of $\dot{\psi}$ and β at different adhesion coefficients and at the

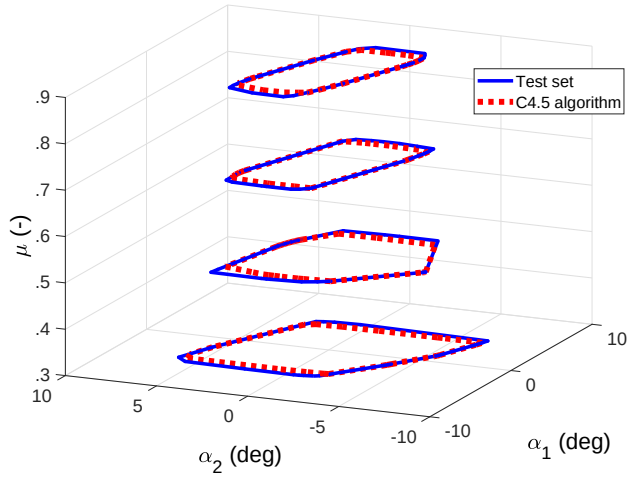


Fig. 3.4: α_f and α_r sets depending on the adhesion coefficient

fixed velocity of 90km/h. The tendency of the set sizes is similar to the previous case. The size of the sets becomes smaller at high adhesion coefficients and larger at low adhesion coefficients. The calculated sets fit to the test sets well.

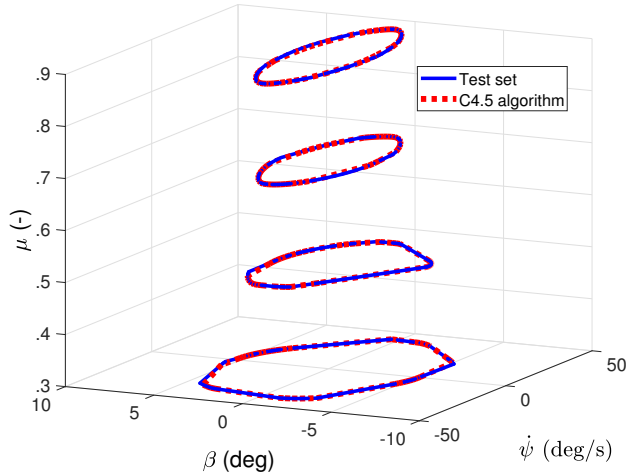


Fig. 3.5: $\dot{\psi}$ and β sets depending on the adhesion coefficient

3.2 LPV-based vehicle control design and velocity selection strategy

The goal of this section is to provide a control strategy in which the design of steering actuation and velocity selection are coordinated and the safe motion of the autonomous vehicle is guaranteed. First, the steering control design based on the LPV method is presented. It uses the result of the μ estimation through a scheduling

variable. Second, the design of the velocity profile based on the steering control and on the result of the data-driven stability set analysis is proposed.

3.2.1 Formulation of control-oriented LPV model with uncertainties

The lateral dynamics of the vehicle is described by the two-wheeled single-track, which has been presented in Subsection 2.2.

Since the original model does not take into account the effect of the different road surfaces, which results in the change of the adhesion coefficient, the computation of the lateral force has been modified in the following way: $F_y = C(\hat{\mu})\alpha$. C is the linearized cornering stiffness, which depends on the estimation of the adhesion coefficient $\hat{\mu}$ linearly: $C(\hat{\mu}) = \hat{\mu}C_n$. Note that the impact of longitudinal slip is assumed to be small comparing to the impact of lateral slip (α) on the lateral force. It can be guaranteed by the longitudinal controller, which aims to minimize the longitudinal acceleration of the vehicle. C_n denotes the nominal cornering stiffness.

$\hat{\mu}$ consists of the following two components:

- μ_n is the result of the estimation, which is numerically the mean value of the current category (dry,wet or icy) as presented in the previous chapter, see Table 2.4.
- Since the result of the estimation is a category, which covers a specific range of μ, the actual μ can be elsewhere within that range. Therefore, an error value is introduced, which aims to describe the largest possible difference between the actual μ and the estimated one.

Finally, the $\hat{\mu}$ is computed as $\hat{\mu} = \mu_n + \mu_e$. Therefore, the relations in (2.11) can be rewritten in the following forms:

$$mv_x(\dot{\psi} + \dot{\beta}) = \mu_n(C_n\alpha_f + C_n\alpha_r) + \mu_e(C_n\alpha_f + C_n\alpha_r), \tag{3.2a}$$

$$J\ddot{\psi} = \mu_n(C_n\alpha_f l_1 - C_n\alpha_r l_2) + \mu_e(C_n\alpha_f l_1 - C_n\alpha_r l_2), \tag{3.2b}$$

$$\dot{v}_y = v_x(\dot{\psi} + \dot{\beta}). \tag{3.2c}$$

The representation of (3.2) is transformed into the parameter-varying state-space model

$$\dot{x} = A(\rho_1, \rho_2)x + B_2(\rho_1, \rho_2)u + \frac{\mu_e}{\mu_n}\Big(A(\rho_1, \rho_2)x + B_2(\rho_1, \rho_2)u\Big), \tag{3.3}$$

where the state vector is $x = \begin{bmatrix}\dot{\psi} & \beta & v_y & y\end{bmatrix}^T$, the control input is $u = \delta$ and $\rho_1 = v_x$, $\rho_2 = \mu_n$ are the selected scheduling variables of the system. $A(\rho_1, \rho_2)$, $B_2(\rho_1, \rho_2)$ are matrices, which include the nominal adhesion coefficient μ_n in a linear form.

In (3.3) the value of μ_e is unknown, and thus, the expression $\frac{\mu_e}{\mu_n}\Big(A(\rho_1, \rho_2)x + B_2(\rho_1, \rho_2)u\Big)$ is handled as a disturbance of the system. The goal of the control-oriented formulation is to find a description of the system (3.3) which is valid for

worst-case scenarios. Therefore, the disturbance is bounded through the following inequality, which is taken element-wisely:

$$B_1(\rho_1,\rho_2)w \geq \frac{\mu_e}{\mu_n}\Big(A(\rho_1,\rho_2)x + B_2(\rho_1,\rho_2)u\Big), \tag{3.4}$$

which results in the state-space representation:

$$\dot{x} = A(\rho_1,\rho_2)x + B_2(\rho_1,\rho_2)u + B_1(\rho_1,\rho_2)w. \tag{3.5}$$

In (3.5) w is a norm-bounded noise, while $B_1(\rho_1,\rho_2)$ is a parameter-varying coefficient matrix, whose selection is detailed below.

The aim of the selection of $B_1(\rho_1,\rho_2)$ is to find the lowest upper bound of $\Big(A(\rho_1,\rho_2)x + B_2(\rho_1,\rho_2)u\Big)\mu_e/\mu_n$. Since it contains several components, the determination of the bound is based on various assumptions.

- Since matrices $A(\rho_1,\rho_2)$ and $B(\rho_1,\rho_2)$ are given and fixed for each grid point, only the upper bounds of the state-vector x and the input signal u must be determined.
- Inequality (6) should be taken elementwisely. for all states.
- During the determination of $B_1(\rho_1,\rho_2)$, the value of the uncertainty μ_e is considered with its maximum bound, such as $\mu_{e,max} \equiv max(\mu_e)$ in (3.5).
- The determination of the maximum of the state vector x is more challenging task. In this paper, the maximum values for each states are determined by using the presented stability sets (see Section 3.1). In this manner, the possible worst cases of the states are computed for each grid points based on the stability sets, using the edges of the sets. Finally, the supremum of the maximum values regarding to all grid points is selected as the worst case scenario. As a result, the maximum stability set must be selected, depending on the adhesion coefficient $\mu_n \pm \mu_{e,max} = \rho_2 \pm \mu_{e,max}$, such as

$$\max\{x\} = \max\Big(\mathcal{R}(\rho_1,\rho_2-\mu_{e,max}),\mathcal{R}(\rho_1,\rho_2+\mu_{e,max})\Big). \tag{3.6}$$

- The maximum value of the control signal u_{max} is also used instead of u, which is a predefined value of the steering system, depending on the physical limits as $u_{max} \equiv max(u)$.

Finally, the matrix $B_1(\rho_1,\rho_2)$ can be calculated as:

$$B_1(\rho_1,\rho_2) = \frac{\mu_{e,max}}{\rho_2}\Big(A(\rho_1,\rho_2)\max\{x\} + B_2(\rho_1,\rho_2)u_{max}\Big), \tag{3.7}$$

where $\max\{x\}$ is computed through (3.6).

Design of robust lateral LPV control

The goal of the control design is to guarantee the required motion of the vehicle with minimum steering control intervention. Thus, the following performances are specified.

- *The minimization of lateral error.* The designed control must reduce the error between the lateral position of the vehicle y and the reference path y_{ref}:

$$z_1 = y_{ref} - y, \qquad |z_1| \rightarrow min. \tag{3.8}$$

- *The minimization of yaw-rate error.* The improvement of path tracking requires the consideration of the turning motion of the vehicle through the yaw-rate, such as

$$z_2 = \dot{\psi}_{ref} - \dot{\psi}, \qquad |z_2| \rightarrow min, \tag{3.9}$$

  where $\dot{\psi}_{ref}$ represents the reference yaw rate of the vehicle, which depends on the longitudinal velocity [59].

- *The minimization of the control input.* The path tracking of the vehicle must be guaranteed with minimum steering intervention, which leads to the performance

$$z_3 = \delta, \qquad |z_3| \rightarrow min. \tag{3.10}$$

The specified performances are compressed into a vector $z = \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T$, which leads to the performance equation

$$z = C_1 x + D_{11} r + D_{12} u, \tag{3.11}$$

where C_1, D_{11}, D_{12} are matrices and r contains the signals y_{ref} and $\dot{\psi}_{ref}$.

The design of the LPV control requires the system dynamics in the state-space form (3.5) and the performance equation (3.11). Moreover, it is necessary to scale the input and output signals of the plant, as it is illustrated in Figure 3.6. Furthermore, Figure 3.7 shows the architecture of the lateral control including the road-surface estimation algorithm. In practice, the scaling of the signals during the design process is performed through transfer functions [71]. The role of $W_{ref,1}, W_{ref,2}$ is to scale the signals y_{ref}, $\dot{\psi}_{ref}$. Similarly, $W_{w,1}, W_{w,2}$ scale the noises on the lateral position and on the yaw-rate measurements. Moreover, W_w reflects on the uncertainty of the μ estimation, which is treated as an external noise (w) in the control system. Noises w_y, $w_{\dot{\psi}}$ and w are incorporated in the vector $\omega = \begin{bmatrix} r & w_y & w_{\dot{\psi}} & w \end{bmatrix}^T$. The priority among the performances is guaranteed by $W_{ref,i}$, $i = \{1, 2, 3\}$. The transfer functions $W_{ref,i}$, $i = \{1, 2\}$ are selected in a second-order form to achieve the smooth tracking

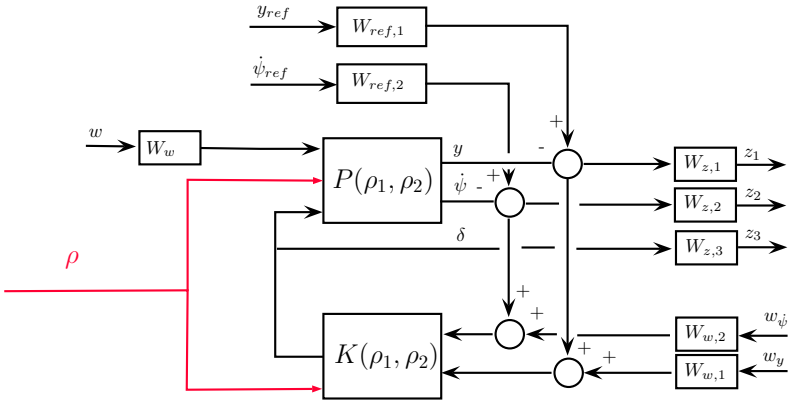


Fig. 3.6: Augmented plant for LPV control design

of the reference signals. However, in the case of $W_{ref,3}$ a first-order proportional transfer function can be sufficient to guarantee the minimization of δ.

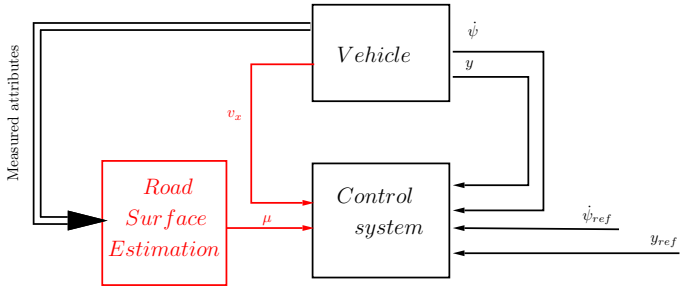


Fig. 3.7: Architecture of the lateral control system

The quadratic LPV performance problem is to choose the parameter-varying controller $K(\rho_1,\rho_2)$ in such a way that the resulting closed-loop system is quadratically stable and the induced $\mathcal{L}_2$ norm from the disturbance and the performances is less than a predefined value γ. The minimization task is the following:

$$\inf_{K(\rho_1,\rho_2)} \sup_{\rho_1,\rho_2 \in F_\rho} \sup_{\|\omega\|_2 \neq 0, \omega \in \mathcal{L}_2} \frac{\|z\|_2}{\|\omega\|_2}, \tag{3.12}$$

where F_ρ bounds the scheduling variables. The design of the controller with a single Lyapunov function is performed. The yielded controller $K(\rho_1,\rho_2)$ is formed as

$$\dot{x}_K = A_K(\rho_1,\rho_2)x_K + B_K(\rho_1,\rho_2)y_K, \tag{3.13a}$$
$$u = C_K(\rho_1,\rho_2)x_K + D_K(\rho_1,\rho_2)y_K, \tag{3.13b}$$

where x_K is the state vector of the dynamic controller, A_K, B_K, C_K, D_K are ρ_1,ρ_2 dependent matrices. y_K is the vector of the lateral error and yaw-rate error measurements, which is formed as

$$y_K = C_2 x + D_{21} r, \tag{3.14}$$

where C_2, D_{21} are matrices.

The existence of a controller that solves the quadratic LPV γ-performance problem can be expressed as the feasibility of a set of LMIs, which can be solved numerically. The constraints set by the LMIs are not finite. The infiniteness of the constraints is relieved by a finite, sufficiently fine grid. To specify the grid of the performance weights for the LPV design the scheduling variables are defined through lookup-tables, see [72, 73].

3.2.2 Design of the velocity profile using the stability regions

In the design of the velocity profile it is necessary to guarantee the safe motion of the vehicle. It means that the vehicle motion is inside the linear region of the tyre-vehicle dynamics, as it is represented by (3.5). Thus, the LPV-based vehicle model and the designed controller are suitable for the path following control problem and the system is inside its validity range.

The design of the velocity profile requires the prediction of the vehicle motion, especially the yaw rate and the side-slip angle. The prediction is based on the closed-loop model of the vehicle (3.5) and the designed LPV control (3.13). The closed-loop system is formed as

$$\dot{x}_{cl} = A_{cl}(\rho_1, \rho_2)x_{cl} + B_{cl}(\rho_1, \rho_2)r, \tag{3.15}$$

where $\dot{x}_{cl} = \begin{bmatrix}\dot{x} & \dot{x}_K\end{bmatrix}^T$ and the matrices are

$$A_{cl}(\rho_1, \rho_2) = \begin{bmatrix} A_{cl,11} & A_{cl,12} \\ A_{cl,21} & A_{cl,22} \end{bmatrix}, \tag{3.16a}$$

$$\begin{aligned} A_{cl,11} &= A(\rho_1, \rho_2) + B(\rho_1, \rho_2)D_K(\rho_1, \rho_2)C_2, \\ A_{cl,12} &= B(\rho_1, \rho_2)C_K(\rho_1, \rho_2), \\ A_{cl,21} &= B_K(\rho_1, \rho_2)C_2, \\ A_{cl,22} &= A_K(\rho_1, \rho_2), \end{aligned}$$

$$B_{cl}(\rho_1, \rho_2) = \begin{bmatrix} B(\rho_1, \rho_2)D_K(\rho_1, \rho_2)D_{21} \\ B_K(\rho_1, \rho_2)D_{21} \end{bmatrix}. \tag{3.16b}$$

For the prediction of $\dot{\psi}$ and β the closed-loop system (3.15) is rewritten as a discrete-time model using the sampling time T [74], which yields the following model

$$x_{cl}(k+1) = A_{cl}(k)x_{cl}(k) + B_{cl}(k)r(k), \tag{3.17a}$$

$$y_{cl}(k) = C_{cl}x_{cl}(k), \tag{3.17b}$$

in which the compact notation $A_{cl}(k), B_{cl}(k)$ is used instead of $A_{cl}(\rho_1(k), \rho_2(k))$ and $B_{cl}(\rho_1(k), \rho_2(k))$, respectively. Moreover, $y_{cl}(k)$ contains the yaw rate and the side-slip angle, C_{cl} is the related matrix for the selection of the states.

The prediction of $y_{cl}(k)$ is performed on the horizon n, such as

$$y_{cl}(k,n) = \begin{bmatrix} y_{cl}(k+1) \\ y_{cl}(k+2) \\ \vdots \\ y_{cl}(k+n) \end{bmatrix} = \begin{bmatrix} C_{cl}A_{cl}(k) \\ C_{cl}A_{cl}(k)A_{cl}(k+1) \\ \vdots \\ C_{cl}\prod_{i=k}^{k+n} A_{cl}(i) \end{bmatrix} x_{cl}(k) + \\ + \begin{bmatrix} C_{cl}B_{cl}(k) & \cdots & 0 \\ C_{cl}A_{cl}(k)B_{cl}(k) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ C_{cl}\prod_{i=k}^{k+n-1} A_{cl}(i)B_{cl}(k) & \cdots & C_{cl}B_{cl}(k) \end{bmatrix} \begin{bmatrix} r(k+1) \\ \vdots \\ r(k+n) \end{bmatrix} = \mathcal{A} + \mathcal{B}R, \quad (3.18)$$

where $\mathcal{A}$ contains the current states of the system with the varying system matrices, $\mathcal{B}$ is built by the state matrices and R contains the reference signals. As an assumption, during the prediction the adhesion coefficient μ is considered to be constant, thus $\rho_2(i) \equiv \rho_2(k)$, $\forall i \geq k$. Moreover, it is necessary to consider that $\dot{\psi}_{ref}(k)$ in $r(k)$ can depend on $v_x(k) = \rho_1(k)$, see (3.9). Thus, the modification of the longitudinal velocity can also result in the variation of the reference signal.

The goal of the velocity profile design is to maximize the elements of the vector $\rho = \left[\rho_1(k+1) \dots \rho_1(k+n)\right]^T$, which represent the velocity of the vehicle on the forthcoming road section. However, there are some constraints which must be guaranteed through the maximization process. First, ρ must be smaller than the upper bound of the scheduling variable ρ_{max}

$$\rho \leq \rho_{max}. \quad (3.19)$$

The elements of ρ_{max} represent the maximum velocity limit on the forthcoming road horizon.

Second, the predicted state vector $y_{cl}(k,n)$ must be inside the stability region of the system. The stability sets are approximated through the method of Section 3.1. In the constraint is necessary to guarantee that $y_{cl}(k,n)$ is inside of the stability sets, such as

$$y_{cl}(k,n) \in \mathcal{R}\big(\rho_1(i), \rho_2(k)\big), \quad \forall k \leq i \leq n. \quad (3.20)$$

Finally, the optimization problem of the longitudinal velocity profile is formed as

$$\max_{\rho_1(k+1)\dots\rho_1(k+n)} \rho \quad (3.21)$$

subject to the constraints (3.19), (3.20):

$$\rho \leq \rho_{max} \quad (3.22a)$$

$$y_{cl}(k,n) \in \mathcal{R}\big(\rho_1(i), \rho_2(k)\big), \quad \forall k \leq i \leq n. \quad (3.22b)$$

The result of the optimization is the velocity profile of the autonomous vehicle on the horizon n, which guarantees the safe motion of the vehicle. Since the presented optimization task is challenging especially in real time application, a gird search based solution is applied. It means that the optimization task is computed for a finite number of longitudinal velocities and the maximal possible velocity is selected as reference longitudinal velocity for the vehicle, e.g. in all computation step $\pm 10m/s$ range related to the actual velocity with $1m/s$ velocity grid.

3.3 Simulation examples

In this section the effectiveness of the proposed control strategy is illustrated through simulation examples. The result of the decision tree-based stability analysis, presented in Section 3.1 is applied. For the adhesion coefficient estimation, the proposed algorithm, presented in Section 2.3 is used.

The control strategy has been implemented in CarSim via Matlab/Simulink and the results of path tracking scenarios are shown. In the example the model of a mid-size D-class passenger car has been used. During the simulation the autonomous vehicle moves along a section of the Waterford Hills Racing track, along which the adhesion coefficient varies. In the first scenario the vehicle is controlled through the proposed control strategy using the results of the μ estimation algorithm and the velocity profile optimization algorithm. However, in the second scenario the autonomous vehicle moves without the information about μ variation, which means that the control and the velocity profile are not adjusted to the changed road conditions. The geometric and other parameters of the lateral bicycle model, which is used in the control design, are shown in Table 3.2.

Tab. 3.2: Parameters of the used D-class vehicle

Parameter	Notion	Value	Unit
Mass of the car	m	1690	kg
Yaw-inertia	J	4192	kgm^2
Location of front axis from COG	l_1	1.11	m
Cornering stiffness of front wheels	C_1	155160	N/rad
Location of rear axis from COG	l_2	1.66	m
Cornering stiffness of rear wheels	C_2	114659	N/rad
Front drag area of the car	A	1.8	m^2
Height of COG	h	0.56	m
Type of front suspensions	-	Independent	-
Mass of front suspensions	$m_{s,f}$	85	kg
Type of rear suspensions	-	Independent	-
Mass of rear suspensions	$m_{s,r}$	85	kg

Furthermore, during the control design, the following weighting functions are

used:

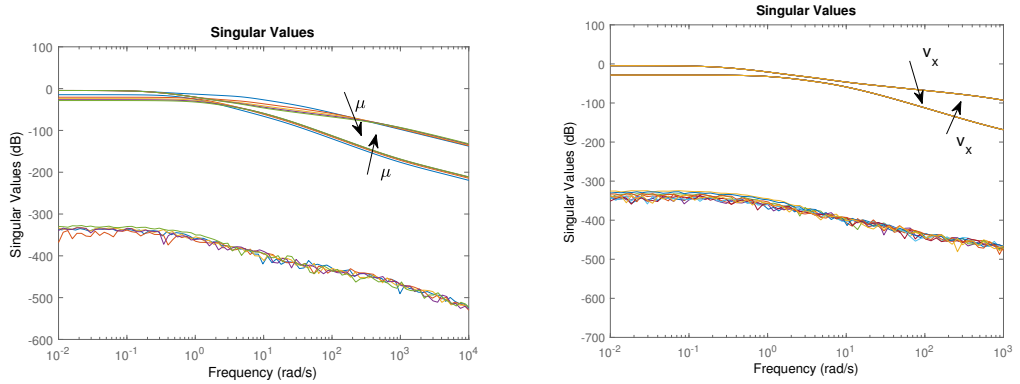
$$W_{ref,1} = 0.1 \cdot \frac{1}{s+1} \quad W_{ref,2} = 0.2 \cdot \frac{1}{s+1} \tag{3.23}$$

$$W_{z,1} = 10 \cdot \frac{1}{0.2s+1} \quad W_{z,2} = 15 \cdot \frac{1}{0.1s+1} \quad W_{z,3} = 0.2 \cdot \frac{5s+1}{10s+1} \tag{3.24}$$

$$W_{w,1} = 0.02 \cdot \frac{1s+1}{10s+1} \quad W_{w,2} = 0.05 \cdot \frac{1s+1}{10s+1} \quad W_w = 1 \tag{3.25}$$

The presented weighting functions have been tuned and chosen in such a way to ensure the predefined performances (3.8,3.9,3.10), while guaranteeing the attenuation of the noises on the measured signals. For example: $W_{z,1}$ ensures that the tracking error of lateral position is below $< 0.1m$, while $W_{z,2}$ guarantees that error of the yaw-rate tracking is less than $0.066rad/s$, Furthermore, the weighting function $W_{z,3}$ reflects on the physical limitations of the actuator.

The analysis of the LPV controller based on the singular values is found in Figure 3.8. As an illustration for the examination, Figure 3.8(a) shows that case when the longitudinal velocity of the vehicle is fixed at its highest value ($v_x = 41m/s$) and the adhesion coefficient varies between the range $[0.2 \dots 1]$. As the figure indicates, the sensitivity functions are below $0dB$ amplitude along the whole frequency range. It means that the designed LPV controller is able to attenuate the effects of the noises and uncertainty on the predefined performances. Figure 3.8(b) shows that case when the adhesion coefficient is fixed at its highest value ($\mu = 1$), while the longitudinal velocity increases from its lowest value ($v_x = 8m/s$) to its highest one ($v_x = 41m/s$). As the figure indicates that gamma plots vary in a narrower range comparing to the previous cases. Since the singular values still remain below the $0dB$, the presented control system is able to resist to noises regardless the variation of the longitudinal velocity.



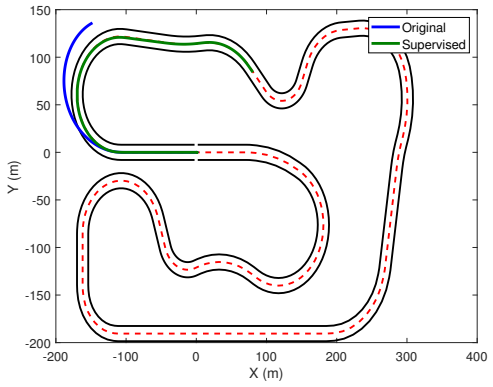
(a) Singular value plots for high $v_x = 41m/s$ (b) Singular value plots for high $\mu = 1$

Fig. 3.8: Illustrations of singular values at frozen scheduling parameters

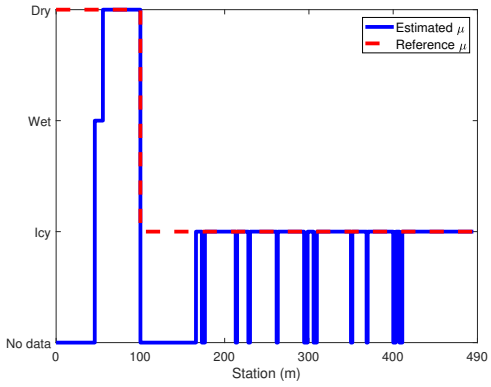
The track and the paths of the vehicle in both scenarios are illustrated in Figure 3.9(a). It can be seen that the vehicle in the second scenario is not able to follow the

path accurately without the proposed algorithms. The reasons for this difference is detailed below.

