energies

MDPI

Article

A Novel Data-Driven Modeling and Control Design Method for Autonomous Vehicles

Dániel Fényes [1], Balázs Németh [2] and Péter Gáspár [2,*]

[1] Department of Control for Transportation and Vehicle Systems, Budapest University of Technology and Economics, H-1111 Budapest, Hungary; daniel.fenyes@sztaki.hu
[2] Systems and Control Laboratory, Institute for Computer Science and Control (SZTAKI), H-1111 Budapest, Hungary; balazs.nemeth@sztaki.hu
* Correspondence: gaspar@sztaki.hu

Abstract: This paper presents a novel modeling method for the control design of autonomous vehicle systems. The goal of the method is to provide a control-oriented model in a predefined Linear Parameter Varying (*LPV*) structure. The scheduling variables of the *LPV* model through machine-learning-based methods using a big dataset are selected. Moreover, the *LPV* model parameters through an optimization algorithm are computed, with which accurate fitting on the dataset is achieved. The proposed method is illustrated on the nonlinear modeling of the lateral vehicle dynamics. The resulting *LPV*-based vehicle model is used for the control design of path following functionality of autonomous vehicles. The effectiveness of the modeling and control design methods through comprehensive simulation examples based on a high-fidelity simulation software are illustrated.

Keywords: *LPV* control design; machine learning in modeling; vehicle dynamics

Citation: Fényes, D.; Németh, B.; Gáspár, P. A Novel Data-Driven Modeling and Control Design Method for Autonomous Vehicles. *Energies* **2021**, *14*, 517. https://doi.org/10.3390/en14020517

Received: 14 December 2020
Accepted: 15 January 2021
Published: 19 January 2021

1. Introduction and Motivation

Nowadays, one of the major challenges for the automotive industry is the development of the fully autonomous vehicles, which is represented by level 5 of SAE J3016. This challenge involves several disciplines (e.g., sensing, controlling, decision making), which must work together to develop safe and reliable solutions for autonomous vehicles. One of the most significant tasks is the control design, because in the dynamics of the vehicle, it is highly nonlinear in its entire operation range. The nonlinear behavior becomes especially important at dangerous maneuvers (e.g., low road adhesion, high speed etc.), in which the vehicle is close to its physical limits.

Controlling of nonlinear system is still a challenging task, which draws attention from both engineers and researchers. In the past decades, several algorithms and control methods have been developed to deal with this problem. The developed approaches in two main categories can be classified: classical methods and machine learning-based solutions.

The classical methods includes the model-based control analysis and synthesis techniques, i.e., linear robust and optimal control methods ($\mathcal{H}_2, \mathcal{H}_\infty$), parameter-varying methods, e.g., Linear Parameter Varying (*LPV*) and Nonlinear Parameter Varying (*NLPV*) [1,2], Model Predictive Control (*MPC*) approaches [3] or polynomial methods [4,5]. One of the main sources of the nonlinearities is the tire-road contact, it stability is analyzed in [6] and a Lyapunov control method for this problem is presented in [7]. The uncertainties and the nonlinearities of the vehicle may have external sources, which can be handled by an adaptive robust controller, see [8]. The steering system and its actuator also have nonlinearities, which can be controlled by an *LPV*-based controller [9]. The identification of the uncertainties and nonlinearities is also a challenging task. The authors of [10] present a grey-box *LPV* identification method for vehicles using side-slip angle estimation. An *LPV* framework-based uncertainty identification is proposed in [11] using experimental data.

These algorithms handle nonlinearities and unmodelled dynamics in the form of uncertainties and thus, they provide conservative solutions. Nevertheless, the result of the design is a control loop with guarantees on the stability and on the performances.

The machine-learning-based solutions include various methods, such as deep learning through the training of neural networks [12], Support Vector Machine (*SVM*), or decision logic algorithms [13–15]. 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 provided. 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 that deal with the identification problem of the dynamical systems. For example, in [16,17] *SVM-LS*-based *LPV* system identification methods are presented. Another data-driven identification solution can be found in [18], which can ease the tuning of the hyperparameters of the model identification process. A non-parametric and probabilistic approach is proposed by [19] for identifying nonlinear system considering uncertain noises on the measured signals. particle Bernstein polynomials-based regression method is presented in [20], 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, [21] 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. Another solution is the Model Free Control (*MFC*), which is proposed in [22,23]. The *MFC* method does not require a model of the controlled system, but it uses a local model, in which the model of each time step is updated.

The goal of this paper is to reformulate the design of vehicle path tracking functionality as a modeling problem with learning features and a control design problem using a model-based approach. The main contribution of the paper is a novel identification method, by which an *LPV* model of the nonlinear system can be determined. Moreover, the selection of the scheduling parameters of the *LPV* system is determined by a machine-learning-based pace regression algorithm in the modeling process of the system. In this way, the performances of the *LPV* controller are enhanced, while the stability of the closed-loop system is also preserved. The efficiency and the operation of the proposed control algorithm is validated by a complex test scenario, which is performed in the high-fidelity simulation software, CarSim.

The process of the proposed method for achieving an enhanced vehicle control system is presented in Figure 1. The process consists of three main layers with several tasks, i.e., *Simulation environment*, *Modeling error computation using ML algorithm* and *Model identification and control design*. Layer *Simulation environment* consists of two tasks, namely, performing vehicle dynamic simulations for data generation and data acquisition. The vehicle dynamic simulations are performed on the high-fidelity simulation environment CarSim. The layer *Modeling error computation using ML algorithm* consists of several tasks. It contains the selection and categorization of the scheduling parameters using the pace regression algorithm. It provides the categorized datasets for the identification process. The *Model identification and control design* uses the categorized datasets for computing the models for each grid points of the *LPV* system. In this layer, the *LPV*-based control design is also presented, which guarantees the trajectory tracking of the vehicle.

In the paper, each layer and task are presented as follows. Section 2 presents the vehicle model parameter tuning method for achieving control-oriented *LPV* model. The control design based on the robust *LPV* synthesis method is proposed in Section 4. The effectiveness of the resulting control system through comparative simulation scenarios is examined in Section 5. Finally, the achievements of the paper are concluded in Section 6.

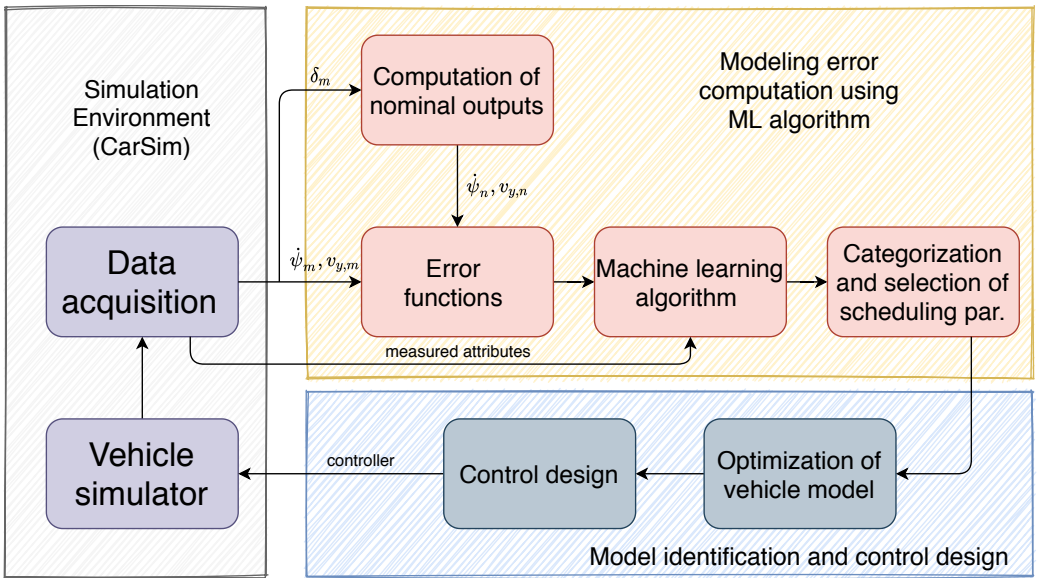


Figure 1. Methodological process for modeling, control design, and evaluation.

2. Formulation of the Control-Oriented *LPV* Model Using Data-Driven Approach

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 elements 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.

2.1. Acquisition of Data from Simulations

Since all of the machine learning methods require a large amount of data to provide satisfying results, the first the of the algorithm is the data acquisition. In this paper, the high-fidelity simulation software, CarSim is used to generate the datasets for the machine learning algorithm in the control-oriented model formulation process. Since an appropriate dataset must cover a wide range of the vehicle operation, several simulations have been performed on the simulation environment using different parameter sets. These parameter sets includes the longitudinal velocity profile, the maneuvers, which must be performed by the vehicle. The test scenarios included several different routes with sharp bends (e.g., Melbourne Formula (1) circuit), whose adhesion coefficient has also been modified $\mu \in [0.4 \dots 1]$ during the simulations. The dataset also includes simulations, in which the vehicle loses its stability, in order to cover most of the possible situations that the vehicle can encounter.

During the simulations, the following variables 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 all of the collected signals are 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 they are not required. The sampling time of the variables has been set to $T_s = 0.01$ s. In this way, a large dataset has been created, which consisted of more than 10 million instances.

2.2. 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 [24]. The separation of the stable/unstable instances using only the measured values is not straightforward task. There is no general solution for determining the stability of a measurement. However, for the automotive case, a criterion is proposed in [25], which can be used to separate the instances according to the stability of the vehicle:

$$-\varepsilon < \frac{|1+\alpha_1|}{|1+\delta-\beta-\frac{l_1\dot{\psi}}{v_x}|} - 1 \leq \varepsilon, \tag{1}$$

where ϵ is a experimentally defined parameter.

This criterion is based on the linear bicycle model, which will be detailed in the followings (2). Briefly, it compares the measured side-slips angle of the front wheel α_1, to the linear model, in which that angle is computed as $\alpha_1 = \delta - \beta - \frac{l_1\dot{\psi}}{v_x}$. The inequality (1) describes that the deviation of the measured instance from the linear region, which can be used to determine the stability of the vehicle. 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.

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 nominal model, which is used in this paper, is the two-wheeled bicycle model. The model consists of the following three main equations, ref. [26]:

$$I\ddot{\psi} = C_1\alpha_1 l_1 - C_2\alpha_2 l_2 \tag{2a}$$

$$ma_y = C_1\alpha_1 + C_2\alpha_2 \tag{2b}$$

$$a_y = \dot{v}_y + v_x\dot{\psi} \tag{2c}$$

where l_1, l_2 are the distances between the CoG and the front, rear axes of the vehicle, C_1, C_2 are the reduced cornering stiffness of the wheels, $\dot{\psi}$ denotes the yaw-rate, β is the slid-slip, m is the mass of the car, I represents the yaw-inertia, α_1, α_2 are the side-slip of the wheels and the longitudinal and lateral velocities are denoted by v_x, v_y. The lateral vehicle model can be transformed into a state-space representation

$$\dot{x} = Ax + Bu, \tag{3}$$

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 $T_s = 0.01$ s, 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

$$\Delta\dot{\psi} = \dot{\psi}_m(t_i) - \dot{\psi}_n(t_i) \tag{4a}$$

$$\Delta v_y = v_{y,m}(t_i) - v_{y,n}(t_i) \tag{4b}$$

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 2 shows the function of $\Delta\dot{\psi} - \Delta v_y$ computed from the collected dataset.