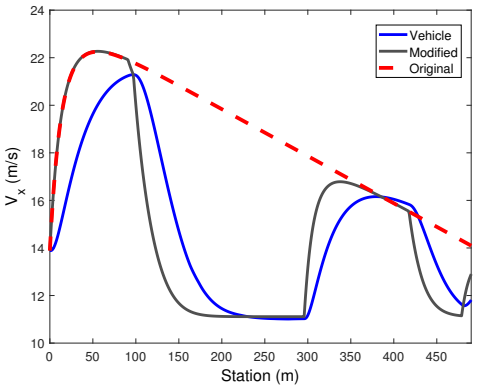
In the simulation the track has two sections depending on μ. In the first section $(0 - 100m)$ the adhesion coefficient of the road falls into the category 'dry' $(0.8 < \mu < 1.0)$, while the second section $(100m-)$ is categorized as 'icy' $(0.0 < \mu < 0.5)$. The result of the μ estimation algorithm is illustrated in Figure 3.9(b). Apart from two short sections, the estimator calculates the type of the road surface accurately. The reason for the lack of estimation is that the vehicle does not reach the peak value of α_f and therefore condition (2.49) is violated.



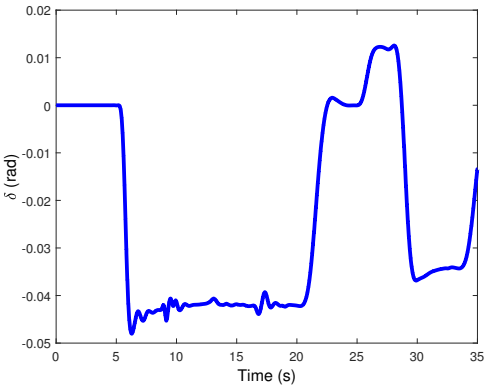
(a) Path of the vehicles on Waterford Hills track



(b) Results of μ estimation



(c) Velocity profiles



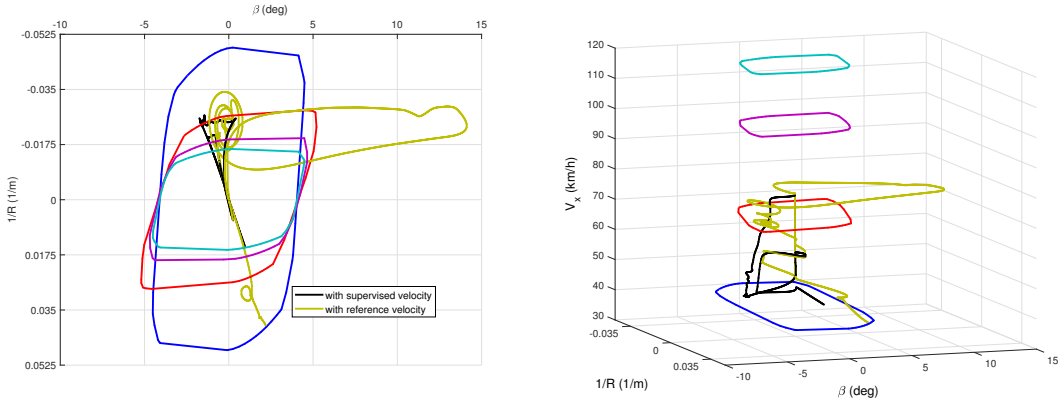
(d) Steering angle actuation

Fig. 3.9: Results of the simulation

The velocity profiles in both scenarios are shown in Figure 3.9(c). The dashed red line represents the predefined, original velocity (second scenario), while the optimized velocity profile is illustrated as a solid grey line. The real velocity profile of the vehicle is depicted by the solid blue line. The optimization algorithm modifies the original velocity profile twice during the simulation (first scenario). Both modifications are caused by bends, in which the vehicle cannot follow the path with the predefined velocity at low μ. Thus, the algorithm computes the optimal velocity

using the stability sets. The setting time of the longitudinal velocity is relatively high (around $1.5s$ to increase the velocity with $10km/h$), which is caused by the requirements against the longitudinal controller. These requirements are the minimization of longitudinal acceleration and jerk in order to improve the riding comfort of the vehicle, and thus, a slower control intervention is preferred.

The stability sets at $\mu = 0.4$ ('icy') with different v_x and the $\kappa - \beta$ state-space trajectory of the vehicle in the first scenario can be seen in Figure 3.10. As the figures show, the state trajectory of the vehicle which is controlled by the proposed algorithm is inside the stability sets during the entire simulation. In contrast the vehicle which uses the reference velocity profile leaves the areas of the stability sets. As a result, the vehicle becomes unstable, and therefore it cannot follow the predefined track.



(a) Sets with respect to the velocity

(b) Sets in the plane of $1/R$ and β

Fig. 3.10: Stability sets and the trajectory of the vehicle

Finally, in Figure 3.9(d) the steering control input of the vehicle in the first scenario is shown. It is computed by the proposed LPV control. It can be seen that the steering angle is between $-0.05 \ldots 0.02$ rad, which is an acceptable range for the presented velocity profile.

Thesis 2 I have developed a method for approximating lateral stability region of passenger vehicles. The separation of the stable and unstable regions are based on a novel stability criterion using the two-wheeled lateral vehicle model. I have used the separated instances to build a classification model, which can be used to determine the stability of the vehicle during its operation using a decision tree-based algorithm. Furthermore, I have developed a Linear Parameter-Varying (LPV)-based lateral controller, which can handle the effect of the adhesion coefficient and the uncertainty of the estimation. I have also shown that the classification model can be used for a longitudinal velocity optimization algorithm, which can guarantee the stable motion of the vehicle by adjusting its velocity to the forthcoming road segment.

Related publications: [FNG20a, FNG19d, FNG19a, FNG19c, FNG19b, FNG18a, FNG21c, FNG21e]

4. LPV-BASED PARAMETER OPTIMIZATION PROCESS USING DATA-DRIVEN APPROACH

In the recent decades the LPV-based control solutions have become a widely used approach to control nonlinear systems including vehicle-oriented problems. The main advantage of this method is that nonlinearities, from e.g. the parameter variation of the system, can be handled as a scheduling parameter, by which a wide range of the system dynamics can be more accurately described. Considering a more accurate model, the performances of the closed-loop system can be improved by adjusting the controller to the actual dynamic of the system.

However, some problems can pop up during the modeling phase of the system. For example, some varying parameters of the system cannot be measured directly and their estimation also can arise difficulties. Considering a vehicle-oriented example, inertia or the cornering stiffness of the vehicle can vary significantly during the operation of the car. It can result in the degradation of the performances of the closed-loop.

In this chapter, two methods are presented to determine the appropriate scheduling parameters of the system using different machine learning algorithms. In the first case the pace-regression algorithm is applied, which approximates the parameter in a linear form. Although this linear approximation can be beneficial in several applications, it may have low approximation capability in the case of highly nonlinear system. The second method uses a decision tree machine learning algorithm. This approach is capable of coping with highly nonlinear behavior. However, its results may be difficult to interpret and use in some cases.

The efficiency o proposed methods is demonstrated through a vehicle oriented example.

4.1 Pace regression-based parameter optimization method

In this section, the data acquisition and the LPV-based data-driven model parameter tuning process are presented. Firstly, the collection of the dataset with numerous variables is detailed, which is provided by the vehicle dynamics simulation software, CarSim. Secondly, the prepocess of the collected data is explained, which includes the selection and the scaling of the data. Finally, the data-driven LPV-based model parameter tuning process is presented, in which the scheduling parameters and the model parameters are selected through a machine-learning-based pace regression algorithm.

Acquisition of data from simulations

Similarly to the previous method, the first step of the algorithm is the data acquisition. During the data acquisition, the longitudinal velocity varied between $10 - 20m/s$ and the car were driven on tracks with different circuits. During the simulations the following signals have been measured and collected:

1. longitudinal velocity (v_x)
2. angular velocity of the wheels ($\omega_{x,y}$), $x \in \{front, rear\}$, $y \in \{left, right\}$
3. steering angle (δ)
4. yaw-rate ($\dot{\psi}$)
5. accelerations (a_x, a_y)
6. lateral velocity (v_y).
7. side-slips of the wheels (α_x), $x \in \{front, rear\}$.

Note that each of the collected signals is available from the onboard system of the vehicle except the lateral velocity (v_y) and the side-slip angles. These signals are only used in the model formulation process, but during the operation of the proposed control method are not required. The sampling time of the variables has been set to $T_s = 0.01s$. In this way, a large dataset has been created, which consisted of more than 10 million instances.

Categorization of instances

The goal of the parameter optimization process is to get a set of models which are stable, as a first step the instances in the dataset must be categorized by their stability. The stability of the instances have a high impact on the performances of the optimization process, thus the unstable instances must be removed from the dataset. [75] The separation of the stable/unstable instances using only the measured values, is not straightforward task. However, the stability condition, proposed in the previous chapter Eq. (3.1) can be used for this problem as well:

$$-\varepsilon < \frac{|1 + \alpha_1|}{|1 + \delta - \beta - \frac{l_f \dot{\psi}}{v_x}|} - 1 \leq \varepsilon, \tag{4.1}$$

where ϵ is a experimentally defined parameter.
Using this criterion the dataset can be divided into two categories: R_{st} represents the set of the instances, where the motion of the vehicle is approximated as stable. Similarly, R_{ust} denotes the set consisting of the instances with the approximation of unstable vehicle motion.

Nominal model for computing the error functions

After separating the instances, error functions are computed, which reflects on the nonlinear behavior of the vehicle. Basically, the error function describes the deviation between the nominal model and the measured variables.

The same nominal model is used as in the previous method Subsection 4.2 with the following state-space representation:

$$\dot{x} = Ax + Bu, \tag{4.2}$$

whose state-vector consists of $x = [\dot{\psi} \quad v_y]^T$ and its control input is the steering angle $u = \delta$. Furthermore, the measured variables have a sampling time of $T_s = 0.01s$, therefore the model is sampled by the same sampling time. Using the measured input signal δ, the outputs of the discrete system are computed for each measurement point.

The labeling of the collected data is based on the deviation of the measured signals from the signals of the nominal system. In this process the yaw-rate and the lateral velocities are involved, which are the independent states of the physical system. The labeling is based on the relative errors of the signals in time t_i, such as

$$\Delta\dot{\psi} = \dot{\psi}_m(t_i) - \dot{\psi}_n(t_i) \tag{4.3a}$$

$$\Delta v_y = v_{y,m}(t_i) - v_{y,n}(t_i) \tag{4.3b}$$

where $\dot{\psi}_m$ and $v_{y,m}$ denote the measured outputs while $\dot{\psi}_n$ and $v_{y,n}$ are the outputs of the nominal system.

Figure 4.1 shows the function of $\Delta\dot{\psi} - \Delta v_y$ computed from the collected dataset. Since neither $\Delta\dot{\psi}$ nor Δv_y are easy to calculate during the operation of the vehicle,

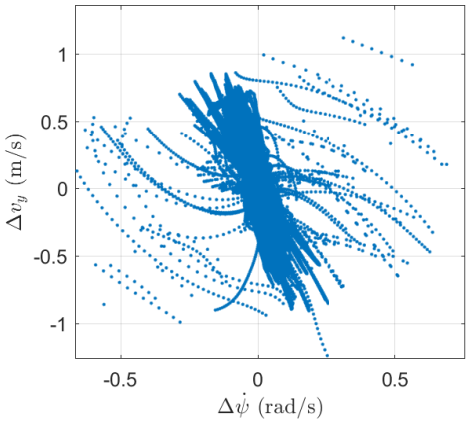


Fig. 4.1: Illustration of the error functions and their resolution

a method must found, which is able to appropriately approximate the error signals using only the available measurements.

4.1.1 Parameters optimization and determination of scheduling parameters

The main goal of the algorithm is to compute (or approximate) the selected output signal y by using another measured attributes, which are written in a matrix X. The algorithm tries to find the parameter vector ξ^*, which is the parameter vector of the true model and the output can be computed as

The pace regression algorithm (see Section 2.1.1) is used to compute the scheduling variables of the LPV system $\Delta\dot{\psi}, \Delta v_y$ from the dataset, which contains the measured attributes. The result of the pace-regression algorithm is a model, which can be used to approximate $\Delta\hat{\dot{\psi}}, \Delta\hat{v_y}$.

Parameter selection of the control-oriented model

The goal of the identification process is to determine the parameters of the model (4.5) for each segment. The structure of the state-space representation is determined in such a way to preserve the original structure of the physical model (4.2). In this case, the lateral model can be formed as:

$$\dot{x}_d = A_d(\rho)x_d + B_d u_d(\rho), \tag{4.4}$$

where

$$A_d(\rho) = \begin{bmatrix} a_{11}(\rho) & a_{12}(\rho) \\ a_{21}(\rho) & a_{22}(\rho) \end{bmatrix}, \quad B_d(\rho) = \begin{bmatrix} b_1(\rho) \\ b_2(\rho) \end{bmatrix}, \tag{4.5}$$

and $a_{11}(\rho), a_{12}(\rho), a_{21}(\rho), a_{22}(\rho)$ and $b_1(\rho), b_2(\rho)$ are parameters and the state-vector of the system is $x_d = [\dot{\psi}_i \quad v_y]$, the control input is $u_d = \delta$. The scheduling parameters are written into a vector form: $\rho = [\Delta\dot{\psi}, \quad \Delta v_y, \quad v_x]$. Since the parameter optimization process is especially difficult for continuous variables, the optimization is performed for a finite number of operating points. Each operating point is represented by constant vector ρ, in which the scheduling variables are fixed at constant values. The resolution of the scheduling variables is a crucial aspect of the parameter identification process. In order to cover the nonlinear dynamics of the vehicle precisely, the resolution must be as high as possible. However, the high resolution may make the computation of the system and the controller difficult. Therefore, a balance must be found between them. In this case, equidistant resolution is used: $\Gamma_{\dot{\psi}}$, Γ_{v_y} as shown in Figure 4.2. The last scheduling variable v_x is also ordered into a finite number of groups. n_{v_x}, $n_{\Delta v_y}$ and $n_{\Delta\dot{\psi}}$ represent the numbers of groups of the scheduling parameters.

The main goal if the identification process is to determine the parameters $a_{11}(\rho_i)$, $a_{12}(\rho_i), a_{21}(\rho_i), a_{22}(\rho_i)$ and $b_1(\rho_i), b_2(\rho_i)$ for each segment, where ρ_i denotes a specific operating point of the system with fixed ranges of the scheduling parameters. It can be written into a optimization problem:

$$\min_{a_{11}(\rho_i),a_{12}(\rho_i),a_{21}(\rho_i),a_{22}(\rho_i),b_1(\rho_i),b_2(\rho_i)} \sum_{j=0}^{N} \left(x_{m,\rho_i}(t_j) - x(t_j)\right)^2 \tag{4.6}$$

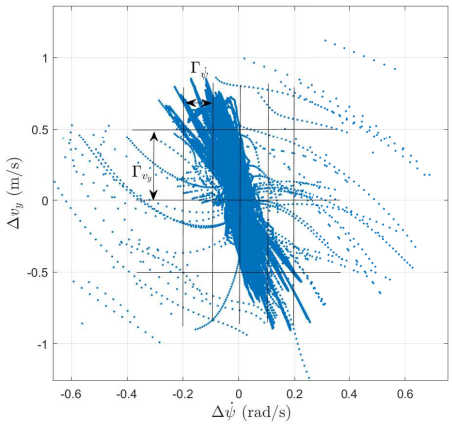


Fig. 4.2: Resolution of the scheduling variables

where $x_{m,\rho_i}(t_{ji})$ denotes the instances of the dataset, which belong to the operating range defined by ρ_i. N is the number of the samples within given operating range. $x(t_j)$ is the output of the nominal system. During the solution of (4.6) the systems can be computed independently for fixed ρ_i values in the grids. Thus, the parameter-dependent quadratic optimization problem leads to a least-squares problem [76, 77]. The result of the optimization is a set of polytopic systems, which represents the LPV description of the vehicle model.

4.1.2 Evaluation of the data-driven LPV models

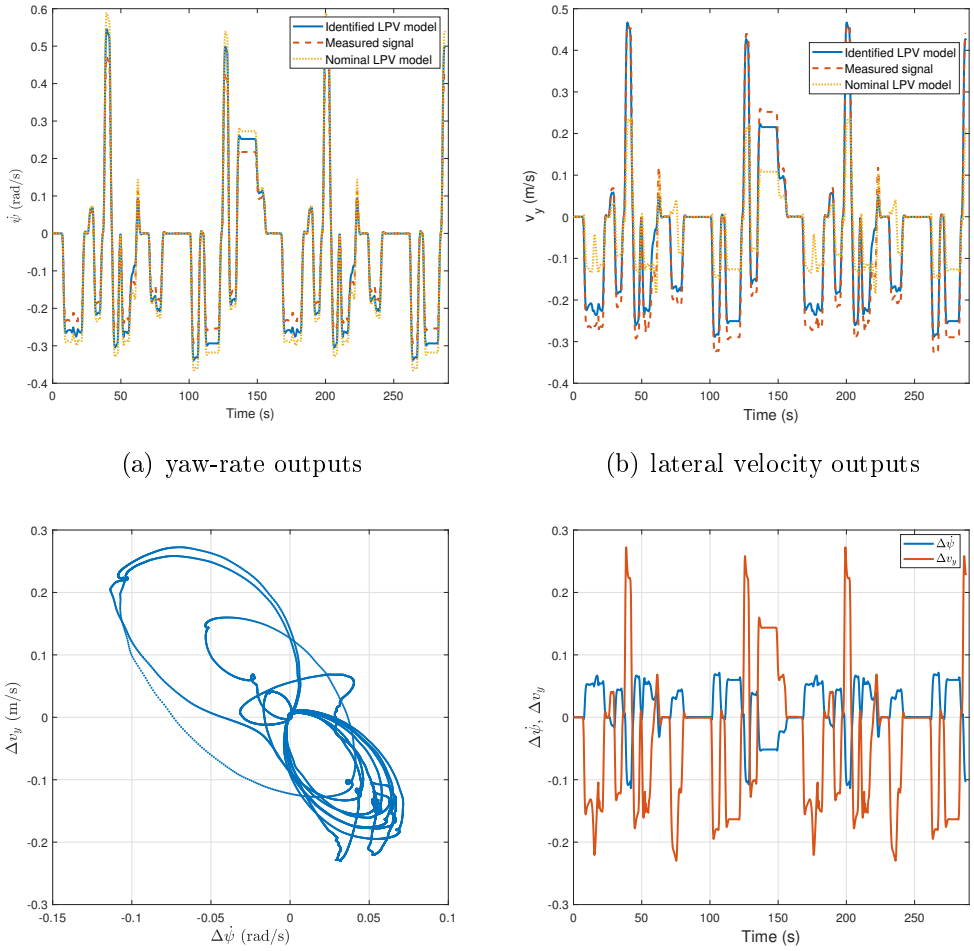
In the followings, a test case is presented to show the efficiency of the proposed parameter optimization method. The outputs of the optimized system is compared to the outputs of a nominal model at the end of this chapter. The parameters of the nominal model is given by the simulation software (CarSim) such as mass, inertia, geometrical parameters etc, specific values can be found in Table 4.1.

In the simulation, the vehicle is controlled by the in-built driver model of CarSim, and it is driven along a segment of Michigan Waterford Hill Track. Figures in 4.3 show the results of the control-oriented LPV system for a simulation-based test case. The dashed red lines represents the measured outputs ($\dot{\psi}$, v_y) of the nonlinear vehicle model from CarSim, dotted yellow lines are the outputs of the nominal LPV models (see (4.2)) and black lines illustrates the outputs of the identified system. It can be seen that the error between the measured and the computed outputs are smaller when the identified model is used. The difference is significant in case of v_y. When the nominal model is used the averaged error is $\approx 0.07m/s$, in case of the identified system is reduces to $\approx 0.017m/s$. The scheduling parameters are shown in Figures 4.3 (c,d). The scheduling parameters have significantly high ranges, which means the identified system works well at different operating points. It can be said that the identified system together with the selected scheduling parameters fits

Tab. 4.1: Parameters of the used D-class vehicle

Parameter	Notion	Value	Unit
Mass of the car	m	1690	kg
Yaw-inertia	J	4192	kgm^2
Location of front axis from COG	l_1	1.11	m
Cornering stiffness of front wheels	C_1	155160	N/rad
Location of rear axis from COG	l_2	1.66	m
Cornering stiffness of rear wheels	C_2	114659	N/rad
Front drag area of the car	A	1.8	m^2
Height of COG	h	0.56	m
Type of front suspensions	-	Independent	-
Mass of front suspensions	$m_{s,f}$	85	kg
Type of rear suspensions	-	Independent	-
Mass of rear suspensions	$m_{s,r}$	85	kg

better to the nonlinear model than the nominal model.



(a) yaw-rate outputs

(b) lateral velocity outputs

(c) Δv_y vs $\Delta\dot{\psi}$

(d) scheduling parameters with respect to time

Fig. 4.3: Evaluation of the optimized model

4.2 C4.5 decision tree algorithm-based parameter optimization method

In this section, another data-driven parameter optimization method is presented for modeling the lateral dynamics of the vehicle. In this case, the scheduling parameter of the model is selected by the C4.5 decision tree algorithm creating an LPV-based model.

Acquisition of data from simulations

The data acquisition is similar to the previously presented method in Section 4.1. However, some parameters from the measurements have also been excluded:

1. Longitudinal velocity (v_x)
2. Angular velocity of the wheels
   ($\omega_{x,y}$), $x \in \{front, rear\}$, $y \in \{left, right\}$
3. Steering angle (δ)
4. Yaw-rate ($\dot{\psi}$)
5. Accelerations (a_x, a_y)
6. Side-slip angle (β). Since the measurement of β can be difficult and expensive, this signal is only used for building up the model. During the operation of the control system it is not required.

The sampling time of the measurements has been set to $T_s = 0.01s$. In this manner, more than 10 million distinct instances have been collected.

Labeling the elements of the collected dataset

The learning of the vehicle dynamics requires the scaling and the ordering with labels of the collected data. The goal of the labeling is to provide categories, with which characteristics in the nonlinear vehicle motion can be distinguished from each other. The results of the layer are training and test sets, which are ready for the supervised learning process.

The labeling of the vehicle dynamic signals is based on their deviation from the signals of a nominal, physical model, which is described by the physical model of the vehicle, detailed in Subsection 2.2:

$$I\ddot{\psi} = C_{\alpha,f}\alpha_f l_1 - C_{\alpha,r}\alpha_r l_2 \tag{4.7}$$

$$mv_x(\dot{\psi} + \dot{\beta}) = C_{\alpha,f}\alpha_f + C_{\alpha,r}\alpha_r \tag{4.8}$$

The computation of the nominal, physical model signals require the state-space representation of the system $\dot{x} = A_p x + B_p u$, whose state vector is $x = \begin{bmatrix}\dot{\psi} & \beta\end{bmatrix}^T$, the control input is the steering angle $u = \delta$ and A_p, B_p are matrices of the physical model. This representation is transformed to a discrete representation through the sampling time $T_s = 0.01$, which is the same as the sampling time of the measurements. Using the measured input signal δ, the outputs of the discrete system are computed for each measurement point.

The labeling of the collected data is based on the deviation of the measured signals from the signals of the nominal system. In this process the yaw-rate and the side-slip angles are involved, which are the independent states of the physical system. The labeling is based on the relative errors of the signals in time t_i, such as

$$\dot{\psi}_e(t_i) = \frac{|\dot{\psi}_m(t_i) - \dot{\psi}_n(t_i)|}{\dot{\psi}_n(t_i)} \tag{4.9a}$$

$$\beta_e(t_i) = \frac{|\beta_m(t_i) - \beta_n(t_i)|}{\beta_n(t_i)} \tag{4.9b}$$

where $\dot{\psi}_m$ and β_m denote the measured outputs while $\dot{\psi}_n$ and β_n are the outputs of the nominal system. In the method categories has been predefined for the classification of the instances, such as n equidistant sections between 0 and 1. It is defined a function $\mathfrak{f}$, which associates the errors (4.9) with the categories. A given instance in t_i is labeled based on the following value

$$cat(t_i) = \max\Big(\mathfrak{f}\big(\dot{\psi}_e(t_i)\big), \mathfrak{f}\big(\beta_e(t_i)\big)\Big), \tag{4.10}$$

which means that the label $cat(t_i)$ is determined based on the maximum of the errors. The results of the labeling are the training and the test sets, where each instances have been categorized.

Selection of the sceduling vaiables

The selection of the scheduling variables of the LPV system is based on a decision tree algorithm. The role of the decision tree is to determine the most influential attributes using the previously labeled sets with *cat*. The main advantage of the decision trees is that they are able to provide reliable models even though the dataset has a highly nonlinear structure. For this classification problem, the C4.5 decision tree algorithm is applied [64, 65], which has been introduced in Subsection 2.3.3. The output of the decision tree is the category of a given instance ξ, which is used as scheduling parameter.

In the given data-driven vehicle modeling problem, the leaves of the decision tree are associated with specific categories *cat*. The resulted tree classifies the instances by using only the measured actual attributes. The output of the decision tree is ξ, which has the value *cat* for a specific leaf.

The parameter ξ is used as a scheduling variable of the system. The values of ξ, which represent categories, can be considered as a linear operating point of the nonlinear system. Figure 4.4 illustrates an example of the results. It can be seen that the location of the resulted categories (represented by different colors) in the plot of the side-slip and the lateral force on the wheel are well distinguished. The shape of the function is close to the tyre force characteristics, see [59]. It means that the machine learning algorithm is able to distinguish various sections, which is related to the levels of the relative errors (4.9).