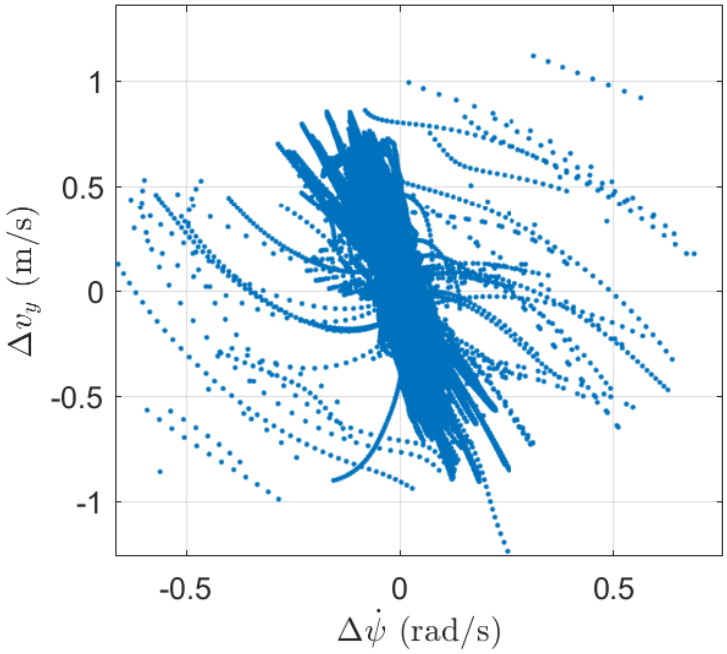


Figure 2. Illustration of the error functions and their resolution.

Since neither $\Delta\dot{\psi}$ nor Δv_y are easy to calculate during the operation of the vehicle, a method must be found that is able to appropriately approximate the error signals using only the available measurements. In this paper the pace-regression algorithm is used for this purpose, which is briefly introduced in the following subsection.

3. Parameters Optimization and Determination of Scheduling Parameters

3.1. Fundamentals of the Applied Machine-Learning-Based Method

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

$$y = X\zeta^* + \epsilon \tag{5}$$

where ϵ denotes the noises, which are computed from $N(0, \sigma^2)$. σ^2 must be determined or estimated ($\hat{\sigma}^2$). The fitted linear model is denoted by $\mathcal{M}(\zeta)$, which has a parameter vector ζ. The goal of the optimization process is to find the best model from the model space $\mathbb{M} = \{\mathcal{M}(\zeta) : \zeta \in \mathbb{R}^k\}$, which has the lowest error on the whole dataset. There are several algorithms, which are able to solve this problem shrinkage, *OLS* subset selection, *CIC*, *RIC*, methods, etc., see [27]. Sometimes, solving this problem may be difficult; therefore, these algorithms can eliminate the redundant and irrelevant variables from the dataset before starting the optimization. The produced models are evaluated by their euclidean distance:

$$\mathcal{D}(\mathcal{M}(\zeta^*), \mathcal{M}(\zeta)) = \frac{||y_{\mathcal{M}(\zeta^*)} - y_{\mathcal{M}(\zeta)}||^2}{\sigma^2} \tag{6}$$

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.

$$\mathcal{D}(\mathcal{M}^*, \mathcal{M}) = min! \tag{7}$$

The formulation of the appropriate model may be difficult when all of the attributes are used. Therefore, it is recommended to create subsets using reduced number of variables to get better results. Suppose that the dataset has k variables, this means that $k+1$ can be created including the null model ($j=0$) and the full model $j=k$. Then the parameter vector for each model ($\mathcal{M}_j$) can be computed as:

$$\hat{\zeta}_{\mathcal{M}_j} = (X'_{\mathcal{M}j} X_{\mathcal{M}j})^{-1} X'_{\mathcal{M}j} y \tag{8}$$

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 $y^*_{\mathcal{M}_j} = \mathcal{P}_{\mathcal{M}_j} y^*$. In order to find the best fitting model, numerous models must be evaluated. Each model contains different set of variables. The number of all the possible combinations of the variables is 2^k. At increased k the computation and the evaluation of 2^k models are not feasible. Therefore, the algorithm reduces the number of the possible cases from 2^k to $k+1$. Firstly, the predefined order of the variables is computed, this order reflects on the correlation between the variables and output. The first variable is the most correlated to the output and the last variable is the least correlated. The ordering can be done by using several algorithms such as [28]. After the ordering process, the algorithm has to evaluate only $k+1$ models starting from the full model (including all variables) to the zero model (no variables). Finally, the best model can be easily chosen from the $k+1$ evaluated models.

The pace regression algorithm 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 $\Delta\hat{\dot{\psi}}, \Delta\hat{v}_y$.

3.2. Parameter Selection of the Control-Oriented Model

The goal of the identification process is to determine the parameters of the model (10) 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 (2). In this case, the lateral model can be formed as:

$$\dot{x}_d = A_d(\rho) x_d + B_d u_d(\rho), \tag{9}$$

where

$$A_d(\rho) = \begin{bmatrix} a_{11}(\rho) & a_{12}(\rho) \\ a_{21}(\rho) & a_{22}(\rho) \end{bmatrix}, \quad B_d(\rho) = \begin{bmatrix} b_1(\rho) \\ b_2(\rho) \end{bmatrix}, \tag{10}$$

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: $\Gamma_{\dot{\psi}}, \Gamma_{v_y}$ as shown in Figure 3. 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.

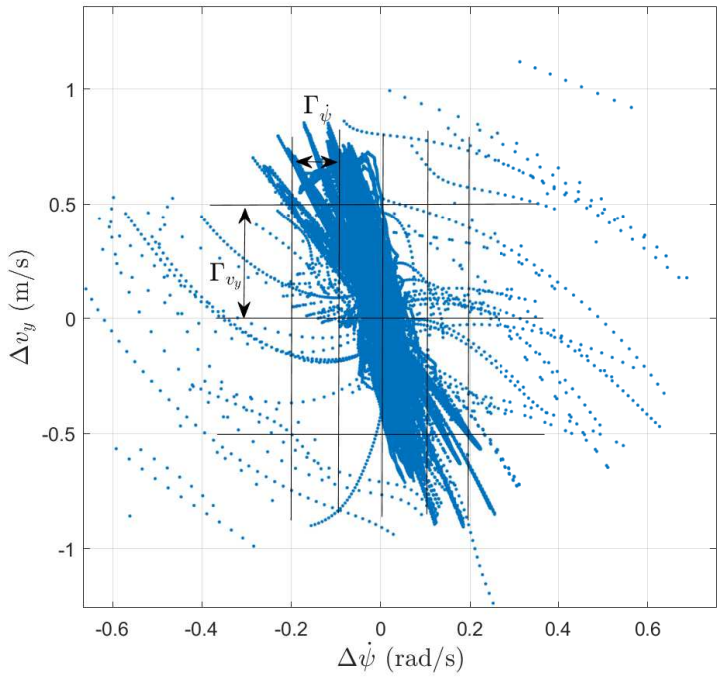


Figure 3. Resolution of the scheduling variables.

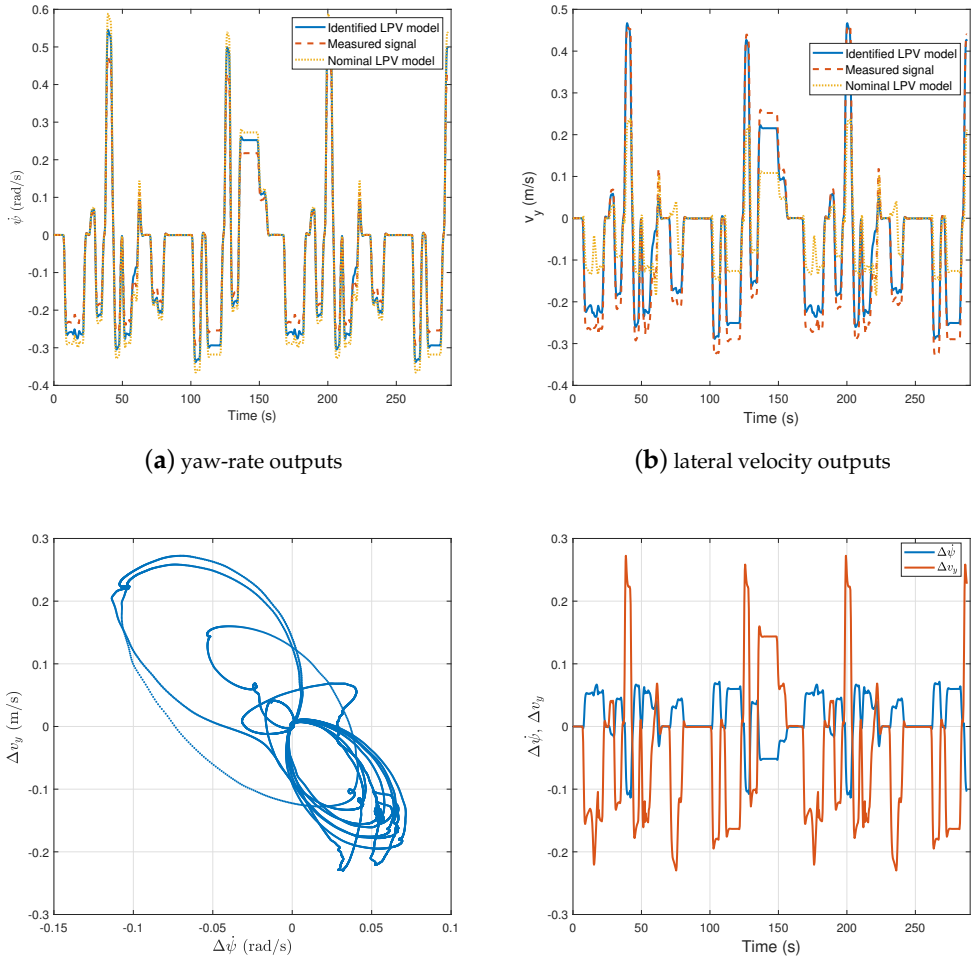
The main goal if the identification process is to determine the parameters $a_{11}(\rho_i)$, $a_{12}(\rho_i), a_{21}(\rho_i), a_{22}(\rho_i)$ and $b_1(\rho_i), b_2(\rho_i)$ for each segment, where ρ_i denotes a specific operating point of the system with fixed ranges of the scheduling parameters. It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

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

where $x_{m,\rho_i}(tji)$ 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 (11) 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 [29,30]. The result of the optimization is a set of polytopic systems, which represents the *LPV* description of the vehicle model.