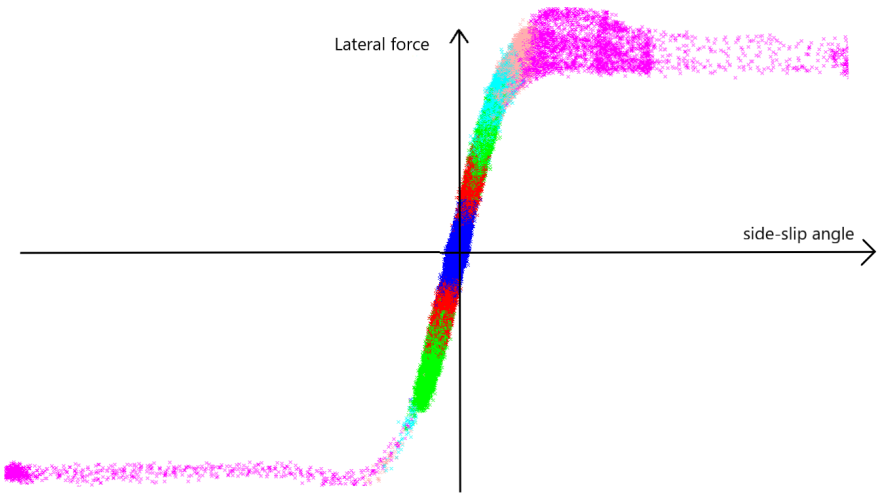


Fig. 4.4: Illustration of the calculated categories

Parameter selection of the LPV model

The identification of the LPV system requires the selection of the parameters for each segments. For each segment, a linear system can be adjusted and they create a gridded LPV system. The structure of the system is determined by the number of the states, which is derived from the physical model (4.7). In case of the lateral dynamics the data-driven system model is formed as:

$$\dot{x}_d = A_d(\xi)x_d + B_d u_d(\xi), \tag{4.11a}$$

$$A_d(\xi) = \begin{bmatrix} a_{11}(\xi) & a_{12}(\xi) \\ a_{21}(\xi) & a_{22}(\xi) \end{bmatrix}, \quad B_d(\xi) = \begin{bmatrix} b_1(\xi) \\ b_2(\xi) \end{bmatrix}, \tag{4.11b}$$

where $a_{11}(\xi), a_{12}(\xi), a_{21}(\xi), a_{22}(\xi)$ and $b_1(\xi), b_2(\xi)$ are parameters and the state-vector of the system is $x_d = [\dot{\psi}_i \ \beta_i]$, the control input is $u_d = \delta$.

Due to the trend of the automated vehicles the lateral positioning of the vehicle has a high importance. Thus, the lateral position of the vehicle y is computed using the states $\dot{\psi}$ and β. The lateral acceleration $\dot{v}_y$ is

$$\dot{v}_y = v_x(\dot{\psi} + \dot{\beta}) = v_x\dot{\psi} + v_x(a_{21}(\xi)\dot{\psi} + a_{22}(\xi)\beta + b_2(\xi)\delta). \tag{4.12}$$

The identified system description is augmented as:

$$\dot{x} = A(\rho)x + B(\rho)u, \tag{4.13a}$$

$$A(\rho) = \begin{bmatrix} a_{11}(\xi) & a_{12}(\xi) & 0 & 0 \\ a_{21}(\xi) & a_{22}(\xi) & 0 & 0 \\ v_x(1+a_{21}(\xi)) & v_x a_{22}(\xi) & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{4.13b}$$

$$B(\rho) = \begin{bmatrix} b_1(\xi) \\ b_2(\xi) \\ v_x b_2(\xi) \\ 0 \end{bmatrix}, \tag{4.13c}$$

the augmented state vector of the system is $x = \begin{bmatrix} \dot{\psi} & \beta & v_y & y \end{bmatrix}^T$, $u = u_d$ and ξ, v_x are selected as scheduling variables, whose vector is $\rho = \begin{bmatrix} \xi & v_x \end{bmatrix}$.

The problem of the identification is to find the parameters $a_{11}(\xi), a_{12}(\xi), a_{21}(\xi), a_{22}(\xi)$ and $b_1(\xi), b_2(\xi)$, with which the results of the state calculation x are close to the measured states x_m. It leads to an optimization problem, which is formed as:

$$\min_{a_{11}(\xi),a_{12}(\xi),a_{21}(\xi),a_{22}(\xi),b_1(\xi),b_2(\xi)} \big(x_m(t_i) - x(t_i)\big)^2 \tag{4.14}$$

for all $x_m(t_i)$ instances in all t_i time step in the training set. During the solution of (4.14) the systems can be computed independently for fixed ξ values in the grids. Thus, the parameter-dependent quadratic optimization problem leads to a least-squares problem [76, 77]. The result of the optimization is a set of systems, which represents the LPV description of the vehicle model.

4.3 Path following LPV control design using the data-driven model

In this section, a lateral control-related application of the proposed modeling methods is presented. This control solution is based on the second method, presented in Section 4.2 but in also can be used with the first method. The main steps of the whole algorithm - including the modeling and the control design phases - are illustrated in Figure 4.5. The modeling and control subtasks can be divided into four main groups, such as: *Prepocess of data*, *Model identification*, *Control design*, data acquisition and implementation using the *Simulation environment*.

The layer 'Simulation environment' contains the vehicle dynamic simulation software. It has role in the data acquisition process for the machine learning algorithm, e.g. training and test sets. Moreover, it is used for the evaluation of the designed controller. In this case, the high-fidelity CarMaker software is used for this purpose.

The role of the layer 'Preprocess of data' is to produce the training and test sets from the collected data. It scales the data and it also orders them into various

categories. The results of the layer are training and test sets, which are ready for the supervised learning process.

The 'optimization process' layer uses the provided labeled sets to determine the scheduling variables of the system. It requires the selection of the variables, which has the most impact on the dynamics of the vehicle. Moreover, in this layer the parameters of the LPV-based vehicle model are calculated.

In the layer 'Control design' the LPV controller based on the resulted model is designed. The performances of the controller is evaluated using the 'Simulation environment'. The results of the simulation scenarios are presented in Section 4.4.

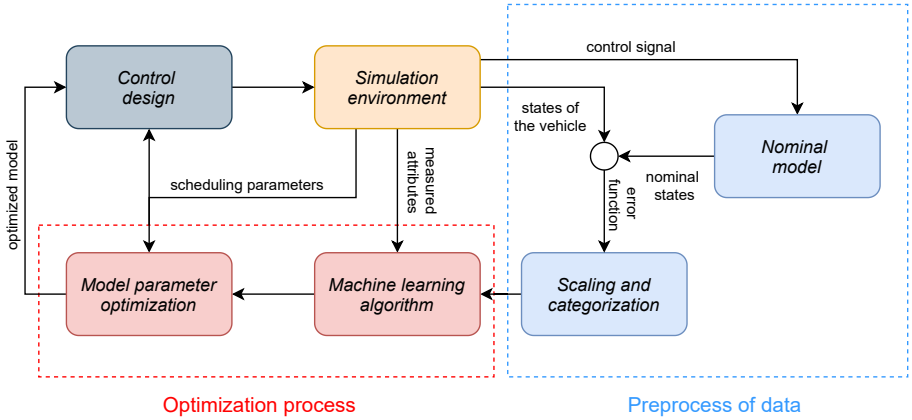


Fig. 4.5: Structure of the system

In this section a path following control design method is presented using the previously identified data-driven vehicle model. The control system is responsible for guaranteeing the trajectory tracking of the vehicle and for minimizing the interventions.

The goals of the control problem are described by the performances.

- *Minimization of the lateral error*
  In order to reach good path following property, the control system has to minimize the lateral error between the road y_{ref} and the lateral position of the vehicle y:

$$z_1 = y_{ref} - y, \qquad |z_1| \rightarrow min, \tag{4.15}$$

  where y_{ref} is determined by the selected route of the vehicle.

- *Minimization of the yaw-rate error*
  Beside the lateral error, the controller has to reduce the error between the reference $\dot{\psi}_{ref}$ and the measured yaw-rate $\dot{\psi}$ in order to reach accurate and smooth tracking:

$$z_2 = \dot{\psi}_{ref} - \dot{\psi}, \qquad |z_2| \rightarrow min, \tag{4.16}$$

where $\dot{\psi}_{ref}$ is determined by the curvature of the route and the velocity of the vehicle, see [59].

- *Minimization of the steering angle*
  The control system has to minimize its interventions to reduce the energy consumption, which means the minimization of the steering angle:

$$z_3 = \delta, \qquad |z_3| \to min. \tag{4.17}$$

The presented performances are summarized in the following vector $z = \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T$, which can be expressed in a performance equation

$$z = C_1 x + D_{11} r + D_{12} u, \tag{4.18}$$

where C_1, D_{11}, D_{12} are matrices and r contains the signal y_{ref}.

The system, which is used for the control design is based on the data-driven model (4.13), the performance equation (4.18) and the measurement equation:

$$\dot{x} = Ax + Bu, \tag{4.19a}$$
$$z = C_1 x + D_{11} r + D_{12} u, \tag{4.19b}$$
$$y_K = C_2 x, \tag{4.19c}$$

where (4.19c) represents the measurement of $y_K = \begin{bmatrix} y & \dot{\psi} \end{bmatrix}$.

In the LPV control design, the extended state-space model (4.19) is employed. Moreover, during the control design several weighting functions are used to scale the measured signals and to reach the specified performance level. The augmented plant with the weighting functions are illustrated in Figure 4.6. The roles of the weighting functions $W_{ref,1}$ and $W_{ref,2}$ are to scale the reference signals y_{ref} and $\dot{\psi}_{ref}$. Functions $W_{z,1}$ and $W_{z,2}$ guarantee the accurate tracking performance of the vehicle, while $W_{z,3}$ scales the control intervention. Finally, the roles of the weighting functions $W_{w,1}$, $W_{w,2}$ and $W_{w,3}$ are to scale the noises of the measured signals.

The quadratic LPV performance problem is to choose the parameter-varying controller $K(\rho)$ in such a way that the resulting closed-loop system is quadratically stable and the induced $\mathcal{L}_2$ norm from the disturbance and the performances is less than the value γ. The minimization task is the following:

$$\inf_{K(\rho)} \sup_{\rho \in F_\rho} \sup_{\|w\|_2 \neq 0, w \in \mathcal{L}_2} \frac{\|z\|_2}{\|w\|_2}, \tag{4.20}$$

where F_ρ bounds the scheduling variables. The yielded controller $K(\rho)$ is formed as

$$\dot{x}_K = A_K(\rho) x_K + B_K(\rho) y_K, \tag{4.21a}$$
$$u = C_K(\rho) x_K + D_K(\rho) y_K, \tag{4.21b}$$

where $A_K(\rho), B_K(\rho)$ and $C_K(\rho), D_K(\rho)$ are scheduling variable dependent matrices.

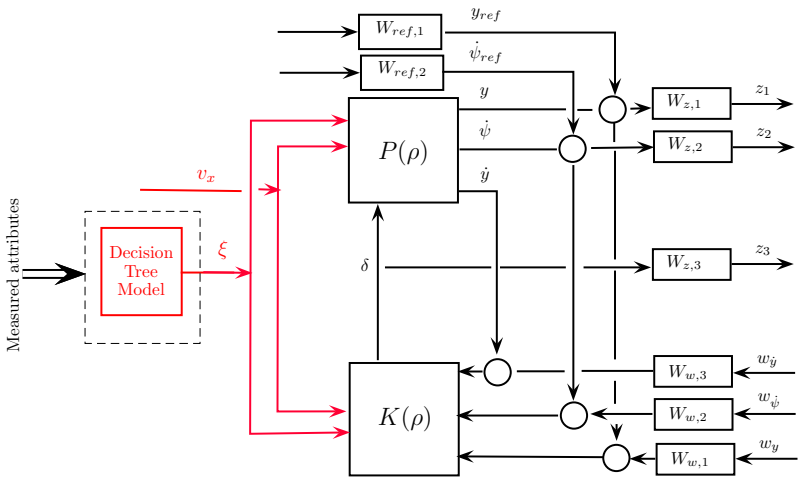


Fig. 4.6: Augmented plant for LPV control design

4.4 Simulation results

In this section a simulation example is presented to show the operation and the effectiveness of the proposed control system. The example presents the results of the decision tree generation, the selection of the weighting functions in the control design and the evaluation of the designed controller through a comparative vehicle dynamic scenario.

In the generation of the decision tree the number of the categories of the error is set to $n = 6$, which means that the values of the error function are divided into 6 groups. Using these groups, several decision have been created, their results are summarized in Table 4.2. The first column of Table 4.2 shows the variables, which are used by the given decision tree. The second column represents the number of the minimal objects per leaf, which parameter highly influences the size of the tree. Third column gives the percentage of the correctly classified instances. As a contribution, the best result is produced by the largest tree, which consists of almost 70 elements. However, due to the increased size it is not recommended to use in practice. Therefore in the fourth case, the tree uses only two $(\delta, \dot{\psi})$ attributes and it also leads to an appropriate classification capability with more than $> 94\%$. In the rest of the example the setting of the fourth tree is selected.

Table 4.3 illustrates the corresponding confusion matrix, which the percentages of the classification for all scenarios. The first row represents the output of the decision tree, while the last column shows the correct class of the instances. It can be seen seen that the misclassification is rare, the largest value is only 0.67%. Moreover, most of the misclassified instances are only 1 class away from the correct class. Thus, the designed decision tree has a good classification performance, it can be used effectively in the value selection of the scheduling variable.

The design of the LPV controller requires the selection of the weighting functions, in which the disturbances and the reference signals are scaled and the performances

Tab. 4.2: Relationship between the tree size and its correctness

Used Parameters	Min. obj.	Corr. Class. Inst.	Size
$\delta, \dot{\psi}, a_x, \omega_{fl}, \omega_{rr}$	500	98.1%	69
$\delta, \dot{\psi}, a_x, \omega_{fl}, \omega_{rr}$	1000	97.5%	41
$\delta, \dot{\psi}, a_x, \omega_{fl}, \omega_{rr}$	2000	96.7%	33
$\delta, \dot{\psi}$	5000	94.5%	17
$\delta, \dot{\psi}$	10000	93.7%	11

Tab. 4.3: Confusion matrix

$\xi = 1$	$\xi = 2$	$\xi = 3$	$\xi = 4$	$\xi = 5$	$\xi = 6$	
47.4	0.67	0	0	0	0	$cat = 1$
2	30.3	1.1	0	0	0	$cat = 2$
0	0.38	18.2	0.1	0	0	$cat = 3$
0	0	0.24	1.8	0	0	$cat = 4$
0	0	0	0.24	0.7	0	$cat = 5$
0	0	0.23	0	0	0.7	$cat = 6$

are specified. In the given lateral vehicle control problem the weighting functions on the reference signals are selected as

$$W_{ref,1} = \frac{0.1}{100s + 1}, \tag{4.22a}$$

$$W_{ref,2} = \frac{0.01}{100s + 1}, \tag{4.22b}$$

where $s \to 0$ depicts the steady state value of the reference and the exponents of s represent the fastness of the signal dynamics. Furthermore, the goals of functions $W_{z,1}$ and $W_{z,2}$ are to guarantee the accurate trajectory tracking of the vehicle:

$$W_{z,1} = 10\frac{1 - \frac{n}{10}}{s^2 + 2s + 1}, \tag{4.23a}$$

$$W_{z,2} = \frac{1}{s + 1}, \tag{4.23b}$$

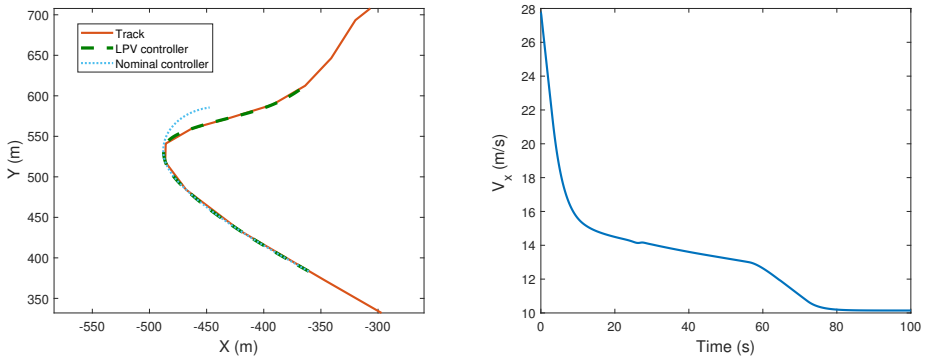
where $s \to 0$ depicts the performance requirement for steady-state case. Weight $W_{z,3}$ scales the control intervention and it provides a balance between minimization in the control energy and the further performances:

$$W_{z,3} = 0.01\frac{1s + 1}{2s + 1}. \tag{4.24}$$

The last three functions scales the noises on the measured signals, such as $W_{w,1} = 0.002$, $W_{w,2} = 0.001$, $W_{w,3} = 0.05$.

In the following, a comparative simulation example is presented, in which the vehicle with different controllers is driven along a sharp bend with varying velocity.

First, the car is controlled by a nominal controller, which is designed based on the physical model of the vehicle. In the second case the vehicle is driven by the proposed control system. Figure 4.7(a) shows the path of the vehicle in both cases and Figure 4.7 (b) presents the variation of the velocity. It can be seen in Figure 4.7(a), when the vehicle is controlled by the nominal controller, it leaves the track at the beginning of the bend. However, when the car is driven by the proposed control algorithm, the vehicle is able to follow the track with a low lateral error.



(a) caption (b) Velocity profile

Fig. 4.7: Positions and velocity of the vehicles

The performance of the yaw-rate tracking is illustrated in Figure 4.8. As it shows, the proposed control system can guarantee the yaw-rate tracking a vehicle despite the varying velocity. The averaged delay of the tracking is below $< 0.2s$, which is a quite low value.

The scheduling parameter ξ of the LPV system is calculated by the using the resulted decision tree. Based on the previous analysis in Table 4.2, the chosen decision tree uses two signals: the steering angle and the yaw-rate. Figure 4.9(a) illustrates the variation of ξ together with $\dot{\psi}$ and δ. The scheduling parameter covers three category $[0; 1; 2]$, which are associated with the specific identified models. The

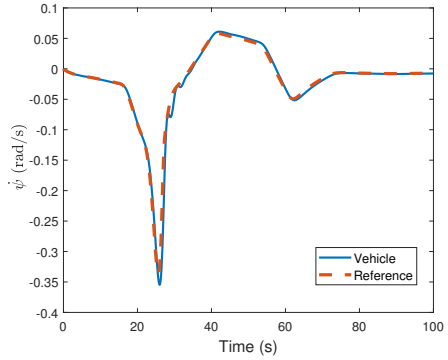
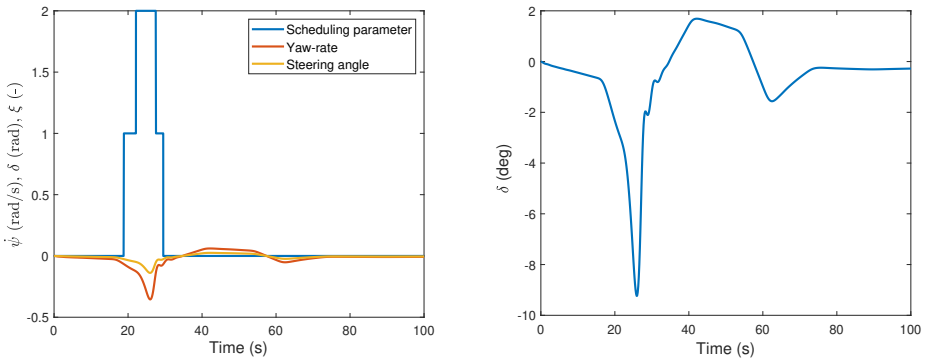


Fig. 4.8: Yaw-rate tracking

decision tree yields the category "2" when the steering angle and the yaw-rate signals reach their peak values. By selecting the appropriate category, the stable motion and tracking performance of the vehicle can be guaranteed.



(a) Scheduling variable ξ (b) Steering angle

Fig. 4.9: Scheduling variable ξ and steering angle (δ)

Finally, the resulted steering angle is illustrated separately in the Figure 4.9(b). As it shows, the values of the steering angle is between $(-9, 2deg)$, which is a physically reasonable range for the steering signal.

Thesis 3 I have developed two methods for modeling the lateral dynamics of the vehicle. Both methods are based on the Linear Parameter Varying (LPV) approach combining with different machine-learning-based algorithms: pace regression and decision tree generation. I have shown that the models provided by the proposed algorithms are more efficient at modeling the lateral dynamics of the vehicle than the classical purely physical-based modeling approaches. I have developed a lateral robust controller using the proposed model structures.

Related publications: [FNG21d, FN20, FNG20b, FNG20c]

5. DESIGN OF AN INTEGRATED CONTROL SYSTEM FOR VARIABLE-GEOMETRY SUSPENSION SYSTEM

Beside the presented data-driven approaches, there are other ways to improve the performances and the stability of the lateral control systems. For example, the possibility of independent steering can be useful tool to improve the maneuverability of the vehicle. A possible way to implement an independent steering system is the variable-geometry suspension. This component has gained attention from the automotive industry in the recent decades. The application of the variable-geometry suspension system can have several reasons. For example, beside the stability of the vehicle, the variable-geometry suspension can improve the riding comfort of the vehicle. However, the application of the variable-geometry suspension, as an independent steering system is a relatively new concept, which still holds some challenges. The data-driven methods can also be used for modeling and control the variable-geometry suspension system, a possible solution will be provided in the next chapter.

In this study, a variable-geometry suspension-based independent steering system is investigated, by which the lateral stability and the maneuverability of the vehicle can be increased. This chapter consists of the following main parts:

- The physical modeling of the variable-geometry suspension-based steering system is presented in Section 5.1.2.

- A hierarchical control structure is proposed in Section 5.2, which includes two main layers: Low-level, the control design for the variable-geometry suspension-based steering system, High-level Lateral control design, which computes the reference signal for the suspension controller.

- In Section 5.3 a coordination strategy is presented, which aims to select the appropriate intervention considering the physical limits of the actuators.

- Finally, a comprehensive simulation example is given to the operation and the effectiveness of the proposed steering system and the control strategy.

5.1 Modeling and control for a variable-geometry suspension-based steering system

In this section modeling of the variable-geometry suspension system is proposed. It is incorporated in the formulation of the suspension, steering and lateral dynamics. The goal of this section is to formulate the control-oriented models of these dynamics, by which the functionalities of the variable-geometry suspension, such as independent steering and torque vectoring can be described.

Modeling of suspension dynamics

The purpose of the variable-geometry suspension is to perform the wheel camber angle and the scrub radius modification through a torque actuation. The actuator is incorporated in the suspension between the wheel hub and the wheel, therefore it is able to generate an active torque M_{act} around B to tilt the wheel, see the scheme of the suspension in Figure 5.1. It results in the modification of the scrub radius, by which the steering angle through the longitudinal force is influenced. However, the motion of the suspension and the wheel is influenced by several effects, such as the suspension and tyre compression, the suspension arm and the lateral force in the tyre-ground contact. In the followings the resulted forces are formulated to generate the motion equation of the wheel tilting.

The force of the suspension compression and damping F_{susp} is formulated as

$$F_{susp} = s_{susp}\left(\frac{z_w + z_{w,0}}{\sin\epsilon_1}\right) + d_{susp}\frac{\dot{z}_w}{\sin\epsilon_1} \tag{5.1}$$

where s_{susp} and d_{susp} are the stiffness and damping coefficients, $z_{w,0}$ is the joint position, resulting from the static suspension compression. Similarly, F_{tyre} is resulted from the tyre compression, which has a direction to the wheel:

$$F_{tyre} = s_{tyre}\frac{(r_w\cos\gamma - l_{tyre}\sin\gamma - r_w - z_w) + z_{tyre,0}}{\cos\gamma} \tag{5.2}$$

where s_{tyre} is the tyre stiffness, r_w is the wheel radius and $z_{tyre,0}$ is the static compression of the tyre.

The motion of the suspension is described by three dynamic equations. First, the vertical motion of the wheel hub z_w is formulated (5.3). Second, the torque on the suspension is described to derive the rotation of the wheel hub ϵ_3, see (5.4). Third, the tilting of the wheel γ is formulated (5.5).

$$m_{susp}\ddot{z}_w = -F_{susp}\sin\epsilon_1 + F_{tyre}\cos\gamma + F_{arm}\sin\epsilon_2 \tag{5.3}$$

$$J_{susp}\ddot{\epsilon}_3 = F_{susp}l_{susp} + F_{arm}l_{arm} - M_{act} \tag{5.4}$$

$$J_w\ddot{\gamma} = M_{act} - F_{tyre}l_{tyre} - F_{y,\gamma}l_y \tag{5.5}$$

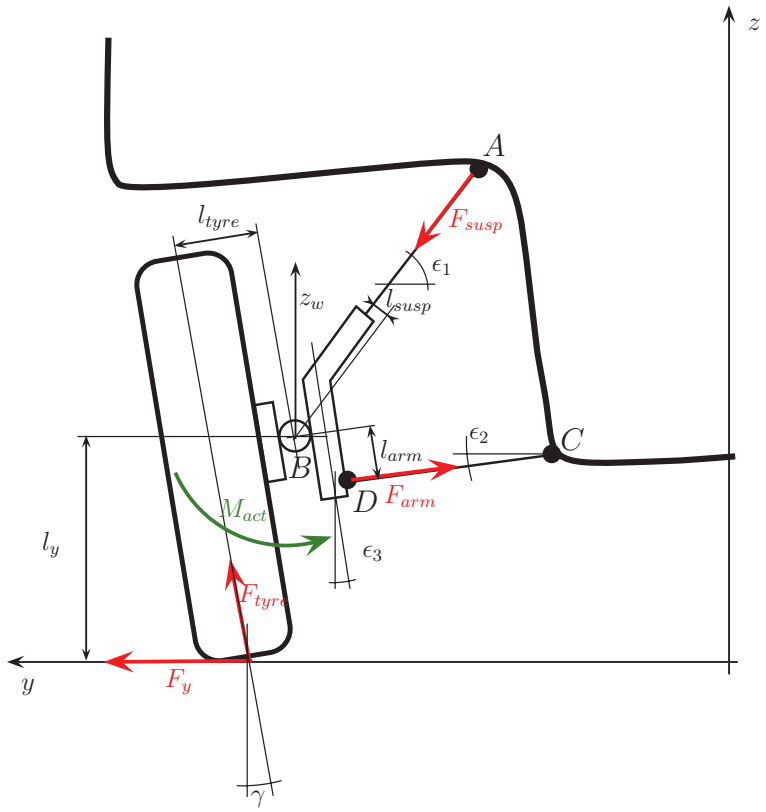


Fig. 5.1: Scheme of the suspension construction

The arm of the lateral tyre force $F_{y,\gamma}$ is

$$l_y = r_w \cos\gamma - l_{tyre} \sin\gamma. \tag{5.6}$$

where r_w and l_{tyre} are construction parameters.

The force in the arm CD is computed through the following assumption. Since the inertia and the stiffness of the suspension through F_{susp} and F_{arm} are high, the effect of M_{act} on the suspension motion is relatively small. Thus, in practice the angle ϵ_3 is constant, thus $\ddot{\epsilon}_3 = 0$. Therefore, F_{arm} is computed as:

$$F_{arm} = \frac{M_{act} - F_{susp} l_{susp}}{l_{arm}} \tag{5.7}$$

Thus, it is assumed that ϵ_1, ϵ_2 and l_{arm}, l_{susp} can be handled as constant suspension parameters.

Since the aim of the modeling is to generate a control-oriented formulation, some further assumptions are made. Therefore, the nonlinear suspension model (5.3)...(5.5) is transformed into a linear form.

- In the formulation small wheel tilting angles are considered. As a result $\cos\gamma = 1$ and $\sin\gamma = \gamma$ approximations are placed into (5.3)...(5.5).

- Since γ values are small, the lateral tyre force $F_{y,\gamma}$ is approximated in a linear form: $F_{y,\gamma} = C_1\alpha$, where α is the side-slip angle of the tyre and C_1 is the tyre stiffness of the front wheel. During the wheel tilting motion $\alpha = \tan\left(\frac{r_w\dot{\gamma}}{v}\right) \approx \frac{r_w\dot{\gamma}}{v}$ results in the lateral side-slip angle, which is the angle between the longitudinal and the lateral component of the velocity vector. Thus, the resulting lateral tyre force is

$$F_{y,\gamma} = C_1\frac{r_w\dot{\gamma}}{v} \tag{5.8}$$

- The static compressions of the suspension and the tyre are neglected, see (5.1) and (5.2).

The approximations yield the following transformed model of equations (5.3)...(5.5)

$$\begin{aligned} m_{susp}\ddot{z}_w = &\left(1 + \frac{l_{susp}}{l_{arm}}\frac{\sin\epsilon_2}{\sin\epsilon_1}\right)(-s_{susp}z_w - d_{susp}\dot{z}_w) + \\ &+ s_{tyre}(-l_{tyre}\gamma - z_w) + \frac{\sin\epsilon_2}{l_{arm}}M_{act} \end{aligned} \tag{5.9}$$

$$\begin{aligned} J_w\ddot{\gamma} = &M_{act} - s_{tyre}l_{tyre}(-\gamma l_{tyre} - z_w) - \\ &- C_1\frac{r_w\dot{\gamma}}{v}(r_w - l_{tyre}\gamma) \end{aligned} \tag{5.10}$$

It can be seen that in (5.10) a nonlinear term $C_1\frac{r_w\dot{\gamma}}{v}\cdot l_{tyre}\gamma$ is found. Since in the approximation the small wheel camber angles are considered, the $\gamma \approx 0$ is used to avoid the nonlinearities. Thus, the following state-space representation of the variable-geometry suspension model is yielded

$$\dot{x}_{susp} = A_{susp}x_{susp} + B_{susp}u_{susp} \tag{5.11}$$

where the state vector is $x = \begin{bmatrix}\dot{z}_w & z_w & \dot{\gamma} & \gamma\end{bmatrix}^T$, $u = M_{act}$ and A_{susp} and B_{susp} are matrices.

5.1.1 Modeling of steering dynamics

The steering dynamics creates interconnection between the camber and the steering angle. In dynamic relationship is formulated below. The steering rotation of a wheel is described by the dynamical equation

$$\ddot{\delta}_i = \frac{F_{l,i}r_{\delta,i}}{J_{\delta,i}}, \quad i \in \{l, r\}. \tag{5.12}$$

where $r_{\delta,i}$ is the scrub radius, $F_{l,i}$ is the longitudinal traction/braking force on the wheel, $J_{\delta,i}$ is the inertia of the steering system for one wheel. The relationship

between $r_{\delta,i}$ and the camber angle γ_i depends on the suspension geometry, e.g. the positions of the points $A \dots D$. Generally, this relationship can be formulated linearly [NFG16b], such as $r_{\delta,i} = \varepsilon\gamma_i$, where ε is a construction parameter. Thus, the steering dynamics is described as

$$\ddot{\delta}_i = \frac{\varepsilon F_{l,i}}{J_{\delta,i}}\gamma_i, \quad i \in \{l, r\}. \tag{5.13}$$

The state-space representation of the steering wheel dynamics for each wheel is formed as

$$\dot{x}_{st,i} = A_{st}x_{st,i} + B_{st,i}(F_{l,i})u_{lat,i}, \tag{5.14}$$

where

$$A_{st} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad B_{st,i}(F_{l,i}) = \begin{bmatrix} \frac{\varepsilon F_{l,i}}{J_{\delta,i}} \\ 0 \end{bmatrix},$$

the state vector is $x_{st,i} = \begin{bmatrix} \dot{\delta}_i & \delta_i \end{bmatrix}^T$ and the control input is $u_{st,i} = \gamma_i$.

5.1.2 Modeling of lateral dynamics

The role of the lateral vehicle dynamics is to formulate the relationship between steering angle δ, the longitudinal wheel force $F_{l,i}$ and the vehicle motion. First, the nonlinear formulation of the lateral model is proposed, and second the nonlinear model is linearized to generate a control-oriented model. However, the nonlinear model is used for analysis purposes, while the aim of the linearizing is to get a model for control design.

The lateral motion of the vehicle is described by the previously presented two-wheeled bicycle model, see Section 2.11:

$$I_z\ddot{\psi} = F_{f,y}(\alpha_f)l_1 - F_{r,y}(\alpha_r)l_2 + M_d, \tag{5.15a}$$

$$mv_x\left(\dot{\psi} + \dot{\beta}\right) = F_{f,y}(\alpha_f) + F_{r,y}(\alpha_r), \tag{5.15b}$$

$$\dot{y} = v_x\beta, \tag{5.15c}$$

where m is the vehicle mass, I_z is the yaw-inertia of the vehicle, l_f and l_r are geometric parameters, $\dot{\psi}$ is the yaw rate of the vehicle, β is the side-slip angle, $\dot{y}$ is the lateral velocity and M_d is the differential yaw moment. $F_{i,y}$ represents the lateral forces on the front and rear wheels. The side-slip angles of the front and rear axles α_f and α_r are expressed: $\alpha_f = \delta - \beta - \frac{\dot{\psi}l_1}{v_x}$ and $\alpha_r = -\beta + \frac{\dot{\psi}l_2}{v_x}$. The vehicle model (5.15a),(5.15b) can be reformulated as follows:

$$\dot{\alpha}_r - \dot{\alpha}_f = \left[\frac{l_1 + l_2}{I_z v_x}\left(F_{f,y}(\alpha_f)l_1 - F_{r,y}(\alpha_r)l_2\right)\right] - \dot{\delta} + \frac{l_1 + l_2}{I_z v_x}M_d, \tag{5.16a}$$

$$\dot{\alpha}_f l_2 + \dot{\alpha}_r l_1 = v(\alpha_r - \alpha_f) - \frac{l_1 + l_2}{mv_x}\left[F_{f,y}(\alpha_f) + F_{r,y}(\alpha_r)\right] + v_x\delta + l_2\dot{\delta}. \tag{5.16b}$$

In the reformulated equations the side-slips α_f, α_r are the states of the system. Thus, the vehicle model is transformed to a state-space representation as below:

$$\dot{x} = \begin{bmatrix} \dot{\alpha}_f \\ \dot{\alpha}_r \end{bmatrix} = \begin{bmatrix} \mathrm{f}_1(\alpha_f, \alpha_r) \\ \mathrm{f}_2(\alpha_f, \alpha_r) \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} M_d + \begin{bmatrix} h_{11} h_{12} \\ h_{21} h_{22} \end{bmatrix} \begin{bmatrix} \dot{\delta} \\ \delta \end{bmatrix} = \\ = f(x) + gM_d + h\Delta \tag{5.17}$$

where $f(x), g, h$ are matrices of functions [78].

For analysis purposes the modeling of the lateral forces $\mathcal{F}_i(\alpha_i)$ are formulated as polynomials [78], such as $\mathcal{F}(\alpha) = \sum_{k=1}^{n} c_k \alpha^k = c_1\alpha + c_2\alpha^2 + \ldots + c_n\alpha^n$, where c_i coefficients are constants. Through the polynomial description the nonlinear characteristics of the tyre can be considered, by which the model is suitable for vehicle dynamic analysis. However, for control design purposes the function of the lateral force is linearized as $\mathcal{F}_i(\alpha_i) = C_i\alpha_i$. Moreover, in the control design the lateral displacement of the vehicle (5.15c) also has an efficient role. Thus, the representation (5.17) together with (5.15c) are transformed to a further state-space representation, which is described as

$$\dot{x}_{lat} = A_{lat} x_{lat} + B_{lat} u_{lat} \tag{5.18}$$

where the new state vector is $x_{lat} = \begin{bmatrix} \dot{\psi} & \beta & y \end{bmatrix}^T$ and $u_{lat} = \begin{bmatrix} \delta & M_d \end{bmatrix}^T$.

5.2 Hierarchical design of the control system

The modeling of variable-geometry suspension requires several subsystems, as presented in the previous section. Since these dynamices are different in their fastness and in the complexity of their modeling, it is recommended to handle them in a hierarchical structure [79, 80]. Thus, the control systems of the variable-geometry suspension is designed separately, such as the design of the lateral control, the steering control and the suspension control. These control systems are connected to each other through the control signals and the references, see Figure 5.2. In case of the proposed structure the steering and suspension controllers have similar structure on the left and the right side. Moreover, in the hierarchical design the stability and the performances must be guaranteed.

The purpose of this section is to propose a robust control design of the independent steering and torque vectoring. In the following, the control design on the lateral, steering and suspension dynamics together with the formulation of the uncertainty of the system are shown. The control design of the in-wheel motors can be found e.g. in [81, 82].

5.2.1 Variable-geometry suspension control

The objective of the variable-geometry suspension control is to realize the camber angle, which is yielded by the robust controller. The controlled system must

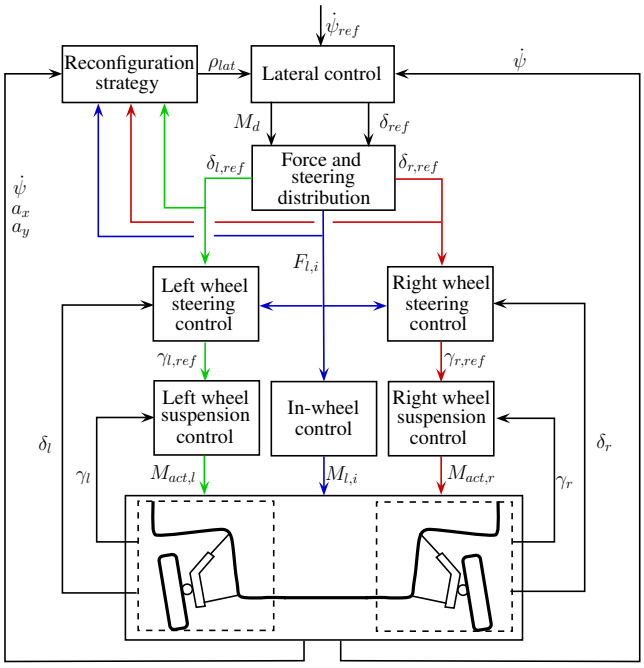


Fig. 5.2: Control system architecture

guarantee the precise tracking of $\gamma_{i,ref}$ without high overshoots and time delays. The control input of the system is $M_{act,i}$, which guarantees wheel tilting.

The performance of the suspension control is the minimization of the deviations of the current wheel camber angle from the reference angle both on either side of the first axle, such as

$$z_{susp} = \gamma_{i,ref} - \gamma_i, \quad |z_{susp}| \rightarrow min \tag{5.19}$$

where $\gamma_{i,ref}$is computed by the control inputs of the steering control. Extending the state-space representation of the suspension (5.11) with the performance, the augmented plant is the following:

$$\dot{x}_{susp} = A_{susp}x_{susp} + B_{susp}u_{susp} \tag{5.20a}$$
$$z_{susp} = C_{susp,1}x_{susp} + D_{susp,1}u_{susp} \tag{5.20b}$$
$$y_{susp} = C_{susp,2}x_{susp} + D_{susp,2}w_{susp} \tag{5.20c}$$

and the measured signals are the camber angle of the wheels on either side of the front axle, i.e., $\gamma_i = C_{susp,2}x_{susp}$. Moreover, w_{susp} is the sensor noise on the measurement, which must be rejected by the controller.

The control design is also based on the robust $\mathcal{H}_\infty$ method. The purpose is to design a suspension controller $K_{susp,i}$, which guarantees that the closed-loop system is asymptotically stable and the closed-loop transfer function from w_{susp} to z_{susp} satisfies the following constraint

$$\left\| T_{z_{susp},w_{susp}} \right\|_\infty < \Gamma_{susp}, \tag{5.21}$$

for a given real positive value Γ_{susp}. The realized control signals on either side of the first axle are the active torques around the center of the vertical axle of the vehicle, i.e., $M_{act,l}$ and $M_{act,r}$.

5.2.2 Design of suspension based steering control and uncertainty modeling

The control design of the steering dynamics is based on the state-space formulation (5.14). The purpose of the steering control is to guarantee the tracking of the reference steering signal $\delta_{i,ref}$, which resulted by the steering distribution:

$$z_{st,i,1} = \delta_{ref} - \delta_i, \quad |z_{st,i,1}| \to min \tag{5.22}$$

Moreover, the performance $z_{st,1}$ must be reached using minimum control input:

$$z_{st,i,2} = \gamma_{i,ref}, \qquad |z_{st,i,2}| \to min \tag{5.23}$$

The performances are formed in a vector, such as $z_{st,i} = \begin{bmatrix} z_{st,i,1} & z_{st,i,2} \end{bmatrix}^T$. The state-space representation of the steering dynamics, which incorporates the performances and the measurement, is the following:

$$\dot{x}_{st,i} = A_{st} x_{st,i} + B_{st,i}(\rho_{st,i}) u_{st,i} \tag{5.24a}$$
$$z_{st,i} = C_{st,1} x_{st,i} + D_{st} u_{st,i} \tag{5.24b}$$
$$y_{st,i} = C_{st,2} x_{st,i} \tag{5.24c}$$

where $y_{st} = \delta_i$ is the measured signal. The matrix $B_{st,i}(\rho_{st,i})$ depends on the scheduling variable $\rho_{st,i} = F_{l,i}$, which is computed by the force distribution strategy, see Figure 5.2.

Thus, it is necessary to design controllers for both the left and the right wheels. Since matrix $B_{st,i}(\rho_{st,i})$ depends on a scheduling parameter, the LPV control design method is applied in the following. The LPV design is based on a weighting strategy, which is formulated through a closed-loop interconnection structure of (5.24), see Figure 5.3. The selection of input and output weighting functions is typically based on the specifications of disturbances and the performances. The purpose of weighting functions $W_{st,1}$ and $W_{st,2}$ are to define the performance specifications in such a way that a trade-off is guaranteed between $z_{st,i,1}$ and $z_{st,i,2}$. They can be considered as penalty functions, i.e. weights should be large where small signals are desired and small where large performance outputs can be tolerated. The weight of $z_{st,i,1}$ is chosen in the form $W_{st,1} = \frac{A_{st,1}}{T_{st,1}s+1}$, which scales the admissible tracking error. The actuation γ_i is scaled with the function in the form $W_{st,2} = A_{st,2}$, which determines the amplitude of the control signal. The aim of the function $W_{st,ref} = A_{st,3}$ is to scale the reference signal $\delta_{i,ref}$. Furthermore, $W_{st,sens} = \frac{A_{st,4}}{T_{st,4}s+1}$ incorporates the scaling of the sensor noise, such as the maximum amplitude and the maximum frequency of the steering angle measurement error. The signals $e_{st,i}$ and $\delta_{i,ref}$ are handled as disturbances in the system, compressed in a vector $w_{st,i} = \begin{bmatrix} e_{st,i} & \delta_{i,ref} \end{bmatrix}^T$.

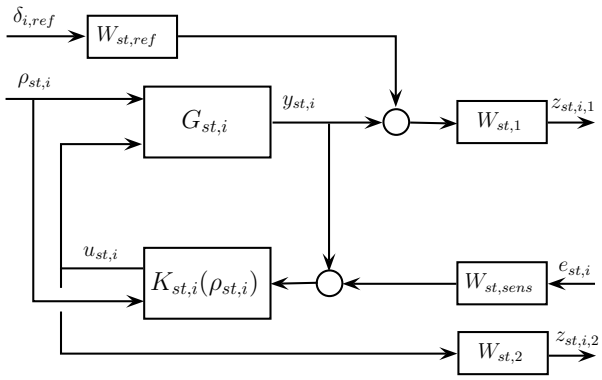


Fig. 5.3: Closed-loop interconnection structure of the steering system

The control design is based on the LPV method that uses parameter-dependent Lyapunov functions, see [83, 84]. The quadratic LPV performance problem is to choose the parameter-varying controller $K_{st,i}(\rho_{st,i})$ in such a way that the resulting closed-loop system is quadratically stable and the induced $\mathcal{L}_2$ norm from the disturbance and the performances is less than a predefined small value Γ_{st}. The minimization task is the following:

$$\inf_{K_{st,i}} \sup_{\rho_{st,i}\in\mathcal{F}_\mathcal{P}} \sup_{\|w_{st,i}\|_2\neq 0, w_{st,i}\in\mathcal{L}_2} \frac{\|z_{st,i}\|_2}{\|w_{st,i}\|_2}. \tag{5.25}$$

The existence of a controller that solves the quadratic LPV Γ-performance problem can be expressed as the feasibility of a set of Linear Matrix Inequalities (LMIs), which can be solved numerically. Finally, the state space representation of the LPV control $K_{st,i}(\rho_{st,i})$ is constructed, see [85, 84].

In the hierarchical structure the purpose of the steering controller is to guarantee the steering angle, which is required by the lateral controller, see Figure 5.2. Moreover, the suspension control has the same role related to the steering controller. Thus, the accuracy of the interconnected steering-suspension control system has an important role in the stability and the performance of the entire system. The aim of the following analysis is to formulate the maximum tracking error of the steering-suspension control. The result of the analysis is incorporated in the design of the lateral robust control. In this way the interconnection in the hierarchy is guaranteed.

The process of the analysis is the following. Several simulations are performed using different initial values $x_{st,i}(0)$ and $\delta_{i,ref}$. The intervals of the initial values are $x_{st,i}(0) = \pm x_{st,max}$ and $\delta_{i,ref} = \pm\delta_{max}$. The intervals of $x_{st,i}$ and δ_{ref} are gridded with ϵ_{st} and ϵ_{ref} samplings, respectively. Altogether $\left(2\frac{x_{st,max}}{\epsilon_{st}}+1\right)^2 \cdot \left(2\frac{\delta_{max}}{\epsilon_{ref}}+1\right)$ simulation scenarios are performed. In the scenarios the steering-suspension control must guarantee achieving $\delta_{i,ref}$ from the $x_{st,i}(0)$. In each case the maximum tracking error during the scenario is calculated. The results of the examinations show that the maximum of the steering error is below 0.5°. Although the probability of the

error with this value is low, for robustness reasons the upper bound of the error 0.5° is considered in the followings.

The results of the simulation-based statistical analysis are used for the modeling of uncertainties in a multiplicative form. Since in the robust control design the worst case scenario is considered, the maximum tracking error of the previous analysis is used in the formulation. Since the controllers $K_{st,i}$ and K_{susp} have inaccuracy, and the steering and the suspension have their own dynamics, there is a difference between $\delta_{i,ref}$ and δ_i. The relation between $\delta_{i,ref}$ and δ_i, using the upper bound of the previous analysis, is formulated in the transfer function $1+W_\Delta$, which represents the uncertainty on the control input signal, such as

$$W_\Delta = \frac{\alpha_{\Delta,2}s^2 + \alpha_{\Delta,1}s + \alpha_{\Delta,0}}{T_{\Delta,2}s^2 + T_{\Delta,1}s + T_{\Delta,0}}. \tag{5.26}$$

W_u scales the bound of input multiplicative uncertainty, where $\alpha_{\Delta,2}$, $\alpha_{\Delta,1}$, $\alpha_{\Delta,0}$ and $T_{\Delta,2}$, $T_{\Delta,1}$, $T_{\Delta,0}$ are design parameters. The ratio of $\alpha_{\Delta,0}/T_{\Delta,0}$ represents the low-frequency error of the steering system. Therefore, it is selected based on the previous analysis, such as $\alpha_{\Delta,0}/T_{\Delta,0} = 0.5$deg. Moreover, the ratios $\alpha_{\Delta,2}/T_{\Delta,2}$ and $\alpha_{\Delta,1}/T_{\Delta,1}$ reflects to the high-frequency error of the steering control and they have low values.

5.2.3 Integrated design of independent steering and torque vectoring control

The goal of the integrated control design is to guarantee trajectory tracking and the robust stability of the entire system through the computation of the control inputs δ_{ref} and M_d. This control has a high impact on the entire system, because it generates the inputs steering and torque vectoring. Moreover, in the design of the the lateral control the inaccuracy of the interconnected steering-suspension control system is involved. It must guarantee the stability and performances of the entire system through the robustness.

The most important performance of the lateral controller is to follow a reference lateral position y_{ref}, which is defined as

$$z_{lat,1} = y_{ref} - y, \quad |z_{lat,1}| \rightarrow min \tag{5.27}$$

where $\dot{\psi}_{ref}$ is computed using e.g. the velocity and the steering wheel angle of the driver [59]. Moreover, the performance $z_{lat,1}$ must be reached using minimum control input $u_{lat} = \begin{bmatrix}\delta_{ref} & M_d\end{bmatrix}^T$. Thus, the next two performances are defined:

$$z_{lat,2} = \begin{bmatrix}\delta_{ref}\end{bmatrix}^T, \quad |z_{lat,2}| \rightarrow min \tag{5.28a}$$

$$z_{lat,3} = \begin{bmatrix}M_d\end{bmatrix}^T, \quad |z_{lat,3}| \rightarrow min \tag{5.28b}$$

The performances are compressed in a vector $z_{lat} = \begin{bmatrix}z_{lat,1} & z_{lat,2} & z_{lat,3}\end{bmatrix}^T$.

The state-space representation of the system (5.18) is extended with the performances and the measurements for the control design:

$$\dot{x}_{lat} = A_{lat}x_{lat} + B_{lat}u_{lat} \tag{5.29a}$$
$$z_{lat} = C_{lat,1}x_{lat} + D_{11,lat}r_{lat} + D_{12,lat}u_{lat} \tag{5.29b}$$
$$y_{lat} = C_{lat,2}x_{lat} + D_{21,lat}r_{lat} \tag{5.29c}$$

where $r_{lat} = y_{ref}$ is the reference signal and the measured signal is the lateral error of the vehicle, such as $y_{lat} = y_{ref} - y$.

Although the system (5.29) is linear, a trade-off between the control inputs δ_{ref} and M_d is reached by a scheduling variable ρ_{lat}. Therefore, in Figure 5.4 the augmented plant for the LPV design is illustrated. In the architecture three performance weighting functions are used. While $W_{lat,1}$ scales the admissible error on the trajectory tracking, the weights $W_{lat,2}(\rho_{lat})$ and $W_{lat,3}$ have impact on the actuation.

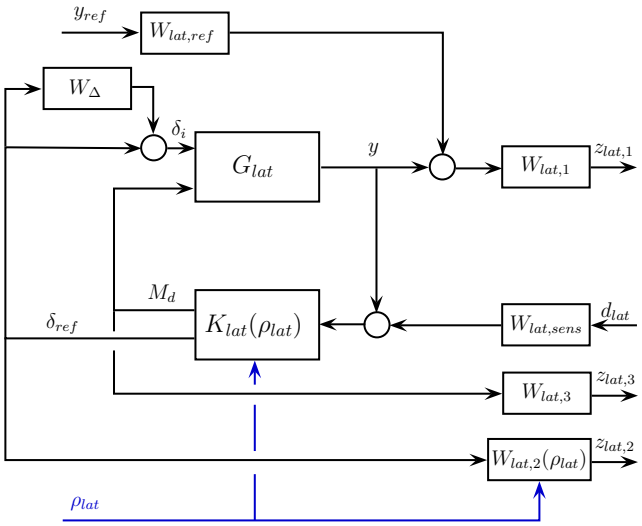


Fig. 5.4: Closed-loop interconnection structure of the lateral dynamics

The weight on the steering angle is selected as parameter-dependent, in the following form: $W_{lat,2}(\rho_{lat}) = \frac{\rho_{lat}}{A_{lat,2}}$, where $\rho_{lat} \in [\rho_{lat,min}, \rho_{lat,max}]$ is a scheduling variable of the system. The role of ρ_{lat} is to influence the actuation of the variable-geometry suspension through the steering intervention. If $\rho_{lat} = \rho_{lat,min}$, then $W_{lat,2}(\rho_{lat,min})$ has a small value, which results in the increase of δ_{ref}. Similarly, if $\rho_{lat} = \rho_{lat,max}$, then $W_{lat,2}(\rho_{lat,max})$ has a high value, which reduces the steering actuation. The weight on M_d is selected a constant value, such as $W_{lat,3} = A_{lat,3}$. Thus, the trade-off between the actuation of δ_{ref} and M_d is determined by ρ_{lat}.

In the augmented plant the weight W_Δ is incorporated, which represents the multiplicative uncertainty of the steering-suspension control, see (5.26). In the consideration of the uncertainty the steering distribution is not considered, thus

$\delta_{ref} = \delta_{i,ref}$. Moreover, $W_{lat,ref}$ scales the reference position and $W_{lat,sens}$ is the weight on the lateral error measurement. The signals d_{lat} and y_{ref} are handled as disturbances in the system, compressed in a vector $w_{lat} = \begin{bmatrix} d_{lat} & y_{ref} \end{bmatrix}^T$.

The minimization task of the high-level LPV control design is formulated as

$$\inf_{K_{lat}} \sup_{\rho_{lat} \in \mathcal{F}_{\mathcal{P}}} \sup_{\|w_{lat}\|_2 \neq 0, w_{lat} \in \mathcal{L}_2} \frac{\|z_{lat}\|_2}{\|w_{lat}\|_2}. \tag{5.30}$$

where w_{lat} disturbance vector contains the reference signal, the sensor noise and the effect of the multiplicative uncertainty.

5.3 Coordination strategy in the variable-geometry suspension

In the previous section the design of the different controllers in a hierarchical structure are proposed. Although the presented control systems guarantee the stability and the performances, the efficient coordination of the independent steering and the torque vectoring is realized by the reconfiguration strategy and the force/steering distribution blocks. This section proposes the algorithms of the blocks, which are based on the nonlinear analysis of the variable-geometry suspension system.

In the coordination four important aspects are considered.

- The longitudinal force influences the efficiency of the steering actuation significantly, see $B_{st,i}(F_{l,i})$ in (5.14). In the case of small $|F_{l,i}|$ values the camber angle has a low impact on the steering angle and the vehicle dynamics. Therefore, the vehicle motion must be influenced through torque vectoring instead of steering. The avoidance of the efficiency reduction is guaranteed by the reconfiguration strategy and the force/steering distribution.