3.3. Evaluation of the Data-Driven LPV Models

In the following, a test case is presented to show the efficiency of the proposed parameter optimization method. The outputs of the optimized system are compared to the outputs of a nominal model presented in Section 2. The parameters of the nominal model are given by the simulation software (CarSim) such as mass, inertia, geometrical parameters, etc. Specific values can be found in Table 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. In Figure 4, we show the results of the control-oriented *LPV* system for a simulation-based test case. The dashed red lines represent 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 (2)) and blue lines illustrate the outputs of the identified system. As it can be seen, the error between the measured and the computed outputs are smaller when the identified model is used.



(**a**) yaw-rate outputs

(**b**) lateral velocity outputs

(**c**) Δv_y vs $\Delta\dot{\psi}$

(**d**) scheduling parameters with respect to time

Figure 4. Evaluation of the optimized model.

The difference is significant in case of v_y. When the nominal model is used, the averaged error is ≈0.07 m/s, in the case of the identified system it reduces to ≈0.017 m/s. The scheduling parameters are shown in Figure 4c,d. A significant range of scheduling parameters can be seen, which means that the identified system works well at different operating points. It can be said that the identified system together with the selected scheduling parameters fits better to the nonlinear model than the nominal model.

4. Path Following *LPV* Control Design Using the Data-Driven Model

In this section , the *LPV*-based lateral control design is presented. However, in some situations, the intervention is very limited using only the steering angle. It must be noted that, at specific operating points, the performance of the nominal intervention (steering angle) is limited due to the nonlinear dynamics of the vehicle. Therefore, the model is extended with the differential drive has a second input signal, which will be used to compensate the degraded performances of the steering system. This means that the model (2) is modified as:

$$I\ddot{\psi} = C_1\alpha_1 l_1 - C_2\alpha_2 l_2 + M_d \tag{12}$$

where M_d denotes the differential torque. Moreover, the original model contains only two states: yaw-rate and the lateral velocity in order to simplify the optimization task by minimizing the number of the parameters. However, the lateral position tracking is also

an important requirement for the autonomous vehicle, which can be computed from the existing states. Therefore, the state-space model is extended with at the lateral position y, which can be computed as $y(t) = \int_{T=1}^{t} v_y(T)dT$ at tth s.

The identified system description is augmented as:

$$\dot{x} = A(\rho)x + B(\rho)u \tag{13}$$

and

$$A(\rho) = \begin{bmatrix} a_{11}(\rho) & a_{12}(\rho) & 0 \\ a_{21}(\rho) & a_{22}(\rho) & 0 \\ 0 & 1 & 0 \end{bmatrix}, B(\rho) = \begin{bmatrix} b_1(\rho) & \frac{1}{I_z} \\ b_2(\rho) & 0 \\ 0 & 0 \end{bmatrix},$$

where the state vector of the system is $x = \begin{bmatrix} \dot{\psi} & v_y & y \end{bmatrix}^T$, the input is $u = [\delta \quad M_d]$.

The main goal of the control design is to guarantee the trajectory tracking of the vehicle, which can be achieved by describing performances for controller:

- *Minimization of the lateral error*: As mentioned, the goal of the control design is to guarantee the trajectory tracking of the vehicle thus the error between the measured and the reference lateral positions must be minimized:

$$z_2 = y_{ref} - y, \qquad |z_2| \to min, \tag{14}$$

where y_{ref} is the reference lateral position computed from the track.

- *Minimization of the yaw-rate error*: In order to achieve smooth trajectory tracking, a reference yaw-rate is also prescribed, which also must be tracked by the vehicle:

$$z_1 = \dot{\psi}_{ref} - \dot{\psi}, \qquad |z_1| \to min, \tag{15}$$

where $\dot{\psi}_{ref}$ is the reference yaw-rate signal, computed from the curvature of the road, see [26].

- *Minimization of the interventions*: Due to the energy consumption, the interventions also must be minimized during the operation of the vehicle:

$$z_3 = \delta, \qquad |z_3| \to min. \tag{16}$$
$$z_4 = M_d, \qquad |z_4| \to min. \tag{17}$$

The performances are compressed into a performance-vector$z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 \end{bmatrix}^T$, which leads to the following performance equation

$$z = C_1 x + D_{11} r + D_{12} u, \tag{18}$$

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

The basis of the control design is the presented data-driven *LPV* model (13), its measurement and performance equations are:

$$\dot{x} = Ax + Bu, \tag{19a}$$
$$z = C_1 x + D_{11} r + D_{12} u, \tag{19b}$$
$$y_K = C_2 x, \tag{19c}$$

where (19c) represents the measurement of $y_K = \begin{bmatrix} y & \dot{\psi} \end{bmatrix}$.

In order to guarantee the predefined performances ((14)–(16)), the system is augmented with several weighting functions. The augmented system is illustrated in Figure 5. The weighting functions $W_{z,1}, W_{z,2}$ aim to guarantee the tracking performances, while

$W_{z,3}, W_{z,3}$ aim to minimize the interventions. Since the parallel minimization of the performances and the interventions is contradictory, another goal of the weighting functions are to describe a balance between them.

$W_{ref,1}$ and $W_{ref,2}$ are used to scale the references signals ($\dot{\psi}_{ref}, y_{ref}$). Moreover, the role of the functions $W_{w,1}$, $W_{w,2}$ is to attenuate the noises on the measured signals.