- The left and the right wheels have own control systems, as illustrated in Figure 5.2. Since the efficiency of the left and right steering can be different through the difference of $F_{l,l}$ and $F_{l,r}$, it is possible to improve the steering actuation of the more effective side. It is realized in the force and steering distribution block.

- Furthermore, the distribution of the longitudinal forces, based on the control signal M_d is also computed in the force and steering distribution block.

- The interventions of steering and torque vectoring have different impacts on the vehicle dynamics. The aim of the coordination is to maximize the operation range of the variable-geometry suspension. Thus, an analysis is performed to find an appropriate control strategy. The results of the examination is built-in the reconfiguration strategy.

In the followings the algorithms of the reconfiguration strategy and the force/steering distribution blocks considering the previous aspects are presented.

5.3.1 Force and steering distribution

The role of the force and steering distribution block in Figure 5.2 is to compute the appropriate δ_l, δ_r and $F_{l,i}$ signals. Through the selection of the longitudinal forces it is necessary to avoid the inefficient actuation of the variable-geometry suspension due to the small $|F_{l,i}|$. Moreover, the steering angles δ_l, δ_r must be adapted to the actual $F_{l,i}$. First, the distribution of $F_{l,i}$ based on M_d is presented, and second the selection of the steering angles are proposed.

Several papers papers deal with the optimal distribution of $F_{l,i}$ as a part of the torque vectoring functionality, see e.g. [86, 87]. In the followings the aim of the distribution is to guarantee the generation of M_d. Since the selected $F_{l,i}$ also have role in the longitudinal dynamics, the sum of the longitudinal forces must be equal to the requested F_0. F_0 is an external signal from the longitudinal control system, which guarantees the longitudinal performances, for example speed tracking or vehicle following. In this study, F_0 is considered as a known command and the distribution of M_d the signal F_0 must not be changed. Thus, the longitudinal forces on either side can be calculated from the following expressions:

$$F_{l,l} = F_0 + \frac{M_d}{b_f}, \quad F_{l,r} = F_0 - \frac{M_d}{b_r} \tag{5.31}$$

where b_f is the track of the vehicle on the front axle.

The purpose of the steering distribution is to define the appropriate steering angle for each wheel. Since the achievable steering angle depends on the actual longitudinal force of the wheel, the steering distribution is a crucial part of the independent steering system. The lateral controller provides a common steering angle, which must be performed. Thus, considering small slip angles, the average of the two steering angles must be equal to the reference steering angle δ_{ref}. The following algorithm computes the desired steering angles for each wheel from the reference steering angle δ_{ref} using the deviation between the two longitudinal forces $F_{l,l}$ and $F_{l,r}$. Moreover, this algorithm guarantees the desired average steering angle.

The distribution of steering angles is based on the computation of division parameters $C_{\delta,i}$, such as

$$\delta_{i,ref} = C_{\delta,i}\delta_{ref} \tag{5.32}$$

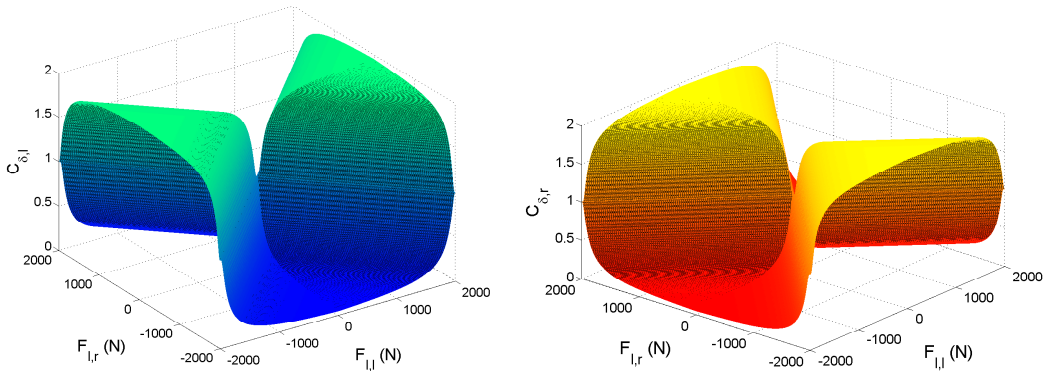
where δ_{ref} is computed by the lateral control, see Figure 5.2. The aim of $C_{\delta,i}$ is to adapt $\delta_{i,ref}$ to the actual $F_{l,i}$, which has an important role if $F_{l,l} \neq F_{l,r}$, e.g. at torque vectoring. For example at $F_{l,l} < F_{l,r}$ in the steering actuation efficiency on the left wheel is decreased. Thus, $\delta_{l,ref}$ must be reduced, while $\delta_{r,ref}$ is increased on the other wheel side simultaneously. The parameters $C_{\delta,i}$ are selected through the

following algorithm:

$$\begin{aligned}
&\text{if } |F_{l,l}| < |F_{l,r}|, \text{ then} && C_{\delta,l} = 1 - e^{-\frac{F_{lim}}{|F_{l,r}|-|F_{l,l}|}} \\
& && C_{\delta,r} = 2 - C_{\delta,l} \\
&\text{if } |F_{l,r}| < |F_{l,l}|, \text{ then} && C_{\delta,r} = 1 - e^{-\frac{F_{lim}}{|F_{l,l}|-|F_{l,r}|}} \\
& && C_{\delta,l} = 2 - C_{\delta,r} \\
&\text{if } |F_{l,l}| = |F_{l,r}|, \text{ then} && C_{\delta,l} = 1 \text{ and } C_{\delta,r} = 1
\end{aligned} \tag{5.33}$$

where F_{lim} is a tuning parameter.

Figure 5.5(a) illustrates an example on the surface of $C_{\delta,l}$ in terms of longitudinal forces $F_{l,l}$ and $F_{l,r}$. It can be seen that $C_{\delta,l}$ parameter is close to zero when $F_{l,l}$ is low and $F_{l,r}$ is high. On the other hand, if $F_{l,l}$ is high and $F_{l,r}$ is low, then $C_{\delta,l}$ parameter is close to the maximum value. Moreover, the selection of $C_{\delta,r}$ is illustrated in Figure 5.5(b). Note that if $C_{\delta,r}$ has a high value then $C_{\delta,l}$ is small. It guarantees the achievement of the computed δ_{ref}.



(a) Division parameter $C_{\delta,l}$ (b) Division parameter $C_{\delta,r}$

Fig. 5.5: Illustration of the parameter selection

5.3.2 Reconfiguration strategy

The goal of the reconfiguration strategy is to provide an appropriate selection between the independent steering and the torque vectoring functionalities. In the following a nonlinear analysis on the lateral dynamics is performed, by which the efficiency of the different interventions is examined. These results are built-in the reconfiguration strategy, together with the consideration of the impact of longitudinal forces on the steering capability. Thus, a selection strategy between the steering and the torque vectoring in the reconfiguration, based on the scheduling variable ρ_{lat} is constructed.

The analysis of the steering and torque vectoring efficiency with limited control intervention is based on the determination of the reachability. Formally, the set of the reachable states is defined in [88]. Given is a continuous-time system $\dot{x} = f(x(t)) + gu(t)$ with initial condition $x(0) = 0$. It is considered the set of reachable states with bounded inputs:

$$\mathcal{R} \triangleq \left\{ x(T) \;\middle|\; \begin{array}{c} (x(t), M_d(t), \Delta(t)) \,\big| \\ \dot{x}(t) = f(x(t)) + gM_d(t) + h\Delta(t), x(0) = 0, \\ \Delta_{min} \leq \Delta(t) \leq \Delta_{max}, \\ M_{d,min} \leq M_d(t) \leq M_{d,max}, T \geq 0 \end{array} \right\} \tag{5.34}$$

The resulted set $\mathcal{R}$ contains the states of the system, which can be reached through the bounded control inputs. Thus, it is possible to find the states, which can be reached with different actuations, such as steering and torque vectoring.

The intervention of the variable-geometry suspension depends on two dynamics, such as the generation of the steering angle (5.13) and the effect of steering/torque vectoring on the vehicle motion (5.17). Although these dynamics can be combined, it results in a system with increased number of states. Since the increase of the system complexity can be disadvantageous due to numerical reasons, the reachable sets are computed separately in the following way.

The reachability of the steering dynamics can be computed analytically. The $\delta_i, \dot{\delta}_i$ solutions of the steered wheel, which is described by (5.13), are formed as

$$\dot{\delta}_i = \frac{\varepsilon F_{l,i}}{J_{\delta,i}} t\gamma_i \tag{5.35a}$$

$$\delta_i = \frac{\varepsilon F_{l,i}}{2J_{\delta,i}} t^2\gamma_i \tag{5.35b}$$

where $F_{l,i}$ is fixed. Note that the reachability sets are computed for fixed maximal $\gamma_i = 1.5deg$ values, which means that $\dot{\gamma}_i(0) = 0$, $\ddot{\gamma}_i(0) = 0$ are considered. Moreover, the time domain is bounded to $t = T$, in which the reachability of the system $\dot{\delta}_i(T), \delta_i(T)$ is asked. Thus, (5.17) is reformulated as

$$\dot{x} = f(x) + gM_d + H(F_{l,i}, T)\gamma \tag{5.36}$$

In the followings the computation of the reachable is based on the trajectory reversing method [89]. It means that the null-controllability region of the forward-time nonlinear system is equivalent to the reachability region of the reverse-time system [90]. The reverse-time system is formed as

$$\dot{x} = -f(x) - gM_d - H(F_{l,i}, T)\gamma \tag{5.37}$$

The advantage of the method is the computation of the controllability set for polynomial systems. [78] proposes a Sum-of-Squares programming based method, by which the controllability set of the polynomial system (5.37) can be computed. In

the followings the reachable set computation method based on the trajectory reversing method is discussed briefly.

The set computation method requires the existence of a smooth, proper and positive-definite Control Lyapunov Function $V : \mathbb{R}^n \to \mathbb{R}$, which requires that $\inf_{u\in\mathbb{R}} \left\{ \frac{\partial V}{\partial x}(-f(x)) + \frac{\partial V}{\partial x}(-M)\cdot u \right\} < 0$ must be guaranteed for each $x \neq 0$, where $M = \begin{bmatrix} g & H \end{bmatrix}$ and $u = \begin{bmatrix} M_d & \gamma \end{bmatrix}^T$, respectively.

Thus, the next optimization problem is formed to find the maximum Controlled Invariant Set:

$$\max \beta \tag{5.38}$$

over SOS polynomials $s_1, s_2, s_3, s_4, s_5 \in \Sigma_n$ and polynomials $V, p_1, p_2 \in \mathcal{R}_n,\ V(0) = 0$
such that

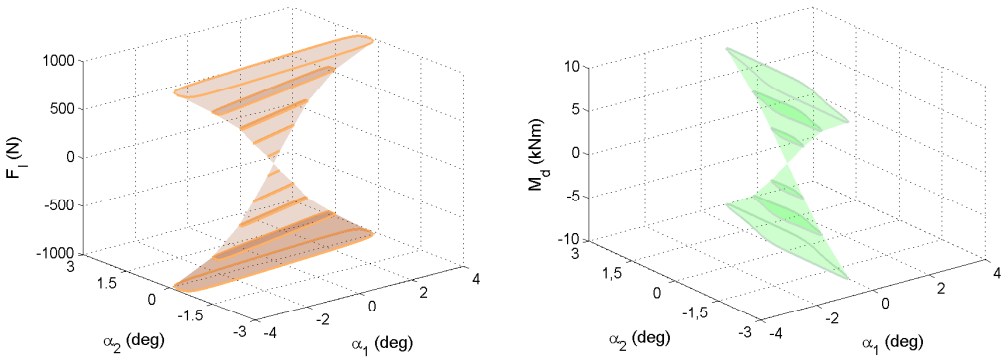
$$-\left(-\frac{\partial V}{\partial x}f(x) + \frac{\partial V}{\partial x}g\ u_{max}\right) - s_1\left(-\frac{\partial V}{\partial x}g\ -\epsilon\right) -$$
$$- s_2\,(1-V) - p_1 L_1 \in \Sigma_n \tag{5.39a}$$
$$-\left(-\frac{\partial V}{\partial x}f(x) - \frac{\partial V}{\partial x}g\ u_{max}\right) - s_3\left(\frac{\partial V}{\partial x}g\ -\epsilon\right) -$$
$$- s_4\,(1-V) - p_2 L_2 \in \Sigma_n \tag{5.39b}$$
$$-\left(s_5(\beta - p) + (V-1)\right) \in \Sigma_n \tag{5.39c}$$

where $L_{1,2}(x)$ is chosen as a positive definite polynomial, $\epsilon \in \mathbb{R}^+$ is as small as possible, $p \in \Sigma_n$ is a fixed and positive definite function and β defines a $P_\beta := \{x \in \mathbb{R}^n | p(x) \leq \beta\}$ level set.

The reachable sets of the system are computed for the actuations γ_i and M_d. In Figure 5.6(a) the reachable sets of the steering for different F_l longitudinal forces are illustrated. The maximum actuation is $\gamma = \pm 1.5°$. It can be seen that the interventions possibility of the system is influenced significantly by F_l. The results demonstrate that in case of $F_l = 0$ scenario the reachable set is zero, thus the actuation of the camber angle is ineffective. However, if F_l is increased, the reachability of the system is also improved. It means that the reconfiguration of the camber angle actuation at varying F_l can be important.

The reachable sets of the torque vectoring intervention for different maximum M_d torque values are shown in Figure 5.6(b). The results show that the shape of the sets is different from the reachability domain of the variable-geometry suspension. The differences of the sets lead to the possibility of the reconfiguration between the wheel tilting actuation and the torque vectoring.

Furthermore, the reachability analysis is examined on the integrated actuation of wheel tilting and torque vectoring, see Figure 5.7. In this case the maximum camber angle $\gamma = \pm 1.5°$ and $M_d = 9000Nm$ are applied on the vehicle. The results show that the reachable sets of the system with a coordinated actuation can be



(a) Reachable sets of the steering (b) Reachable sets of the torque vectoring

Fig. 5.6: Reachable sets of the steering

significantly increased. In case of the integrated intervention the efficiency of the vehicle control system at all F_l values can be guaranteed.

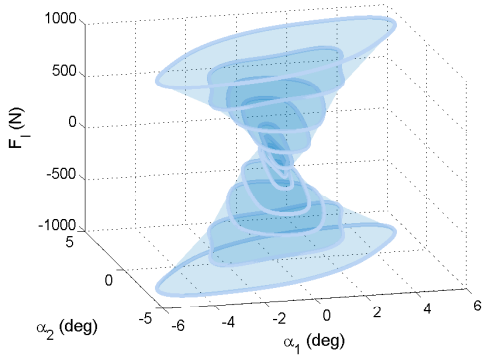


Fig. 5.7: Reachable sets of the integrated actuation

The proposed analysis has two main consequences on the reconfiguration strategy. First, the interventions of wheel tilting and torque vectoring have different effects on the wheel slips. Tilting has significant impact on α_f, while the torque vectoring influences α_r. Second, the impact of γ depends significantly on the longitudinal force on the wheels. Thus, $F_{l,i}$ influences the reachability domain significantly.

The analysis resulted in the exact sets of the reachability domain, which have been used to examine the impacts of the some factors on the vehicle dynamics. However, in the practical applications the set are not recommended to use directly, because the vehicle parameters during the vehicle motion can be varied, e.g. tyre characteristics F_i. Furthermore, the on-line computation of the sets, depending on the actual vehicle parameters can be numerically difficult. Thus, in the design reconfiguration strategy the previously formed two consequences are built-in. It leads to the following rules of the reconfiguration.

- Since the torque vectoring actuation can have relevant power loss [91], the steering intervention is more preferred. Thus, for economy reasons the steering with $\rho_{lat,min}$ is actuated in normal cruising.

- If the the tyre slip α_r is increased, the pure steering intervention is not efficient, see Figure 5.6. Therefore, at the increase of α_r the scheduling variable ρ_{lat} is also increased. It results in the enhanced intervention capability of the system through the integrated actuation, see Figure 5.7.

- Since the longitudinal force has a high impact on the reachability of the variable-geometry suspension based steering, at the reduction of $F_{l,i}$ the system is reconfigured to the torque vectoring. Thus, the decrease of $F_{l,i}$ leads to the increase of ρ_{lat}.

The reconfiguration rules are formulated in a function, such as

$$\rho_{lat} = \mathfrak{f}(\alpha_r, F_{l,1}, F_{l,2}) \tag{5.40}$$

Based on the previous rules, the form of the function $\mathfrak{f}$ is illustrated in Figure 5.8. The scheduling variable can be varied between ρ_{min}, where steering has priority and ρ_{max}, which is related to the torque vectoring. If α_r has a high value then the integration through $\rho_{lat,int}$ is preferred, and if $F_{l,i}$ is decreased then ρ_{lat} is increased. The variation of ρ_{lat} is based on $\mathfrak{f}$, of which shape has been set by the design parameters $p_{1,2,3,4}$. The role of the sections between p_1, p_2 and p_3, p_4 are to avoid the chattering in the control inputs.

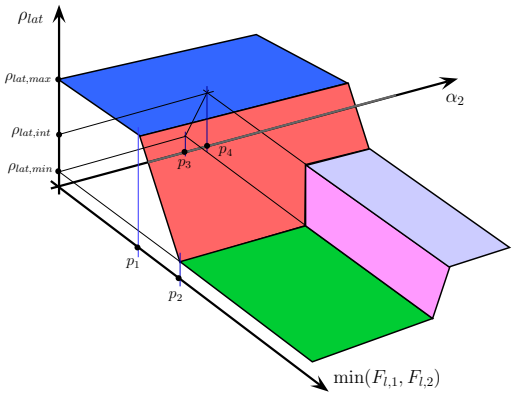


Fig. 5.8: Selection of ρ_{lat} value based on $\mathfrak{f}$

The reconfiguration strategy requires the $F_{l,i}$ values, which are delivered by the force distribution block, see Figure 5.2. Moreover, the estimation of α_r is requested, of which can be found e.g. in [92] with the measurement of longitudinal and lateral acceleration, yaw-rate, steering wheel angle, and wheel angular velocities.

5.4 Simulation examples

In the following a simulation example of the proposed control system is presented. The aim of this section is to show the efficiency and the operation of the integrated steering system. The simulation is performed through a high-fidelity simulation software CarSim. In the scenario a D-class car passenger is used, whose mass is 1690 kg. The vehicle is driven along the course of the Melbourne Grand Prix, see Figure 5.9(a).

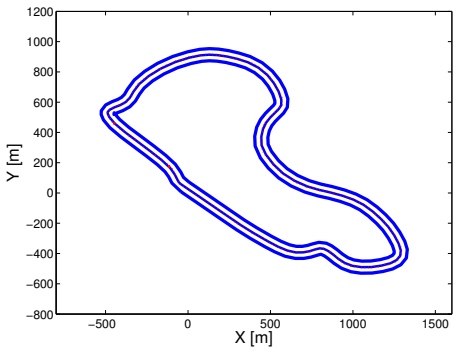
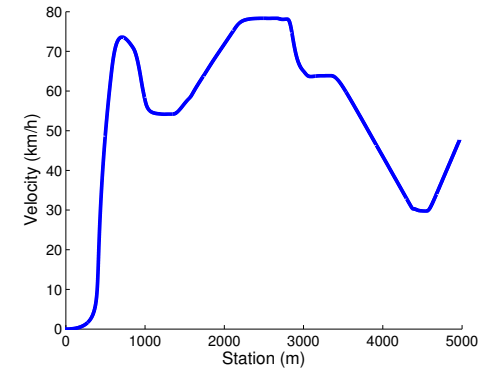


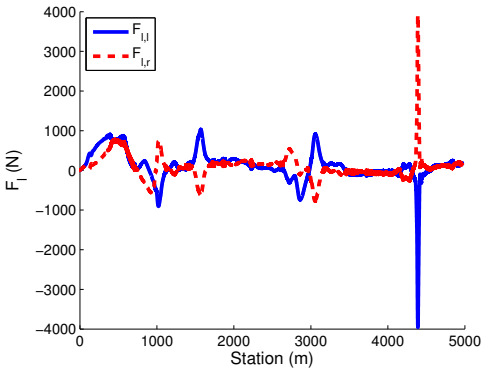
Fig. 5.9: Course and velocity of the vehicle

The track is around 5000 m long and contains several bends, therefore the velocity of the vehicle also varies, as shown in Figure 5.10(a). It results in the variation of the longitudinal forces of the wheels, see Figure 5.10(b). The tracking of the trajectory is sufficiently accurate, the maximum of the lateral error is about 0.18 m as illustrated in Figure 5.10(c). This lateral error is acceptable in the functionality of trajectory tracking.

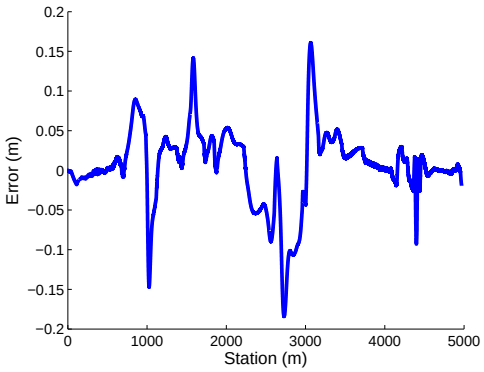
The actuation of the steering and the differential yaw-moment are shown in Figure 5.10(d) and Figure 5.10(e), respectively. The left camber angle, which realizes the left steering angle together with $F_{l,l}$, is illustrated in Figure 5.11(a). The intervention of the left suspension can be seen in Figure 5.10(b). This signal is provided by the low level controller and M_{act} signal varies between the physical limits $-300Nm \dots 300Nm$ during the simulation. Although the actuator model in this simulation is not considered, a hydraulic actuator can be used for this task, see the results of the work by [FNG17]. The scheduling variable ρ_{lat} is demonstrated in Figure 5.11(c). The purpose of this variable is to guarantee the reconfiguration strategy of the interventions. It can be seen between the section $2900m \dots 3000m$ that the longitudinal forces have relatively high values, which increase the influence of γ_i on the steering. However, α_r also has a high value, which indicates that the variable-geometry suspension based steering system is not able to ensure the accurate trajectory tracking by itself. Therefore, in this section the steering is reconfigured from torque vectoring to the integrated solution, which means that the value of ρ_{lat} is close to 0.5. The lateral position tracking of the vehicle is guaranteed



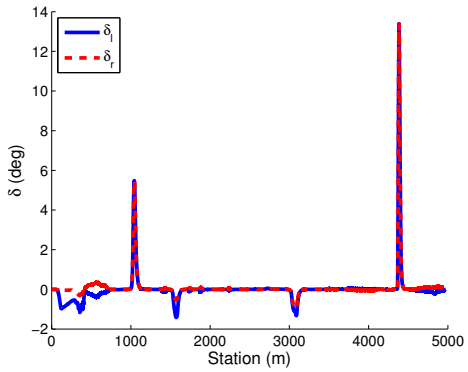
(a) Velocity



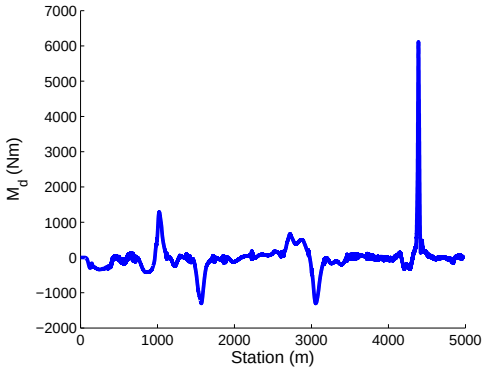
(b) Longitudinal wheel forces



(c) Lateral error



(d) Steering angle

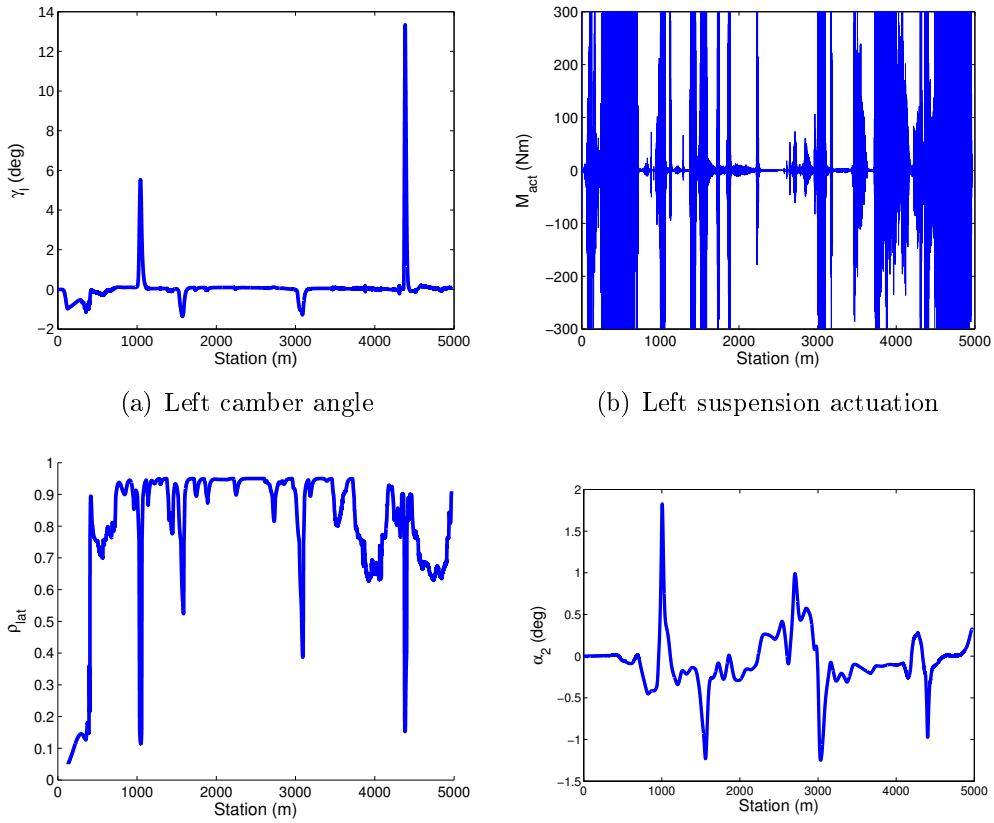


(e) Differential yaw moment

Fig. 5.10: Signals of the trajectory tracking

by this reconfiguration.

Summarizing the results of the simulation, the proposed controller is able to fulfill the predefined performances. The coordination of the variable-geometry suspension and the torque vectoring guarantees the achieving low values for the trajectory tracking. Moreover, the controls of the steering and the wheel tilting actuation are



(a) Left camber angle

(b) Left suspension actuation

(c) Scheduling variable

(d) Slip angle

Fig. 5.11: Scheduling variable and slip angle

acceptable in the hierarchical structure.

Thesis 4 I have developed a novel physical model for a steering concept based on a variable-geometry suspension system. Using the model, I have designed an integrated steering control system that combines the variable-geometry suspension with the benefits of torque vectoring-based steering method. The proposed method guarantees the stability and the controllability of the vehicle in critical situations. I have developed a coordination strategy, which can be used to select the appropriate intervention during the operation of the vehicle.

Related publications: [NFGB19, FNG18c, FNG18b, FNG17, NGFB17b, NGFB17a, FNG16a, FNG16b, NFGB16, NGFB17b, NFG16a]

6. DATA-DRIVEN CONTROL FOR A VARIABLE-GEOMETRY SUSPENION-BASED TEST-BED

In order to validate the concept of the variable-geometry suspension-based steering system, presented in Chapter 5, a test-bed has been built in Systems and Control Lab, Institute for Computer Science and Control (SZTAKI). Although the basic structure is similar to the original idea, some modifications have been carried out during the planning and building phases. Due to this modification in the structure of the testbed, the control system also needs some improvement before the implementation. Therefore, a improved control algorithm has become necessary, which is specifically designed for this suspension system and setup. In this chapter, a data-driven modeling and control algorithm is presented for the suspension system and implemented on it. This study consists of the following parts:

- Firstly, the construction and the main components of the suspension test bed are presented in Section 6.1.
- Then, a two-layered control strategy is detailed in Section 6.2. The low-level layer is responsible for controlling the test-bed, which is based on a data-driven modeling method. Whilst, the upper-level layer controls the lateral dynamics of the vehicle, and computes the reference signal for the low-level controller.
- Finally, the presented algorithm is implemented on the test-bed and validated through a Hardware-in-the-Loop (HiL) simulation using the test-bed and the CarMaker simulation software.

6.1 Construction of the variable-geometry suspension test bed

In this section the construction of the variable-geometry suspension test bed with the most important units is introduced. The construction follows the concept, which is introduced in Section 5.1.2, see Figure 5.1. Nevertheless, in the practical application the location of the actuator is modified for constructional reasons, i.e. in the original concept the actuator is placed on the hub of the suspension, while in the test bed the actuator is linked to the frame of the test bed, see Figure 6.1.

- **Linear actuator** (1) is responsible for the modification of the camber angle. The type of the actuator: LATT 4A 1/12, its supply voltage is 12V and its stroke is 165mm. The linear actuator is also equipped with a hall sensor-based encoder, which is used to determine the current position of the actuator.

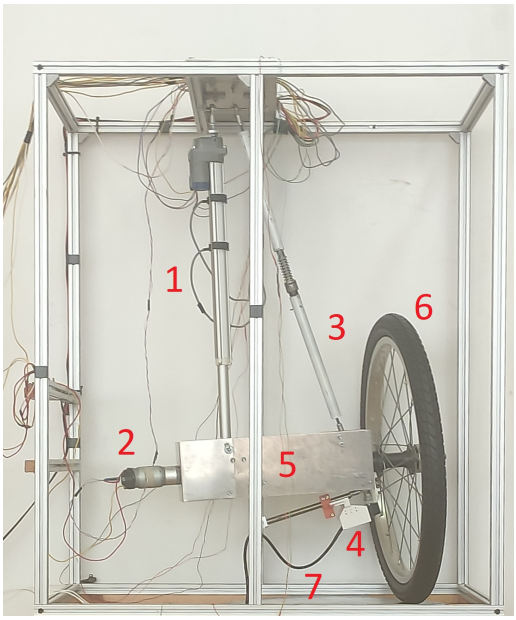


Fig. 6.1: Construction of the test bed

The actuator has its own electronic board, which controls its position, at a maximum speed of 37mm/s. In the present construction the speed of the stroke is constant in both directions. This limitation on actuation is considered in the design of the low level controller.

- **DC motor** (2), which is used to drive the wheel. The type of the motor is Actobotics 638280, which has a planetary gearbox to ensure the adequate torque for the wheel. The supply voltage of this part is 6-12V, maximum speed is 313 RPM (Rotation Per Minute) and maximum load is 3Nm. Furthermore, the gear ratio is 121/3249. The motor is equipped with an encoder on the motor shaft whose resolution is 48 Countable Events Per Revolution.

- The **upper suspension arm** (3) is an essential part of the MacPherson strut. The spring on it ensures the appropriate vertical load for generating the required longitudinal force. The upper arm is linked to the frame through a three-dimensional joint and it also has a connection to the hub of the suspension.

- The **lower suspension arm** (4) is also a crucial part of the MacPherson suspension system. It is linked to the frame through a one-dimensional joint. The other side of the arm is linked to the hub using a three-dimensional joint. On the lower arm, another encoder is found, which is used to determine the steering angle. It actually measures the rotation of the axis of the joint. This encoder works with 5V current, and its resolution is 3-480LPI (Lines Per Inch).

- The **hub** (5) contains the bearing of the shaft. The left side of the shaft is linked to the motor (2) through a clutch, which can compensate for the eccentricity in the connection of the shafts. On the other side, the shaft is coupled to the wheel. Moreover, the hub has connections to the the linear actuator and both to the upper and lower arms.

- The **wheel** (6) of the test bed has diameter of 18", which is generally used for bicycles. The wheel is rigidly linked to the shaft of the hub, which means that it rotates at the same speed as the motor. The wheel can be rotated in both forward and reverse directions.

- The **rotating plate** (7) is the lower part of the suspension test bed, which can rotate around the vertical axis. Its purpose is to guarantee the rotating motion and the limited displacement of the wheel, which are generated by the driving and steering effects.

6.2 Control design for the variable-geometry suspension testbed

The hierarchical structure of the proposed control system is illustrated in Figure 6.2. The low level of the structure contains two controllers, i.e. steering and driving controllers. The role of the driving controller on the low level is to realize the required longitudinal velocity $v_{x,ref}$. It is carried out by a PID controller, whose design method is out of the focus of this study. However, the applied design method can be found in [93, 94]. The steering controller on the low level is responsible for the realization of the desired steering angle δ_{ref}. These controllers are implemented on an AutoBox device. Meanwhile, the high level controller computes the reference steering angle δ_{ref} using the processed measurements and the computed errors (y_e, ψ_e) from the CarMaker software.

6.2.1 Design of steering controller on the low level

The design of the steering controller on the low level requires the modeling of the steering dynamics in the variable-geometry suspension systems. Although the dynamics of the original concept can be described by simplified equations, see (5.14), the dynamics of the test bed contains significant amount of nonlinearities. These nonlinearities are mainly caused by the geometrical placement of the linear actuator in the construction of the test bed. Figure 6.3 shows a test scenario, when the actuator moves within a predefined range. As it can be seen, when $\delta \in \{-6° \dots 0°\}$, the actuator moves between $d \in \{0 \dots 300\}$, while in the range of $\delta \in \{0° \dots 6°\}$, the position moves between $d \in \{300 \dots 500\}$. Furthermore, there is another effect, which highly influences the dynamics of the test bed, such as the speed of the wheel. Its reason is that the speed of the wheel significantly correlates to the longitudinal force, which appears at the tire-road contact.

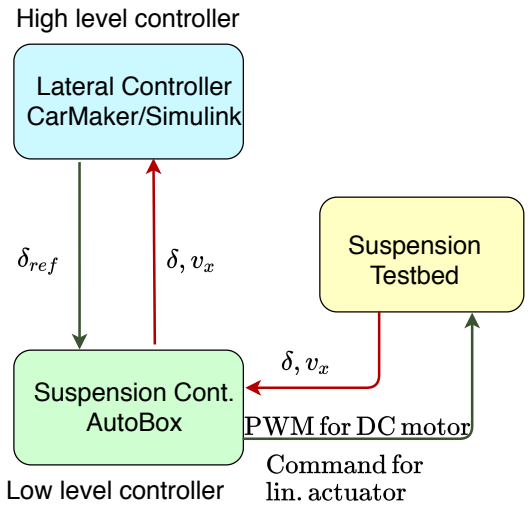


Fig. 6.2: Structure of the hierarchical control system

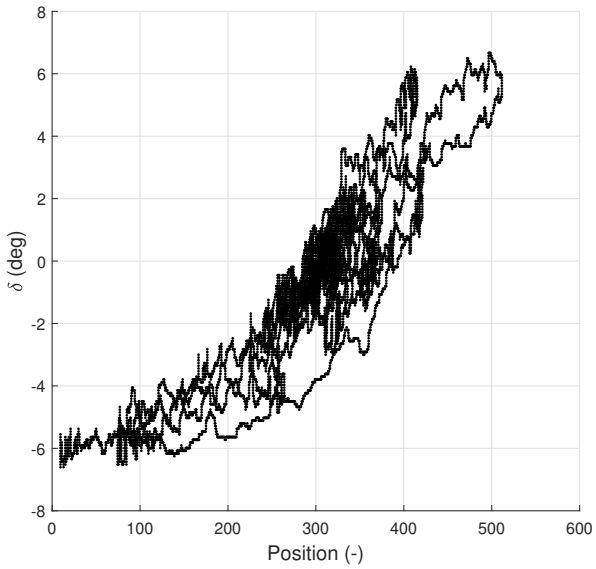


Fig. 6.3: Relation between the position of the actuator and the steering angle

The design of the steering control requires the identification of the model of the real suspension test platform. Thus, in the first step a novel machine-learning-based identification method is presented, which results in a set of polytopic models creating a gridded LPV system. In the second step the achieved control-oriented model is used for the design of the steering controller.

6.2.2 Modeling the steering dynamics of the test platform

The modeling of the steering dynamics is based on the machine-learning-based pace regression method to provide an LPV-based form. Since each of the machine-learning algorithms requires a lot of data to create appropriate models, several test scenarios have been performed on the test bed. During the tests, the speed of the wheel varied between $300 - 900RPM$ and the actuator follows different motion profiles. In this way more than 500.000 distinct instances have been collected, which consisted of several variables, e.g. speed of the wheel, position of the actuator, steering angle. The identification process consists of the following steps:

- First, the collected dataset is divided into subsets according to the measured steering angle. In the given system the step size for the division is selected to $2deg$.

- Second, in order to boost the performance of the modeling process, differences of the steering angle and the position of the actuator are computed for each instances, such as:

$$\Delta\delta = \delta(t) - \delta(t-1) \tag{6.1a}$$
$$\Delta d = d(t) - d(t-1). \tag{6.1b}$$

  The motivation of the transformation is that the correlation between $\Delta\delta$ and Δd is more relevant on the process than the absolute values of the steering angle and the actuator position. Its reason is that the absolute values can depend on the initial values in the measurement, which can varied in the setting of the test bed.

- Third, the relation between the actual $\Delta\delta(t)$ and the attributes of the system is formed. The relationship contains the actual values of $\Delta d(t)$ and wheel speed $v_x(t)$ and the past values of $\Delta d, \Delta\delta, v_x$. The form of the model for the computation of $\Delta\delta$ is described as

$$\hat{y}_{\mathcal{M}_j} = X_{\mathcal{M}_j}\zeta_{\mathcal{M}_j}, \tag{6.2}$$

  where $\hat{y}_{\mathcal{M}_j} = \Delta\delta(t)$, and $\zeta_{\mathcal{M}_j}$ contains the coefficients of the applied variables: $\Delta\delta(t-1)\dots\Delta\delta(t-n), \Delta d(t)\dots\Delta(t-n), v_x(t)\dots v_x(t-n)$. Vector $X_{\mathcal{M}_j}$ contains given actual values of the variables. In the given test bed n is set to 5. Note that (6.2) must be formed for each subsets. The vectors X of each subsets are considered to be independent from each other.

- The fourth step is the selection of the coefficients in X for each subsets, which leads to an optimization process. The main goal of the optimization process is to find the best linear model, which minimizes the deviation between the estimated and the measured outputs. The minimization task can be formulated

as

$$\min_{X} \left(\hat{y} - y\right)^2, \tag{6.3}$$

where y represents the real measurements $\Delta\delta(t)$ on the test bed. The process of the optimization is performed through the pace regression method, which is applied to find appropriate linear models for each subsets [55].

The resulted models in the form of (6.2) are evaluated by the k-fold cross validation technique. In each subsets the pace regression method resulted in a linear model, whose correlation coefficient for the given example is above > 0.985.

The resulted linear models are transformed into a discrete state-space representation for control design purposes. The state-space description for each subsets contains the relations

$$\hat{y}_{\mathcal{M}_j} = X_{\mathcal{M}_j}\zeta_{\mathcal{M}_j}, \tag{6.4a}$$

$$\delta(t) = \sum_{T=t-n}^{0} \Delta\delta(T). \tag{6.4b}$$

The state-space representation of the model, which incorporates in all subsets, is formed as:

$$x_s(t+1) = A_s(\rho)x_s(t) + B_{s1}(\rho)u_s(t) + B_{s2}(\rho)\omega_s(t) \tag{6.5}$$

where $A_s, B_{s1}(\rho), B_{s2}(\rho)$ are matrices and vectors. The control input variable u_s is Δd, the state vector x_s consists of $\Delta\delta$, Δd and their past values, and the last state is the $\delta(t)$. In the formulation the wheel speed is an independent variable from the longitudinal dynamics, and thus, it is handled as an external disturbance of the system: ω_s contains v_x and their past values. The scheduling parameter ρ represents that (6.5) contains all subsets. The value of reflects to each subsets, which depends on the steering angle δ. It means that the scheduling variable of (6.5) depends on a state, but the relationship is hidden by the selection of each subsets and it is not formed mathematically. The resulted system is a polytopic LPV model, whose elements are represented by the models from the subsets.

Figure 6.4 shows an example on the resulted LPV model. As it can be seen the output of identified model is close to the measured steering signal. Its mean error is smaller than $< 0.24deg$, which means that the model is able to cover the dynamics of the test bed.

6.2.3 LPV-based control design for steering functionality

The role of the control design on the low level is to provide steering functionality with low error. Thus, the predefined performances, which must be guaranteed by the controller, are:

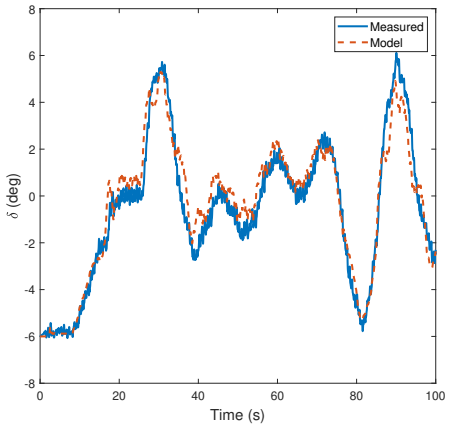


Fig. 6.4: Evaluation of the resulted LPV model based on the comparison of simulation and measurement

- The main goal of the LPV control design is to guarantee the accurate tracking of the reference steering angle (δ_{ref}), which can be formulated as the minimization of the error signal:

$$z_1 = \delta_{ref} - \delta, \qquad |z_1| \to min, \tag{6.6}$$

- Since the system has its own limitations, the minimization of the intervention (position of the linear actuator) must be also guaranteed:

$$z_2 = d, \qquad |z_2| \to min. \tag{6.7}$$

The presented performances are written into a vector $z = \begin{bmatrix} z_1 & z_2 \end{bmatrix}^T$, which leads a performance equation:

$$z = C_1 x_s + D_{11} r_s + D_{12} u_s, \tag{6.8}$$

where C_1, D_{11}, D_{12} are matrices and r_s contains the signal δ_{ref}.

The previously identified state-space representation (6.5) is extended with the performance equation and it is written to a continuous form as:

$$\dot{x}_s = A_s(\rho)x_s + B_{s1}(\rho)u_s + B_{s2}(\rho)\omega_s \tag{6.9a}$$
$$z = C_1 x_s + D_{11} r_s + D_{12} u_s, \tag{6.9b}$$
$$y_K = C_2 x_s, \tag{6.9c}$$

where $y_K = \begin{bmatrix} \delta \end{bmatrix}$.

In order to guarantee the predefined performances, several weighting functions are used in the control design. $W_{z,1}$ aims to guarantee the accurate tracking of the reference signal. The weighting function $W_{ref,1}$ is to scale the reference signal from

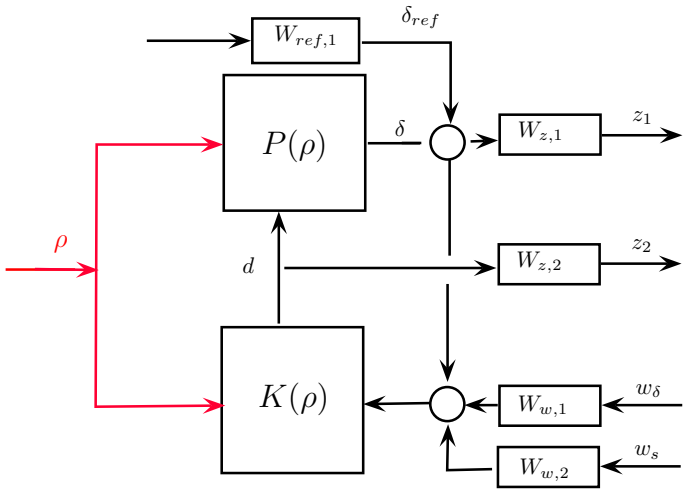


Fig. 6.5: Augmented plant for LPV control design

the high level controller. $W_{z,2}$ is used to scale the intervention. Furthermore, $W_{w,1}$ is to attenuate the noises and $W_{w,2}$ is used to compensate the effect of the speed v_x.

The quadratic LPV performance problem is to choose the parameter-varying controller $K(\rho)$ in such a way that the resulting closed-loop system is quadratically stable and the induced $\mathcal{L}_2$ norm from disturbance and to performances is less than the value γ. The minimization task is the following:

$$\inf_{K(\rho)} \sup_{\rho \in F_\rho} \sup_{\|w\|_2 \neq 0, w \in \mathcal{L}_2} \frac{\|z\|_2}{\|w\|_2}, \tag{6.10}$$

where F_ρ bounds the scheduling variables. The yielded controller $K(\rho)$ is formed as

$$\dot{x}_K = A_K(\rho)x_K + B_K(\rho)y_K, \tag{6.11a}$$
$$u = C_K(\rho)x_K + D_K(\rho)y_K, \tag{6.11b}$$

where $A_K(\rho), B_K(\rho), C_K(\rho), D_K(\rho)$ are variable-dependent matrices.

Finally, the presented LPV controller computes the reference position difference for the linear actuator using the sampling time $T_{s,1} = 0.01s$. However, the electronic board of the linear actuator has only three dedicated pins for controlling the position of the linear actuator: up, down and stop. Therefore, the reference position signal is transmitted to the Autobox device, which activates the corresponding command on the electronic board of the actuator and stops it when the actuator reaches the required position. In order to guarantee the accurate tracking of the position a faster sampling time is use : $T_{s,2} = 0.0001s$.

6.2.4 Design of path following controller on the high level

The same lateral model is used in this case as presented in Section 2.2:

$$mv_x(\dot{\psi} + \dot{\beta}) = C_1\alpha_f + C_2\alpha_r, \tag{6.12a}$$

$$I_z\ddot{\psi} = C_1\alpha_f l_1 - C_2\alpha_r l_2, \tag{6.12b}$$

$$\dot{v}_y = v_x(\dot{\psi} + \dot{\beta}), \tag{6.12c}$$

Using these equations the following state-space can be built up:

$$\dot{x}_v = A_v x_v + B_v u_v, \tag{6.13}$$

where u_v consists of the steering angle and the state-vector is: $x_v = \begin{bmatrix}\beta & \dot{\psi} & \psi & v_y & y\end{bmatrix}$ and A_v, B_v are system matrices. ψ and y are computed by integrating their derivatives: $\dot{\psi}$ and v_y.

The goal of the high level control is to guarantee the path tracking of a vehicle on a road segment through automated steering. In the field of steering control design there are several approaches which can be applied. For example, in the previous chapter, a Linear Parameter-Varying (LPV) based method for variable-geometry suspension based steering control design is proposed. In the selection of the control method it is necessary to consider the constraint regarding the path following and the specifications of the test bed.

- In the design of the high level control the edges of the road, as the constraints on the path must be incorporated.

- The generation of the steering angle has time requirements, i.e. the low-level control and the motion of the suspension result in a delay in the system.

For these reasons a Model Predictive Control (MPC) algorithm is developed for the control design [95, 96]. In the MPC design problem the constraints can be incorporated. Furthermore, the impact of the delay on the low level can be reduced through the preliminary knowledge on the reference path.

The MPC method requires a discrete-time state-space representation of the model. Therefore the presented state-space model is converted to a discrete one using the sample time T_s. Then, the discrete-time state-space representation can be formulated as:

$$x_v(k+1) = \phi_v x_v(k) + \Gamma_v u_v(k), \tag{6.14}$$

where ϕ_v and Γ_v are the matrices of the disceretized system.

The prediction of the motion of the vehicle must be performed for the horizon n, which can be computed as, see [97]:

$$z_{pred}(k,n) = \begin{bmatrix} z(k+1) \\ z(k+2) \\ \vdots \\ z(k+n) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}^T \begin{bmatrix} \phi_v \\ \phi_v^2 \\ \vdots \\ \phi_v^n \end{bmatrix} x_v(k) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}^T \begin{bmatrix} \Gamma_v & 0 & \cdots & 0 \\ \phi_v\Gamma_v & \Gamma_v & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \phi_v^{n-1}\Gamma_v & \phi_v\Gamma_v & \cdots & \Gamma_v \end{bmatrix} \begin{bmatrix} u_v(k) \\ u_v(k+1) \\ \vdots \\ u_v(k+n-1) \end{bmatrix}. \tag{6.15}$$

The goal of the control design is to guarantee the trajectory tracking of the vehicle, which consists of two main components: the tracking of the lateral position and the tracking of the yaw-angle of the road. The errors of tracking can be expressed as:

$$e(k,n) = z_{ref}(k,n) - z_{pred}(k,n), \tag{6.16}$$

where $e(k,n)$ is a vector, which contains both error signals.

Using this error vector, the following cost function can be determined, which must be minimized in order to guarantee the trajectory tracking of the vehicle.

$$J = \frac{1}{2} e(k,n)^T Q e(k,n) + U(k,n)^T R U(k,n), \tag{6.17}$$

where $U(k,n) = \begin{bmatrix} u_v(k) & \dots & u_v(k+n-1) \end{bmatrix}^T$. Moreover, Q and R are weighting matrices, which guarantee a balance between tracking performance and control actuation (steering angle).

Using (6.15) and (6.16) the cost function can be transformed to

$$J = U(k,n)^T \sigma U(k,n) + \nu^T U(k,n), \tag{6.18}$$

where σ and ν are matrices.

Finally, the following quadratic optimization task must be solved to obtain the optimal control input sequence.

$$\min_{U(k,n)} U(k,n)^T \sigma U(k,n) + \nu^T U(k,n). \tag{6.19}$$

$$s.t. \begin{cases} B_b < H_{in} U < B_u \\ l_b \leq u_i \leq l_u \end{cases} \tag{6.20}$$

However, the vehicle and the steering system have their own bounds, therefore the minimization problem is subject to constraints. B_b and B_u are constraints of

the states of the system, such as the edges of the road and the limitations of the yaw-rate signal. In the meantime, l_i guarantees that the computed control input does not exceed the limitations of the steering system. The matrix H_{in} is formed as:

$$H_{in} = \begin{bmatrix} \Gamma_v & 0 & \cdots & 0 \\ \phi_v \Gamma_v & \Gamma_v & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \phi_v^{n-1} \Gamma_v & \phi_v^{n-2} \Gamma_v & \cdots & \Gamma_v \end{bmatrix} \tag{6.21}$$

The vectors B_b and B_u are:

$$B_b = \begin{bmatrix} x_{v,low}(t+1) \\ \vdots \\ x_{v,low}(t+n-1) \end{bmatrix} B_u = \begin{bmatrix} x_{v,up}(t+1) \\ \vdots \\ x_{v,up}(t+n-1) \end{bmatrix} \tag{6.22}$$

where $x_{v,low}(T)$ denotes the lower limits of the states at the time step T and $x_{v,up}(T)$ denotes the upper limits of the states at T.

The computed vector of the input signals U contains the requested control inputs on the horizon ahead of the vehicle. In the implementation of the controller the first element u_v of U is applied. It represents the steering angle, which must be actuated so that the vehicle follows the path. Nevertheless, u_v cannot be achieved directly, so that the steering angle must be generated through the modification of the scrub radius. Therefore, in the hierarchical structure of the control system u_v is a reference signal for the low level control, such as

$$\delta_{ref} = u_v. \tag{6.23}$$

6.3 Simulation example on Hardware-In-the-Loop test platform

The effectiveness of the designed hierarchical control through HiL simulation is demonstrated. First, the HiL simulation setup is presented and second, the results of the simulation are shown. The setup of the Hil simulation is illustrated in Figure 6.6. The main components of the setup are the following.

The **computer** has two features in the HIL simulation. First, it contains the CarMaker and the Matlab/Simulink simulation environments. The dynamics of the whole vehicle is modeled by the high-fidelity simulation sotfware, CarMaker. The main purpose of the CarMaker software is to model lateral, vertical and longitudinal dynamices of the vehicle excluding the steering dynamics, which is realized in the test bed. CarMaker receives the processed measured signals: longitudinal velocity (v_x) and the steering angle δ. Using these measurements, the software computes the errors (y_e, ψ_e) of the lateral position and the yaw angle from the current position of the vehicle and the predefined road. Second, the computer contains the high

level control, which is also implemented in the Matlab/Simulink environment. It computes the reference signals for longitudinal velocity $v_{x,ref}$ and the steering angle δ_{ref}. All the signals (received and computed) are transmitted through a serial communication protocol from/to the vector device.

The **vector interface** is responsible for transmitting the measurements from the sensors towards the computer, on which the CarMaker and the Simulink run. Vice versa, the computed reference signals are also transmitted through this device to the next device, which is the AutoBox.

The **AutoBox II** is an essential part of the setup. It serves several purposes: It is responsible for processing the raw signals from the sensors placed on the suspension system (the position of the actuator, the velocity of the wheel, and the realized road wheel angle). The lower level controller is also implemented on this device. The lower level controller computes the PWM signal for the DC motor and the command for the linear actuator. The AutoBox also transmits the measured and processed signals to the Vector device through CAN communication. This device also receives the reference signals from the Vector device.