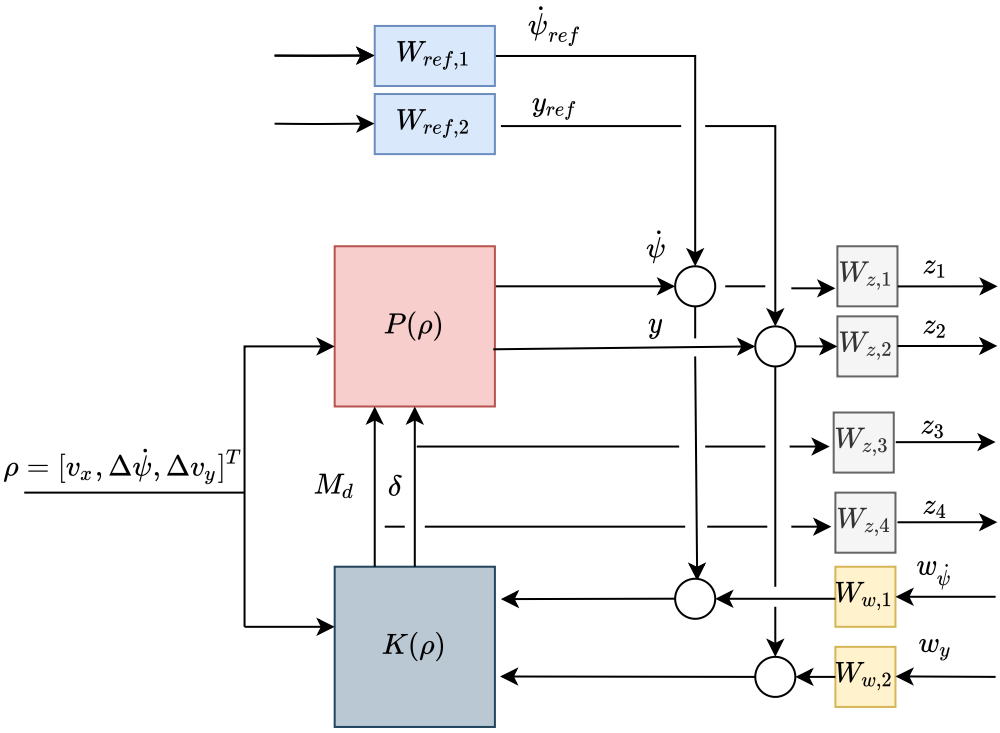


Figure 5. Augmented plant.

The goal of the control design is to find a $K(\rho)$ controller, which can guarantee the predefined performances and by which the closed-loop system is quadratically stable and its induced $\mathcal{L}_2$ norm is smaller than a γ value. The optimization task can be formed as:

$$\inf_{K(\rho)} \sup_{\rho \in F_\rho} \sup_{\|w\|_2 \neq 0, w \in \mathcal{L}_2} \frac{\|z\|_2}{\|w\|_2}, \tag{20}$$

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

$$\dot{x}_K = A_K(\rho)x_K + B_K(\rho)y_K, \tag{21a}$$
$$u = C_K(\rho)x_K + D_K(\rho)y_K, \tag{21b}$$

where $A_K(\rho), B_K(\rho), C_K(\rho), D_K(\rho)$ are variable-dependent matrices. The interconnected system is illustrated in Figure 6 showing all of the main components of the proposed control system.

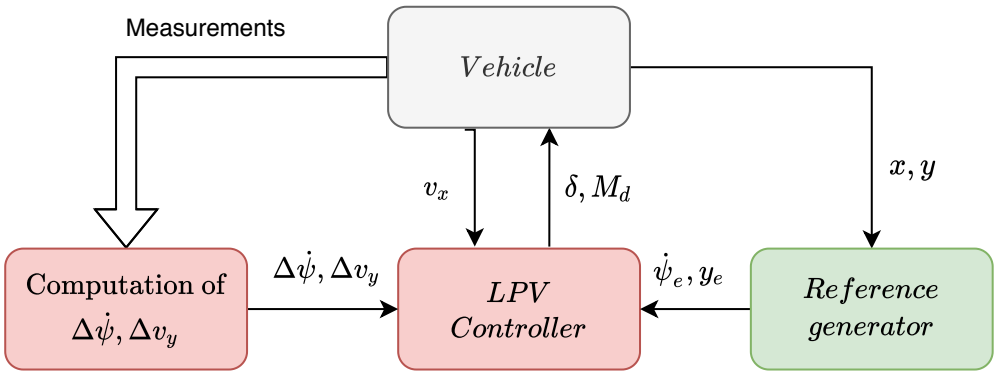


Figure 6. Structure of control system.

5. Simulation Results

In the followings, a complex test scenario is presented to show the efficiency and the operation of the proposed control algorithm in the high-fidelity vehicle dynamics simulation software, CarSim. In the simulations, the vehicle is driven along the Michigan Waterford hill track twice. In the first run, the vehicle is controlled by the proposed data-driven *LPV* controller, while in the second turn, the car is driven by a nominal *LPV* controller. Note that the nominal *LPV* controller is tuned similarly to the proposed *LPV* controller using the same number of weighting functions. However, the weights of the nominal *LPV* controller are optimized to the nominal model presented in Section 2.

The parameters of a car (e.g., mass, inertia, geometric attributes) are listed in Table 1.

Table 1. Parameters of the used D-class vehicle.