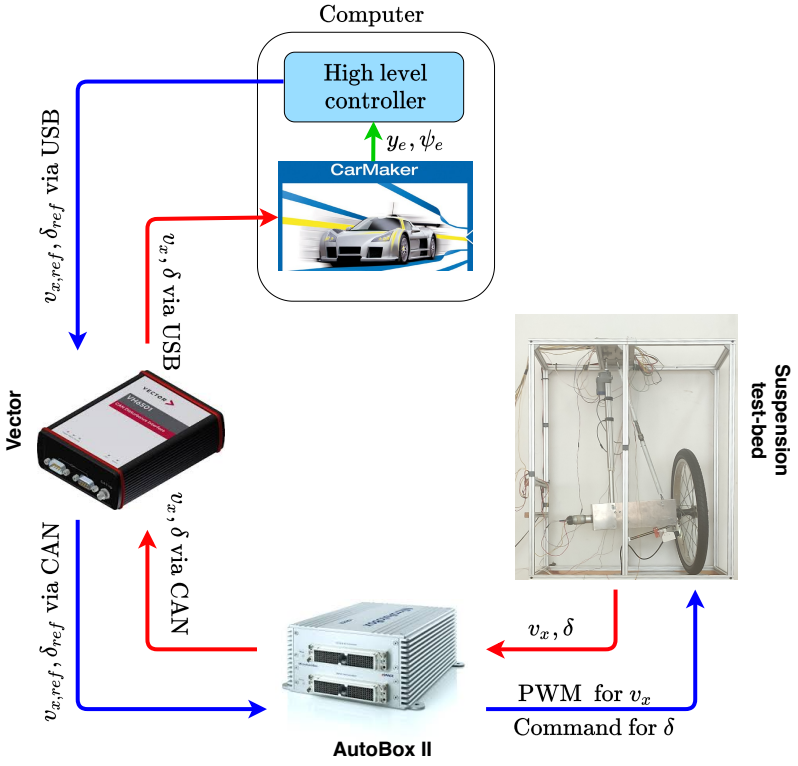
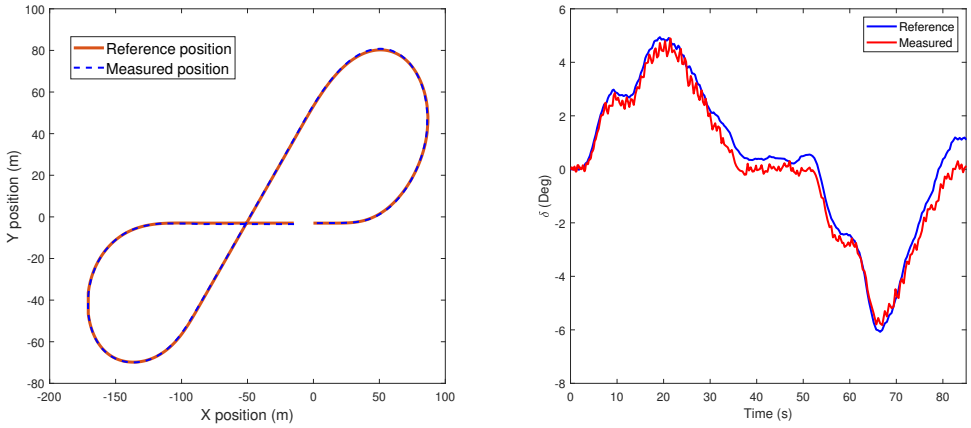


Fig. 6.6: Hardware in the loop setup

In the rest of this section a comprehensive HiL test scenario is presented, which aims to show the efficiency of the proposed control strategy and the operation of the

test bed. In CarMaker, a D-Class passenger car is chosen, whose mass is $1538kg$. The vehicle is driven along a track, which consists of two bends (one left and one right), see Figure 6.7(a). Furthermore, the speed of the car is set to a constant value. The high level controller computes the reference steering angle, while the low level controller realizes the reference value on the test bed.

In Figure 6.7(a), the path of the vehicle and the reference trajectory can be seen. As the figure shows, the proposed control system is able to guarantee the accurate trajectory tracking of the vehicle. Figure 6.7(b) illustrates the reference steering angle, computed by the high level controller, and the measured steering angle from the test bed. The error between the two signals is small, the maximal deviation is below $< 0.4deg$. which means that the test bed and the proposed LPV controller are able to ensure the accurate trajectory tracking of the vehicle.

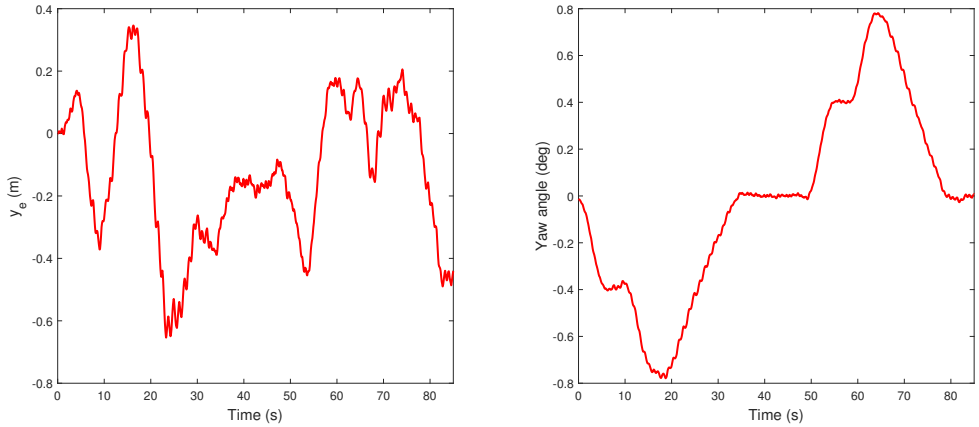


(a) Path of the vehicle (b) Steering angle

Fig. 6.7: Path and steering angle of the vehicle

Figures 6.8 demonstrates the error signals. The lateral error can be seen in Figure 6.8(a). The maximum of the lateral error can be considered to be large $0.6m$, however the capacity of the linear actuator does not allow better performance. The next figure sows the error of the yaw angle. This error is small, its maximal value is below $< 0.8deg$.

Finally, the last figures illustrate the speed and the yaw rate of the vehicle. Figure 6.9(a) demonstrates the reference and the measured velocities. As it can be seen, the tracking of the reference signal is accurate, however small oscillation can be observed, which is caused by the structure of the test bed. Figure 6.9(b) depicts the measured yaw rate. As the figure shows, the maximal yaw rate value is below $< 0.3rad/s$, which is an acceptable value for a common passenger car. Figure 6.9(c) illustrates the intervention of the linear actuator. It can be seen that the control signal on the test bed can be realized. Although the control signal in a previous simulation (see Figure 5.11(b)) requires rapid changes against the actuator, in practical application with the consideration of the actuator properties these effects can be eliminated.

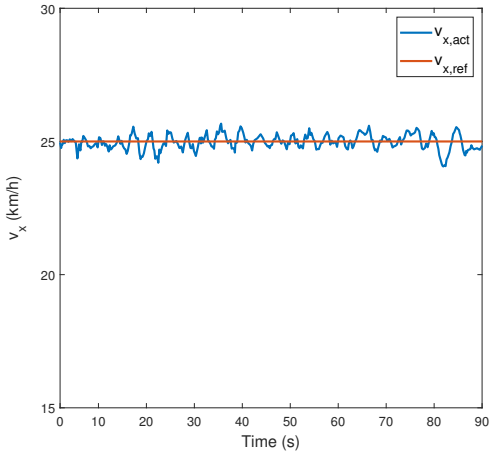


(a) Error of lateral position (b) Error of yaw-angle

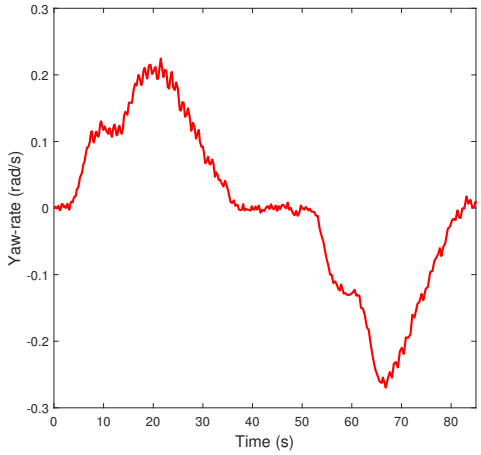
Fig. 6.8: Tracking errors

Thesis 5 I have developed a data-driven Linear Parameter-Varying (LPV)-based modeling method for a variable-geometry suspension testbed. Using the data-driven modeling method, I have designed a lower-level controller, which is able to realize the required steering angle on the variable-geometry suspension testbed. I have also developed a Model Predictive Control (MPC)-based lateral control method, which provides reference signal for the low-level controller. I have implemented and tested the proposed control strategy in Hardware-in-the-Loop (HiL) simulations.

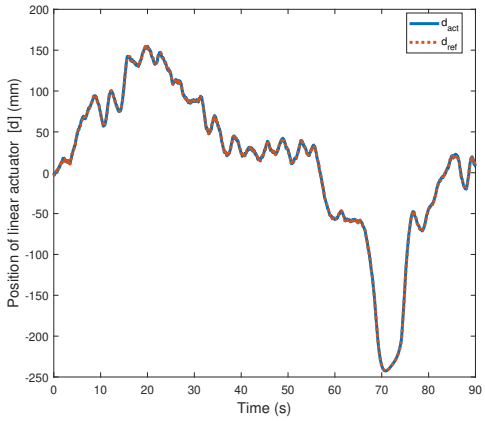
Related publications: [FFNG21, FNG21b, FNG21a]



(a) Speed of the vehicle



(b) Yaw-rate



(c) Actuator intervention

Fig. 6.9: Speed and yaw-rate signals

7. CONCLUSIONS AND FURTHER CHALLENGES

7.1 Conclusion

This thesis has proposed novel data-driven methods for improving the lateral control design of autonomous vehicles. Based on different machine-learning-based algorithms, novel estimation methods have been developed to determine the tyre pressure and the adhesion coefficient of the tyre-road contact. A data-driven lateral stability analysis of the vehicle has also been proposed, which can be used to approximate the operation regions of the vehicle at different conditions.

The presented estimation algorithms and the result of the stability analysis have been built in a lateral control design, which aimed to calculate the optimal velocity profile of the vehicle by which the stability of the vehicle can be maintained at safety-critic traffic scenarios.

The thesis has also addressed the modeling problem of the autonomous vehicle systems. Two methods have been proposed for improving the modeling of the lateral dynamics of the vehicle. The main concept behind both methods has been to determine the scheduling parameters of a system, which has been resulted in a set of Linear Time Invariant (LTI) systems. Then, the parameters of the resulted LTI systems have been optimized to get a more accurate model, which can accurately describe the nonlinear behavior of the vehicle. The first solution has been based on the decision tree algorithm, while the second one has used the pace-regression method. Finally, the efficiency of the proposed modeling algorithm has been demonstrated through a comprehensive simulation examples, performed in CarSim simulation software.

The thesis has also dealt with the modeling and control problems of the variable-geometry-suspension-based steering system, which can be used to improve the performances of the lateral control systems. A new physical model has been proposed, which can be used to describe the dynamics of the variable-geometry suspension. Based on this model, a hierarchical control strategy has been developed, which is able to combine the advantages of the variable-geometry-suspension-based steering system and the torque vectoring method. Finally, a data-driven modeling and control method has also been developed for a variable-geometry suspension testbed, which aimed to validate the theoretical concept. Then, the proposed control system has been implemented and tested in a HiL simulation, in which the lateral dynamics of the vehicle were modeled by the CarMaker simulation software.

7.2 New scientific results

Thesis 1 I have developed new data-driven estimation methods for vehicle control applications using different machine-learning-based algorithms. I have shown that the proposed algorithms can be used for estimating the tyre pressure and for approximating the adhesion coefficient between the road and the tyre. For the estimation of the adhesion coefficient, I have developed a method based on the C4.5 decision tree algorithm, while the estimation of the tyre pressure has been based on the pace regression and the neural network approaches. Furthermore, I have also designed a Linear Parameter-Varying (LPV) based trajectory tracking controller, which can incorporate in the result of the tyre pressure estimation algorithm in order to increase the performances and to guarantee the stable motion of the vehicle.

Related publications: [FNG20a, FNG21c, HFNG20, FNGA19, FNGS19, FNG19a, FNG$^+$19e, FNG19c, FNG19b]

Thesis 2 I have developed a method for approximating lateral stability region of passenger vehicles. The separation of the stable and unstable regions are based on a novel stability criterion using the two-wheeled lateral vehicle model. I have used the separated instances to build a classification model, which can be used to determine the stability of the vehicle during its operation using a decision tree-based algorithm. Furthermore, I have developed a Linear Parameter-Varying (LPV)-based lateral controller, which can handle the effect of the adhesion coefficient and the uncertainty of the estimation. I have also shown that the classification model can be used for a longitudinal velocity optimization algorithm, which can guarantee the stable motion of the vehicle by adjusting its velocity to the forthcoming road segment.

Related publications: [FNG20a, FNG19d, FNG19a, FNG19c, FNG19b, FNG18a, FNG21c, FNG21e]

Thesis 3 I have developed two methods for modeling the lateral dynamics of the vehicle. Both methods are based on the Linear Parameter Varying (LPV) approach combining with different machine-learning-based algorithms: pace regression and decision tree generation. I have shown that the models provided by the proposed algorithms are more efficient at modeling the lateral dynamics of the vehicle than the classical purely physical-based modeling approaches. I have developed a lateral robust controller using the proposed model structures.

Related publications: [FNG21d, FN20, FNG20b, FNG20c]

Thesis 4 I have developed a novel physical model for a steering concept based on a variable-geometry suspension system. Using the model, I have designed an integrated steering control system that combines the variable-geometry suspension with the benefits of torque vectoring-based steering method. The proposed method guarantees the stability and the controllability of the vehicle in critical situations. I

have developed a coordination strategy, which can be used to select the appropriate intervention during the operation of the vehicle.

Related publications: [NFGB19, FNG18c, FNG18b, FNG17, NGFB17b, NGFB17a, FNG16a, FNG16b, NFGB16, NGFB17b, NFG16a]

Thesis 5 I have developed a data-driven Linear Parameter-Varying (LPV)-based modeling method for a variable-geometry suspension testbed. Using the data-driven modeling method, I have designed a lower-level controller, which is able to realize the required steering angle on the variable-geometry suspension testbed. I have also developed a Model Predictive Control (MPC)-based lateral control method, which provides reference signal for the low-level controller. I have implemented and tested the proposed control strategy in Hardware-in-the-Loop (HiL) simulations.

Related publications: [FFNG21, FNG21b, FNG21a]

7.3 Further challenges

In the followings, further challenges and possible future research directions of data-driven control are suggested. The presented methods and algorithms can be extended in several ways:

- One of the main future challenges is the implementation and validation of the presented algorithms. Especially, the estimation algorithms, presented in Chapter 2, which will be carried out on the test vehicles of SZTAKI and BME, see [98].

- Regarding the contributions of Chapter 3,4, the implementation and validation are the main challenges. Although the implementation can be difficult on a real test vehicle, on a modified RC car it can be carried out. As a part of the Autonomous Systems National Laboratory, in SZTAKI advanced indoor RC cars have been used to validated learning-based control methods. Together with ELKH Cloud, it can be an appropriate platform for implementing the developed data-driven methods.

- Regarding the variable-geometry suspension test bed, there are also some open questions. Although the implementation and the validation of the original variable-geometry suspension-based steering system have been carried out, the integrated actuation (independent steering and torque vectoring) is still open. Therefore, the main goal is to improve the testbed with a direct torque measurement device and to design a control system, which can handle the multiple interventions. This purpose is connected to provide a method for constructional design of the variable-geometry suspension, i.e., the characteristics of the control intervention in the constructional design of the actuator must be incorporated.

- Another relevant topic, which is related to the data-driven control, is the data-driven observer design. In some complex vehicle-oriented control problems, the accurate measurement or estimation of the system states is essential. However, some problem arises as in the case of the control of nonlinear systems. Data-driven methods can also be used for this problem. The author has already published some preliminary results, see [FHNG21]. In this work, a neural network-based observer is combined with a classical robust observer by which the performances of the estimation can be significantly improved. However, the combination of different observer methods is still an open question.

PUBLICATIONS OF THE AUTHOR

[FFNG21] Dániel Fényes, Máté Fazekas, Balázs Németh, and Péter Gáspár. Implementation of a variable-geometry suspension-based steering control system. *Vehicle System Dynamics*, 0(0):1–18, 2021.

[FHNG21] Dániel Fényes, Tamás Hegedűs, Balázs Németh, and Péter Gáspár. Observer design with performance guarantees for vehicle control purposes via the integration of learning-based and LPV approaches. In *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshop)*, pages 122–127, 2021.

[FN20] Dániel Fényes and Balázs Németh. LPV alapú irányítástervezés automatizált járművek számára adatvezérelt módszerek alkalmazásával. In *XIV. Innováció és fenntartható felszíni közlekedés konferencia, IFFK 2020*, pages 1–6, 2020.

[FNG16a] Dániel Fényes, Balázs Németh, and Péter Gáspár. Control-oriented modelling of the variable-geometry suspension for independent steering purposes. In *Proceedings of the 15th MINI Conference on Vehicle System Dynamics, Identification and Anomalies*, pages 303–309, 2016.

[FNG16b] Dániel Fényes, Balázs Németh, and Péter Gáspár. Handling of zero-crossing problems in the design of variable-geometry suspension control. In *2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 237–242, 2016.

[FNG17] Dániel Fényes, Balázs Németh, and Péter Gáspár. Optimal control design of a variable-geometry suspension with electro-hydraulic actuator. In *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 337–342, 2017.

[FNG18a] Dániel Fényes, Balázs Németh, and Péter Gáspár. Analysis of autonomous vehicle dynamics based on the big data approach. In *2018 European Control Conference (ECC)*, pages 219–224, 2018.

[FNG18b] Dániel Fényes, Balázs Németh, and Péter Gáspár. Control design of variable-geometry suspension systems using a reconfiguration strategy. In *2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 81–86, 2018.

[FNG18c] Dániel Fényes, Balázs Németh, and Péter Gáspár. Data-driven reachability analysis for the reconfiguration of vehicle control systems. *IFAC-PapersOnLine*, 51(24):831–836, 2018. 10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2018.

[FNG19a] Dániel Fényes, Balázs Németh, and Péter Gáspar. A predictive control for autonomous vehicles using big data analysis. *IFAC-PapersOnLine*, 52(5):191–196, 2019. 9th IFAC Symposium on Advances in Automotive Control AAC 2019.

[FNG19b] Dániel Fényes, Balázs Németh, and Péter Gáspár. Enhancement of autonomous vehicle control via the contributions of big data analysis. In *2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES)*, pages 319–324, 2019.

[FNG19c] Dániel Fényes, Balázs Németh, and Péter Gáspár. Impact of big data on the design of MPC control for autonomous vehicles. In *2019 18th European Control Conference (ECC)*, pages 4154–4159, 2019.

[FNG19d] Dániel Fényes, Balázs Németh, and Péter Gáspár. Lateral control design for autonomous vehicles using a big data-based approach. *Lecture Notes in Mechanical Engineering*, pages 1137–1143, 2019. Advances in Dynamics of Vehicles on Roads and Tracks. IAVSD 2019.

[FNG+19e] Dániel Fényes, Balázs Németh, Péter Gáspár, Dániel Pup, and Ferenc Szauter. Study on a road surface estimation method based on big data analysis. In *2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 57–62, 2019.

[FNG20a] Dániel Fényes, Balázs Németh, and Péter Gáspár. LPV-based autonomous vehicle control using the results of big data analysis on lateral dynamics. In *2020 American Control Conference (ACC)*, pages 2250–2255, 2020.

[FNG20b] Dániel Fényes, Balázs Németh, and Péter Gáspár. LPV-based control for automated driving using data-driven methods. *IFAC-PapersOnLine*, 53(2):13898–13903, 2020. 21th IFAC World Congress.

[FNG20c] Dániel Fényes, Balázs Németh, and Péter Gáspár. LPV based data-driven modeling and control design for autonomous vehicles. In *2020 European Control Conference (ECC)*, pages 1371–1376, 2020.

[FNG21a] Dániel Fényes, Balázs Németh, and Péter Gáspár. Data-driven modeling and control design in a hierarchical structure for a variable-geometry suspension testbed. In *2021 IEEE 60th Conference on Decision and Control (CDC) (accepted)*, 2021.

[FNG21b] Dániel Fényes, Balázs Németh, and Péter Gáspár. Data-driven modeling approach for control design of a variable-geometry suspension system. *Lecture Notes in Mechanical Engineering (accepted)*, 2021. Advances in Dynamics of Vehicles on Roads and Tracks. IAVSD 2021.

[FNG21c] Dániel Fényes, Balázs Németh, and Péter Gáspár. Design of LPV control for autonomous vehicles using the contributions of big data analysis. *International Journal of Control*, 0(0):1–12, 2021.

[FNG21d] Dániel Fényes, Balázs Németh, and Péter Gáspár. A novel data-driven modeling and control design method for autonomous vehicles. *Energies*, 14(2), 2021.

[FNG21e] Dániel Fényes, Balázs Németh, and Péter Gáspár. Stability analysis of semi-active suspension systems using a data-driven approach. *Periodica Polytechnica Transportation Engineering*, 2021.

[FNGA19] Dániel Fényes, Balázs Németh, Péter Gáspár, and Máté Asszonyi. Possibilities of vehicle state estimation using big data approaches. In *Proceedings of the 16th MINI Conference on Vehicle System Dynamics, Identification and Anomalies*, pages 395–401, 2019.

[FNGS19] Dániel Fényes, Balázs Németh, Péter Gáspár, and Zoltán Szabó. Road surface estimation based LPV control design for autonomous vehicles. *IFAC-PapersOnLine*, 52(28):120–125, 2019. 3rd IFAC Workshop on Linear Parameter Varying Systems (LPVS) 2019.

[HFNG20] Tamás Hegedűs, Dániel Fényes, Balázs Németh, and Péter Gáspár. Handling of tire pressure variation in autonomous vehicles: an integrated estimation and control design approach. In *2020 American Control Conference (ACC)*, pages 2244–2249, 2020.

[NFG16a] Balázs Németh, Dániel Fényes, and Péter Gáspár. Control of a variable-geometry suspension as independent wheel steering. In *Advanced Vehicle Control: Proceedings of the 13th International Symposium on Advanced Vehicle Control, AVEC'16*, pages 261–268, 2016.

[NFG16b] Balázs Németh, Dániel Fényes, and Péter Gáspár. Independent wheel steering control design based on variable-geometry suspension. *IFAC-PapersOnLine*, 49(11):426–431, 2016. 8th IFAC Symposium on Advances in Automotive Control AAC 2016.

[NFGB16] Balázs Németh, Dániel Fényes, Péter Gáspár, and József Bokor. Trajectory tracking based on independently controlled variable-geometry suspension for in-wheel electric vehicles. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1570–1575, 2016.

[NFGB19] Balázs Németh, Dániel Fényes, Péter Gáspár, and József Bokor. Coordination of independent steering and torque vectoring in a variable-geometry suspension system. *IEEE Transactions on Control Systems Technology*, 27(5):2209–2220, 2019.

[NGFB17a] Balázs Németh, Péter Gáspár, Dániel Fenyes, and József Bokor. Reconfigurable control design of steering and torque vectoring based on reachability set analysis. *IFAC-PapersOnLine*, 50(1):3702–3707, 2017. 20th IFAC World Congress.

[NGFB17b] Balázs Németh, Péter Gáspár, Dániel Fényes, and József Bokor. Robust control design for the integration of steering and torque vectoring using a variable-geometry suspension system. In *2017 American Control Conference (ACC)*, pages 291–296, 2017.

BIBLIOGRAPHY

[1] Adnan Shaout, Dominic Colella, and S. Awad. Advanced driver assistance systems - past, present and future. In *2011 Seventh International Computer Engineering Conference (ICENCO'2011)*, pages 72–82, 2011.

[2] Ayman A. Aly, El-Shafei Zeidan, Ahmed Hamed, and Farhan Salem. An antilock-braking systems (ABS) control: A technical review. *Intelligent Control and Automation*, 2(3):186–195, 2011.

[3] Worrawut Pananurak, Somphong Thanok, and Manukid Parnichkun. Adaptive cruise control for an intelligent vehicle. In *2008 IEEE International Conference on Robotics and Biomimetics*, pages 1794–1799, 2009.

[4] A. Galip Ulsoy, Huei Peng, and Melih Cakmakci. *Vehicle Stability Control*, pages 257–271. Cambridge University Press, 2012.

[5] Abdelhamid Mammeri, Guangqian Lu, and Azzedine Boukerche. Design of lane keeping assist system for autonomous vehicles. In *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5, 2015.

[6] Rahul Kala. 4 - Advanced Driver Assistance Systems. In Rahul Kala, editor, *On-Road Intelligent Vehicles*, pages 59–82. Butterworth-Heinemann, 2016.

[7] Chinmay Vilas Samak, Tanmay Vilas Samak, and Sivanathan Kandhasamy. Control strategies for autonomous vehicles, 2020.

[8] P Ioannou and Z Xu. Throttle and brake control systems for automatic vehicle following, 1994.

[9] Andrew P. White, Guoming Zhu, and Jongeun Choi. *Linear Parameter-Varying Control for Engineering Applications*. Springer Briefs in Control, Automation and Robotics. Springer-Verlag London, 2013.

[10] Moustapha Doumiati, Alessandro Victorino, Reine Talj, and Ali Charara. Robust LPV control for vehicle steerability and lateral stability. In *53rd IEEE Conference on Decision and Control*, pages 4113–4118, 2014.

[11] S. Fergani, O. Sename, and L. Dugard. A LPV suspension control with performance adaptation to roll behavior, embedded in a global vehicle dynamic control strategy. In *2013 European Control Conference (ECC)*, pages 1487–1492, 2013.

[12] S. Fergani, L. Menhour, O. Sename, L. Dugard, and B. D'Andrea Novel. Full vehicle dynamics control based on LPV H_∞ and flatness approaches. In *2014 European Control Conference (ECC)*, pages 2346–2351, 2014.

[13] Wang Qiu, Qu Ting, Yu Shuyou, Guo Hongyan, and Chen Hong. Autonomous vehicle longitudinal following control based on model predictive control. In *2015 34th Chinese Control Conference (CCC)*, pages 8126–8131, 2015.

[14] Yunao Li, Senchun Chai, Ruiqi Chai, and Xiaopeng Liu. An improved model predictive control method for vehicle lateral control. In *2020 39th Chinese Control Conference (CCC)*, pages 5505–5510, 2020.

[15] Shilp Dixit, Umberto Montanaro, Mehrdad Dianati, David Oxtoby, Tom Mizutani, Alexandros Mouzakitis, and Saber Fallah. Trajectory planning for autonomous high-speed overtaking in structured environments using robust mpc. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2310–2323, 2020.

[16] Balázs Németh, Tamás Hegedűs, and Péter Gáspár. Model predictive control design for overtaking maneuvers for multi-vehicle scenarios. In *2019 18th European Control Conference (ECC)*, pages 744–749, 2019.

[17] Richard Bellman. Proceedings of the national academy of sciences of the united states of america. In *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, volume 40, pages 231–235, 1954.

[18] Richard Bellman. *Dynamic Programming*. Princeton University Press, N.J., 1957.

[19] J.L. Lagrange, J.P.M. Binet, and J.G. Garnier. *Mécanique analytique (Analytical Mechanics)*. Ve Courcier, Paris, 1811.

[20] M. Canale, L. Fagiano, and V. Razza. Vehicle lateral stability control via approximated nmpc: real-time implementation and software-in-the-loop test. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 4596–4601, 2009.

[21] Paolo Falcone, H. Eric Tseng, Francesco Borrelli, Jahan Asgari, and Davor Hrovat. Mpc-based yaw and lateral stabilisation via active front steering and braking. *Vehicle System Dynamics*, 46(sup1):611–628, 2008.

[22] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah. A survey of deep learning applications to autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–22, 2020.

[23] C. Hubschneider, A. Bauer, J. Doll, M. Weber, S. Klemm, F. Kuhnt, and J. M. Zollner. Integrating end-to-end learned steering into probabilistic autonomous driving. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7, Oct 2017.

[24] V. Rausch, A. Hansen, E. Solowjow, C. Liu, E. Kreuzer, and J. K. Hedrick. Learning a deep neural net policy for end-to-end control of autonomous vehicles. In *2017 American Control Conference (ACC)*, pages 4914–4919, May 2017.

[25] D.A. Pomerleau. Knowledge-based training of artificial neural networks for autonomous robot driving. *Robot Learning*, 233:13–43, 1993.

[26] L. Cavanini, F. Ferracuti, S. Longhi, and A. Monteriu. LS-SVM for LPV - ARX identification: Efficient online update by low-rank matrix approximation. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1590–1595, 2020.

[27] R. A. Romano, P. L. dos Santos, F. Pait, T. Perdicoulis, and J. A. Ramos. Machine learning barycenter approach to identifying LPV state-space models. In *2016 American Control Conference (ACC)*, pages 6351–6356, 2016.

[28] Y. Bao and J. M. Velni. Data-driven linear parameter-varying model identification using transfer learning. *IEEE Control Systems Letters*, 5(5):1579–1584, 2021.

[29] S. Abdufattokhov and B. Muhiddinov. Stochastic approach for system identification using machine learning. In *2019 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, pages 1–4, 2019.

[30] G. Biagetti, P. Crippa, L. Falaschetti, and C. Turchetti. Machine learning regression based on particle bernstein polynomials for nonlinear system identification. In *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2017.

[31] Ugo Rosolia and Francesco Borrelli. Learning model predictive control for iterative tasks. a data-driven control framework, 2017.

[32] Michel Fliess and Cédric Join. Model-free control. *International Journal of Control*, 86(12):2228–2252, 2013.

[33] S. Formentin, P. De Filippi, M. Corno, M. Tanelli, and S. M. Savaresi. Data-driven design of braking control systems. *IEEE Transactions on Control Systems Technology*, 21(1):186–193, Jan 2013.

[34] Rie B. Larsen, Andrea Carron, and Melanie N. Zeilinger. Safe learning for distributed systems with bounded uncertainties. *IFAC-PapersOnLine*, 50(1):2536–2542, 2017. 20th IFAC World Congress.

[35] Jaime F. Fisac, Anayo K. Akametalu, Melanie N. Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J. Tomlin. A general safety framework for learning-based control in uncertain robotic systems, 2018.

[36] Lianfei Zhai, Tianyou Chai, and Shuzhi Sam Ge. Stable adaptive neural network control of nonaffine nonlinear discrete-time systems and application. In *2007 IEEE 22nd International Symposium on Intelligent Control*, pages 602–607, 2007.

[37] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. MIT Press Ltd, 2018.

[38] Arpad Feher, Szilard Aradi, and Tamas Becsi. Q-learning based reinforcement learning approach for lane keeping. In *2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 000031–000036, 2018.

[39] Wei Xia, Huiyun Li, and Baopu Li. A control strategy of autonomous vehicles based on deep reinforcement learning. In *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, pages 198–201, 2016.

[40] R. Matthew Kretchmar, Peter M. Young, Charles W. Anderson, Douglas C. Hittle, Michael L. Anderson, and Christopher C. Delnero. Robust reinforcement learning control with static and dynamic stability. *International Journal of Robust and Nonlinear Control*, 11(15):1469–1500, 2001.

[41] W.J. Evers, A. van der Knaap, I. Besselink, and H. Nijmeijer. Analysis of a variable geometry active suspension. *International Symposium on Advanced Vehicle Control, Kobe, Japan*, 2008.

[42] S. Lee, H. Sung, and U. Lee. A study to the enhancement of vehicle stability by active geometry control suspension (AGCS) system. *13th International Pacific Conference on Automotive Engineering, Gyeongju, Korea*, pages 1–6, 2005.

[43] Willem-Jan Evers, Albert van der Knaap, Igo Besselink, and Henk Nijmeijer. Modeling, analysis and control of a variable geometry actuator. In *2008 IEEE Intelligent Vehicles Symposium*, pages 251–256, 2008.

[44] O. M. Anubi, D. R. Patel, and C. D Crane III. A new variable stiffness suspension system: passive case. *Mechanical Sciences*, 4:139–151, 2013.

[45] U. K. Lee and C. S. Han. A suspension system with a variable roll centre for the improvement of vehicle handling characteristics. *Proceedings of the IMechE, Part D: Journal of Automobile Engineering*, 215(6):677–696, 2001.

[46] C. Arana, S. A. Evangelou, and D. Dini. Series active variable geometry suspension for road vehicles. *IEEE/ASME Transactions on Mechatronics*, 20(1):361–372, 2015.

[47] S.H. Lee, H. Sung, J.W. Kim, and U.K. Lee. Enhancement of vehicle stability by active geometry control suspension system. *International Journal of Automotive Technology*, 7(3):303–307, 2006.

[48] Avesta Goodarzia, Ehsan Oloomia, and Ebrahim Esmailzadehb. Design and analysis of an intelligent controller for active geometry suspension systems. *Vehicle System Dynamics*, 49(1):333–359, 2010.

[49] B. Németh and P. Gáspár. Control design of variable-geometry suspension considering the construction system. *IEEE Transactions on Vehicular Technology*, 62(8):4104–4109, 2013.

[50] J. Wang, Q. Wang, L. Jin, and C. Song. Independent wheel torque control of 4WD electric vehicle for differential drive assisted steering. *Mechatronics*, 21:63–76, 2011.

[51] C. Hu, H. Jing, R. Wang, F. Yan, and N. Chen. Fault-tolerant control of FWIA electric ground vehicles with differential drive assisted steering. *9th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, Paris, France*, 48(21):1180–1185, 2015.

[52] Balázs Németh and Péter Gáspár. Nonlinear analysis and control of a variable-geometry suspension system. *Control Engineering Practice, article in press*, 2017.

[53] Balázs Németh and Péter Gáspár. Control design of variable-geometry suspension considering the construction system. *IEEE Transactions on Vehicular Technology*, 62(8):4104–4109, 2013.

[54] *IPGDriver Reference Manual version 6.5*. IPG Automotive GmbH, 2016.

[55] Yong Wang and Ian H. Witten. *Pace Regression*. (Working paper 99/12). Hamilton, New Zealand: University of Waikato, Department of Computer Science., 1999.

[56] R. Shibata. An optimal selection of regression variables. *Biometrika*, 68:45–54, 1981.

[57] M. L. Thompson. Selection of variables in multiple regression. *International Statistical Society B*, 46:1–21 and 129–146, 1978.

[58] H. Demut, M. Hagan, and M. Beale. *Neural network design*. PWS Publishing Co, 1997.

[59] R. Rajamani. Vehicle dynamics and control. *Springer*, 2005.

[60] H. B. Pacejka. *Tyre and vehicle dynamics*. Elsevier Butterworth-Heinemann, Oxford, 2004.

[61] B. Németh and P. Gáspár. Design of actuator interventions in the trajectory tracking for road vehicles. *Proc. of the Conference on Decision and Control, Orlando, Florida*, 2011.

[62] Lennart Ljung. *System identification: theory for the user*. Prentice-Hall, USA, 2003.

[63] Shoutao Li, Xinglong Pei, and Yongxue Ma. A new road friction coefficient estimation method based on svm. *2012 IEEE International Conference on Mechatronics and Automation*, pages 1910–1914, 2012.

[64] E. B. Hunt. *Concept Learning: An information Processing Problem*. Wiley, 1962.

[65] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, California, 1993.

[66] J. R. Quinlan. Bagging, Boosting, and C4.5. *Proceedings of the Thriteenth National Conference on Artificial Intelligence*, pages 725–730, 1996.

[67] M.I. Masouleh and D.J.N. Limebeer. Region of attraction analysis for nonlinear vehicle lateral dynamics using sum-of-squares programming. *Vehicle System Dynamics*, 56(7):1118–1138, 2018.

[68] S. Sadri and C. Wu. Stability analysis of a nonlinear vehicle model in plane motion using the concept of Lyapunov exponents. *Vehicle System Dynamics*, 51(6):906–924, 2013.

[69] A.M. Lyapunov. *Stability of Motion*. Academic Press, New-York and London, 1966.

[70] I.H. Witten and E. Frank. *Data Mining Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, Elsevier, 2005.

[71] Péter Gáspár, Zoltán Szabó, József Bokor, and Balázs Németh. *Robust Control Design for Active Driver Assistance Systems. A Linear-Parameter-Varying Approach.* Springer Verlag, 2017.

[72] F. Wu, X.H. Yang, A. Packard, and G. Becker. Induced L_2 norm controller for LPV systems with bounded parameter variation rates. *Journal of Robust and Nonlinear Control*, 6:983–988, 1996.

[73] Z. Szabó, A. Marcos, D. P. Mostaza, M. Kerr, G. Rödönyi, J. Bokor, and S. Bennani. Development of an integrated LPV /LFT framework: modeling and data-based validation tool. *IEEE Transactions on Control Systems Technology*, 19(1):104–117, 2011.

[74] Roland Tóth. *Modeling and Identification of Linear Parameter-Varying Systems*, volume 403 of *Lecture Notes in Control and Information Sciences.* Springer, Berlin, Heidelberg, 2010.

[75] Giovanni Palmieri, Miroslav Barić, Luigi Glielmo, and Francesco Borrelli. Robust vehicle lateral stabilisation via set-based methods for uncertain piecewise affine systems. *Vehicle System Dynamics*, 50(6):861–882, 2012.

[76] P. E. Gill, W. Murray, and M.H. Wright. *Practical Optimization.* Academic Press, London UK, 1981.

[77] T. F. Coleman and Y. Li. A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 6(4):1040–1058, 1996.

[78] B. Németh, P. Gáspár, and T. Péni. Nonlinear analysis of vehicle control actuations based on controlled invariant sets. *Int. J. Applied Mathematics and Computer Science*, 26(1), 2016.

[79] O. Sename, P. Gáspár, and J. Bokor. *Robust Control and Linear Parameter Varying Approaches.* Springer Verlag, Berlin, 2013.

[80] B. Németh, B. Varga, and P. Gáspár. Hierarchical design of an electro-hydraulic actuator based on robust LPV methods. *International Journal of Control*, 88(8):1429–1440, 2015.

[81] F.-K. Wu, T.-J. Yeh, and C.-F. Huang. Motor control and torque coordination of an electric vehicle actuated by two in-wheel motors. *Mechatronics*, pages 46–60, 2013.

[82] T. Hsiao. Robust wheel torque control for traction/braking force tracking under combined longitudinal and lateral motion. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1335–1347, 2015.

[83] J. Bokor and G. Balas. Linear parameter varying systems: A geometric theory and applications. *16th IFAC World Congress, Prague*, 2005.

[84] F. Wu, X. H. Yang, A. Packard, and G. Becker. Induced l^2-norm control for LPV systems with bounded parameter variation rates. *International Journal of Nonlinear and Robust Control*, 6:983–998, 1996.

[85] A. Packard and G. Balas. Theory and application of linear parameter varying control techniques. *American Control Conference, Workshop I, Albuquerque, New Mexico*, 1997.

[86] K. Sawase and Y. Ushiroda. Improvement of vehicle dynamics by rightand left torque vectoring system in various drive trains. *Mitsubishi Motors Technical Review*, 20:14–20, 2008.

[87] M. Klomp. Longitudinal force distribution using quadratically constrained linear programming. *Vehicle System Dynamics*, 49(12):1823–1836, 2011.

[88] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[89] S. Horiuchi. Evaluation of chassis control algorithms using controllability region analysis. In *The Dynamics of Vehicles on Roads and Tracks Proceedings of the 24th Symposium of the International Association for Vehicle System Dynamics (IAVSD 2015)*, pages 35–43, Graz, Austria, 2015.

[90] D. R. Snow. *Determining reachable regions and optimal controls*, volume 5 of *Advances in Control Systems*. Academic Press, 1967.

[91] P. Gruber, A. Sorniotti, B. Lenzo, G. De Filippis, and S. Fallah. Energy efficient torque vectoring control. In Johannes Edelmann, Manfred Plöchl, and Peter E. Pfeiffer, editors, *Advanced Vehicle Control AVEC'16*, pages 17–22. CRC Press/Balkema, 2016.

[92] H. Grip, L. Imsland, T. Johansen, T. Fossen, J. Kalkkuhl, and A. Suissa. Nonlinear vehicle side-slip estimation with friction adaptation. *Automatica*, 44(11):611–622, 2008.

[93] Stefan Chamraz and Richard Balogh. Two approaches to the adaptive cruise control (ACC) design. *Proceedings of the 29th International Conference 2018 Cybernetics and Informatics*, pages 1–6, 2018.

[94] Suwandi Dwi Sahputro, Fahmi Fadilah, Nanda Avianto Wicaksono, and Feri Yusivar. Design and implementation of adaptive pid controller for speed control of dc motor. *2017 15th International Conference on Quality in Research*

(QiR) : International Symposium on Electrical and Computer Engineering, pages 179–183, 2017.

[95] Hongliang Zhou, Levent Guvenc, and Zhiyuan Liu. Design and evaluation of path following controller based on MPC for autonomous vehicle. *Proceedings of the 36th Chinese Control Conference*, pages 9934–9939, 2017.

[96] Milad Jalali, Saeid Khosravani, Amir Khajepour, Shih ken Chen, and Bakhtiar Litkouhi. Model predictive control of vehicle stability using coordinated activesteering and differential brakes. *Mechatronics*, 48:30–41, 2017.

[97] M. Choi and S. B. Choi. Model predictive control for vehicle yaw stability with practical concerns. *IEEE Transactions on Vehicular Technology*, 63(8):3539–3548, 2014.

[98] Benedek Szűcs, Ádám Kisari, Péter Kőrös, Dániel Pup, Gábor Rödönyi, Alexandros Soumelidis, and József Bokor. Experimental verification of a control system for autonomous navigation. *IFAC-PapersOnLine*, 53(2):14273–14278, 2020. 21th IFAC World Congress.

[99] Fen Wu. *Control of Linear Parameter Varying Systems*. PhD thesis, University of California at Berkeley, 1995.

[100] Kristin Lee Fitzpatrick. *Applications of Linear Parameter-Varying Control for Aerospace Systems*. PhD thesis, University of Florida, 2003.

LIST OF FIGURES

LIST OF TABLES

APPENDIX

.1 LPV-based modeling and control

The Linear Parameter Varying (LPV) framework is a powerful technique to model and control nonlinear systems [99]. It has been successfully applied to numerous control problems including automotive and aerospace applications, see [79], [100]. In the followings, the basic concept of the LPV framework is presented and a brief introduction is given for the control synthesis based upon LPV approach.

Let us consider a general nonlinear system, which is given by its first order differential equations:

$$\dot{x} = f(x, u, t), \tag{.1a}$$
$$y = g(x, u, t) \tag{.1b}$$

where $x \in \mathbb{R}^n$ denotes the state of the nonlinear system, $u \in \mathbb{R}^m$ is the input vector and y represents the measurement vector $y \in \mathbb{R}^k$. The dynamics of the system is described by two nonlinear functions: $f(x, u, t)$: $\mathbb{R}^{n+m+1} \rightarrow \mathbb{R}^n$, while $g(x, u, t) : \mathbb{R}^{n+m+1} \rightarrow \mathbb{R}^k$. Note that the system also depends in the time variable (t), which means that a general time variant system is considered.

The basic concept behind LPV framework is to transform the original nonlinear time variant system into a set of Linear Time Invariant (LTI) systems. This set of linear systems can be arranged into a general state-space representation as:

$$\dot{x} = A(\rho)x + B(\rho)u \tag{.2a}$$
$$y = C(\rho)x + D(\rho)u \tag{.2b}$$

The state-space representation can also be rewritten in a compact form:

$$\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} A(\rho) & B(\rho) \\ C(\rho) & D(\rho) \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \tag{.3}$$

where $A(\rho), B(\rho), C(\rho), D(\rho)$ are the state matrices, which depend on the scheduling vector ρ. In this case, the scheduling vector ρ, contains all the exogenous variables, which define each specific LTI system of the set, as shown in Figure .1.

However, LPV framework is not limited to exogenous scheduling variables, it can also be used to represent systems, whose scheduling vector consists of one or more state variables of the original nonlinear plant. In that case, this formalization results in a quasi-LPV system, which can be described as:

$$\begin{bmatrix} \dot{z} \\ \dot{w} \\ y \end{bmatrix} = \begin{bmatrix} A_{11}(\rho) & A_{12}(\rho) & B_1(\rho) \\ A_{21}(\rho) & A_{22}(\rho) & B_2(\rho) \\ C_1(\rho) & C_2(\rho) & D(\rho) \end{bmatrix} \begin{bmatrix} z \\ w \\ u \end{bmatrix} \tag{.4}$$

In this case, the scheduling vector consists of two main components: the exogenous and the state-dependent scheduling variables $\rho = [z \quad w]$. Therefore, the original system is also decomposed into two parts: z denotes those states, which are used as scheduling parameters, while w contains the rest of the state variables.

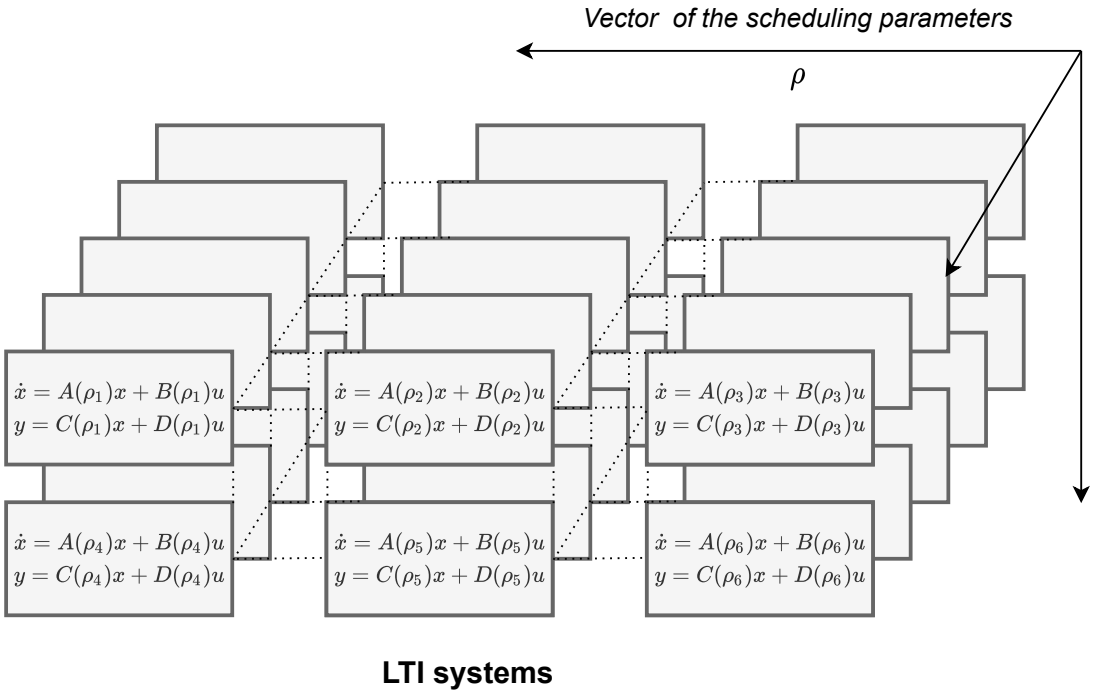


Fig. .1: LPV system as a set of LTI systems

.1.1 Transformation of a nonlinear plant to LPV form

There are several methods to transform a nonlinear plant a LPV form. The most common involves a Jabocian linearization method. The determination of the LPV representation consists of the following main steps:

- First step is the selection of the scheduling parameters of the LPV system. It is important to note that the selected scheduling variables must be measurable (or observable) in real time for the control synthesis.
- Second step is the determination of the operating point of the system using the variables in the scheduling vector ρ.
- Third step is the linearization around the desired operating points.

Then, the presented state-space representation (.4) can be built up. Among the operating points, the current state-space can be determined by using interpolation methods.

.1.2 Control synthesis for LPV systems

The control synthesis of a LPV system is similar to the conventional robust methods, such as $\mathcal{H}_\infty$. The desired performances of the closed-loop system can be achieved by using appropriately tuned weighting functions. An example is shown in

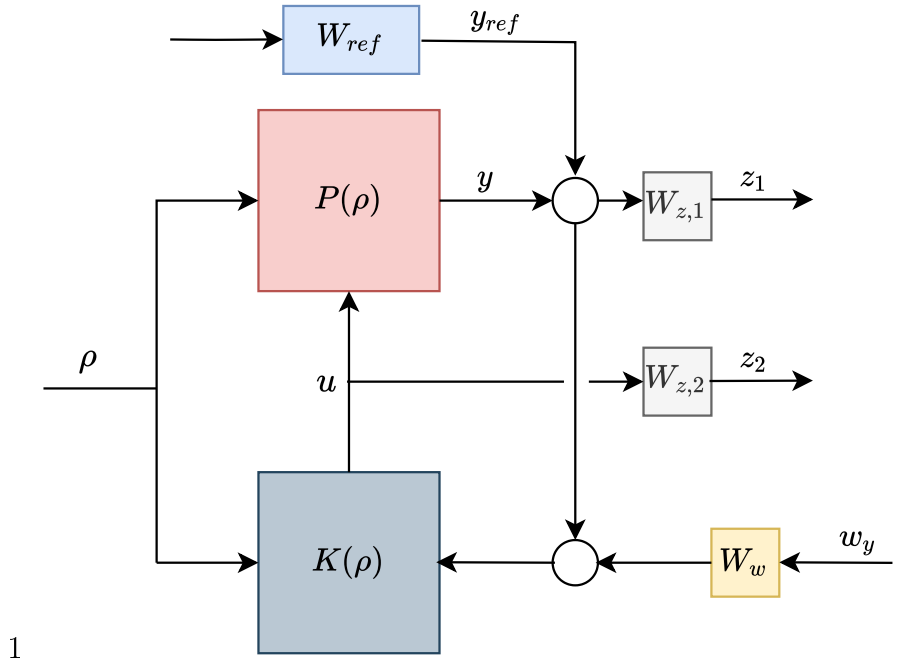


Fig. .2: Example: LPV control synthesis

Figure .2. A common goal is to guarantee the tracking of a specific reference signal ($|y_{ref} - y| \rightarrow min!$), which is guaranteed by the weighting function $W_{z,1}$. Another could be to minimize the control input ($|u| \rightarrow min!$), which is scaled by the function $W_{z,2}$. In real systems, the measurement noises must be taken into account during the control design phase. It can also be done, in this case, using the weighting function W_w.

There are three main methods, which can be used for control synthesis:

- Single Quadratic Lyapunov Function, which can be used for LPV systems, which only have exogenous scheduling variables.
- Parameter Dependent Lyapunov Function, which can handle state-dependent scheduling variables.
- Linear Fractional Transformation (LFT)-based solutions based on the small gain theorem.

The general control synthesis problem to choose the parameter-varying controller $K(\rho)$ in such a way that the resulting closed-loop system is quadratically stable and the induced $\mathcal{L}_2$ norm from disturbance and to performances is less than the value γ. The minimization task is the following:

$$\inf_{K(\rho)} \sup_{\rho \in F_\rho} \sup_{\|w\|_2 \neq 0, w \in \mathcal{L}_2} \frac{\|z\|_2}{\|w\|_2}, \tag{.5}$$

where F_ρ bounds the scheduling variables. The yielded controller $K(\rho)$ is formed as

$$\dot{x}_K = A_K(\rho)x_K + B_K(\rho)y_K, \tag{.6a}$$
$$u = C_K(\rho)x_K + D_K(\rho)y_K, \tag{.6b}$$

where $A_K(\rho), B_K(\rho), C_K(\rho), D_K(\rho)$ are variable-dependent matrices.

LIST OF NOTES

LPV	Linear Parameter Varying
ADAS	Advanced Driver Assistance Systems
MPC	Model Predictive Control
ψ	yaw angle
β	side-slip angle
γ	wheel camber angle
m	mass of vehicle
I_z	yaw-inertia of chassis
α_f, α_r	lateral slip of front/rear tyre
l_1, l_2	distance between vehicle center of gravity and front/rear axle
$C_{\alpha,f}, C_{\alpha,r}$	cornering stiffness of front/rear tyre
y	lateral vehicle position
δ	front steering angle
M_d	generated yaw moment
v_x	longitudinal velocity
v_y	lateral velocity
a_x	longitudinal acceleration
a_y	lateral acceleration
$\omega_{f,f}, \omega_{f,r}, \omega_{r,f}, \omega_{r,f}$	angular velocity of the front/rear, left/right wheels
δ, l, δ, r	road wheel angle left/right
δ_s	angle of the steering wheel
$M_{f,f}, M_{f,r}, M_{r,f}, M_{r,r}$	Torques of the wheels front/rear, left/right
$\dot{\varphi}$	roll rate
B	wheel track
z	performance
h	height of CoG
$m_{s,f}, m_{s,r}$	mass of suspension front/rear
ρ	scheduling variable
$K(\rho)$	scheduling variable dependent controller