Parameter	Notion	Value	Unit
Mass of the car	m	1690	kg
Yaw-inertia	J	4192	kg/m^2
Location of front axis from COG	l_1	1.11	m
Cornering stiffness of front wheels	C_1	155,160	N/rad
Location of rear axis from COG	l_2	1.66	m
Cornering stiffness of rear wheels	C_2	114,659	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

As presented in Section 4, the *LPV* controller contains several weighting functions, which aim to guarantee the predefined performances. The weighting functions of the reference signals are designed in such a way to ensure smooth trajectory tracking, while guaranteeing small tracking error:

$$W_{ref,1} = \frac{0.1}{15\,\mathrm{s}+1}, \quad W_{ref,2} = \frac{0.01}{20\,\mathrm{s}+1}. \tag{22}$$

The goal of the scaling functions $W_{z,1}$, $W_{z,2}$ are to guarantee the trajectory tracking of the vehicle, they are designed by considering the maximal allowable errors, see:

$$W_{z,1} = \frac{5}{0.3\,\mathrm{s}+1}, \quad W_{z,2} = \frac{15}{0.5\,\mathrm{s}+1}. \tag{23}$$

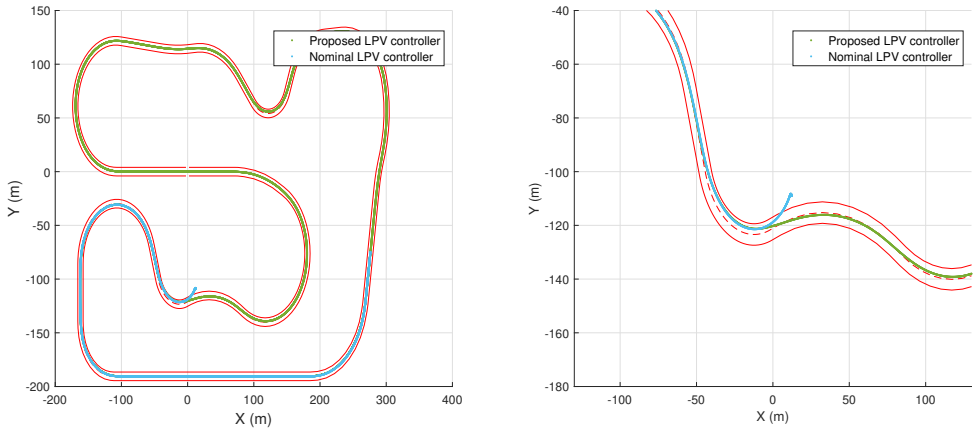
The functions $W_{z,3}$ and $W_{z,2}$ aim to ensure the balance between the interventions and the tracking performances. As it can be seen, the weighting function belonging to the differential drive depends on the scheduling parameters $\Delta\hat{\psi}$ and $\Delta\hat{v}_y$. It is tuned to compensate for the degraded performances of the steering system at the highly nonlinear region.

$$W_{z,3} = 0.1\frac{1\,\mathrm{s}+1}{2\,\mathrm{s}+1}, \quad W_{z,4} = \frac{1}{(10\Delta\hat{\psi}\Delta\hat{v}_y)^2}\frac{1\,\mathrm{s}+1}{4\,\mathrm{s}+1}. \tag{24}$$

Finally, the weighting functions of the measurements are presented. They are used to attenuate the noises on the measured signals, which may have a negative effect on the performances.

$$W_{w,1} = 0.02, \quad W_{w,2} = 0.05. \tag{25}$$

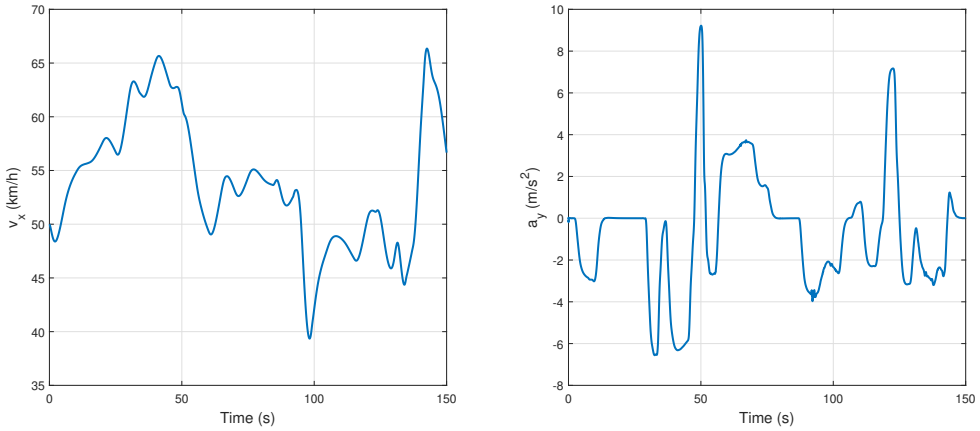
In the followings, the results of the test scenarios are presented. As mentioned, the vehicle has been driven along the track twice. The track and the trajectories of the vehicle are illustrated in Figure 7. As it can be seen, when the vehicle is controlled by the nominal *LPV* controller, it leaves the road at a sharp bend. Whilst, in the second case, when the car is controlled by the proposed data-driven *LPV* controller, it follows the road throughout the whole track.



(**a**) Whole track (**b**) Highlighted section

Figure 7. Positions of the vehicles during the simulations.

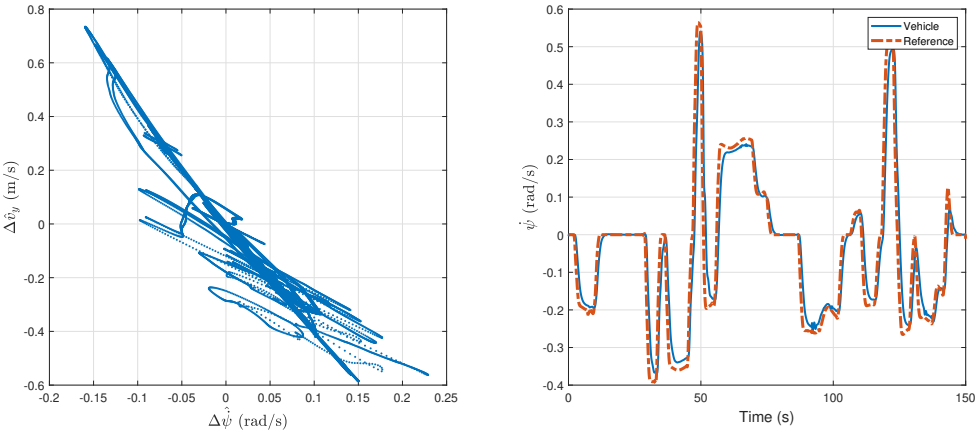
In order to show the operation of the proposed control method in a wide range, the longitudinal velocity of the vehicle varies between $v_x \in$ <40, 73> km/h, as shown in Figure 8a. Figure 8 demonstrates the lateral acceleration of the vehicle. As it can be seen, the maximum of the lateral acceleration is ≈10 m/s^2, which means that the vehicle is close to its physical limits. This situation cannot be handled by the nominal controller; thus, the vehicle leaves the road.



(**a**) Velocity profile (**b**) Lateral acceleration

Figure 8. Velocity profile and lateral acceleration of the vehicle.

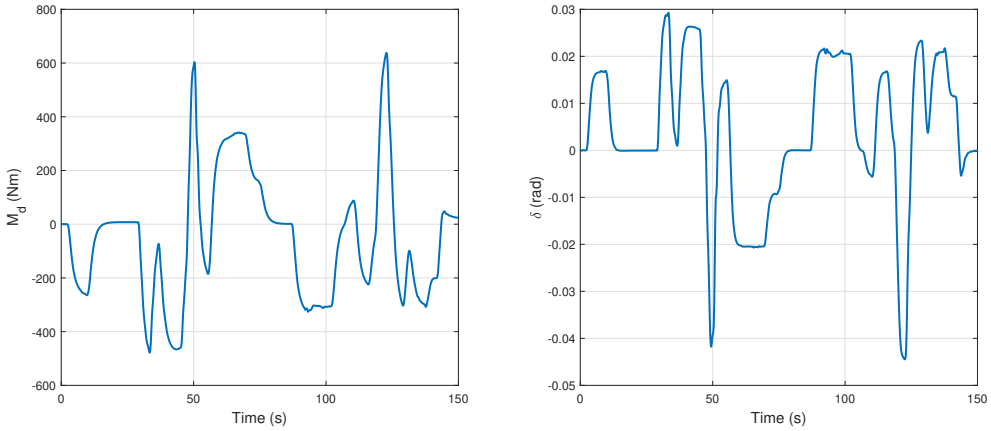
Figure 9 shows the scheduling parameters and the yaw-rate tracking of the vehicle. As Figure 9a demonstrates, the scheduling parameters cover almost its whole range, as shown in Figure 4. The yaw-rate tracking of the vehicle can be seen in Figure 9b. The tracking is accurate the maximum error is only ≈0.02 rad/s. The yaw-rate signal varies in a wide range, its maximum is close to 0.6 rad/s, which is also close to the physical limits of the vehicle.



(**a**) Scheduling parameters (**b**) Tracking of yaw-rate

Figure 9. Scheduling parameters and tracking of yaw-rate.

Finally, the interventions are illustrated in Figure 10. Figure 10a show the differential torque computed by the proposed control algorithm. As it can be seen, its maximal value is ≈600 Nm, which is provided at the sharp bend to compensate the steering angle. The steering angle is depicted in the right figure. It varies between $\delta \in <-0.04, 0.03>$ rad, which is a reasonable range for the presented velocity profile.



(**a**) Differential torque (**b**) Steering angle

Figure 10. Interventions of the vehicles.

6. Conclusions

In the paper, a novel data-driven identification and control method has been proposed for autonomous vehicles. The identification process was based on the *LPV* framework, in which the scheduling parameters have been computed by a machine learning algorithm (Pace-regression). The efficiency of the identification process has been demonstrated through an example, and the identified *LPV* model was the basis of the lateral control design. The operation and the efficiency of the proposed control system has been demonstrated through a complex simulation example performed in the high-fidelity simulation software, CarSim. The simulation example has shown that the proposed control algorithm provided better performances than a nominal one, especially in extreme maneuvers. The scheduling parameters of the system covered a wide range, which meant that the the proposed control algorithm was able to work in the whole range of the vehicle's operation range. The main drawback of the proposed algorithm is that it requires a reasonably large dataset in order to provide the appropriate model for the lateral control design. Moreover, the weighting functions of the controller must be tuned in case of the identified systems, since the parameters of the system can significantly differ from the nominal model. However, as shown in the simulation, the identified system-based controller provides remarkably better performances than the nominal controller.

Author Contributions: Conceptualization, algorithms, software, D.F.; methodology, D.F. and B.N.; supervision, P.G. All authors have read and agreed to the published version of the manuscript.

Funding: The research reported in this paper and carried out at BME has been supported by the National Research Development and Innovation Fund (TKP2020 IES, Grant No. BME-IE-MIFM) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology. The research presented in this paper, carried out by Institute for Computer Science and Control was supported by the Ministry for Innovation and Technology and the National Research, Development and Innovation Office within the framework of the National Lab for Autonomous Systems. The work of D. Fényes was supported by the ÚNKP-20-3 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.The work of B. Németh was partially supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and the ÚNKP-20-5 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gáspár, P.; Szabó, Z.; Bokor, J.; Németh, B. *Robust Control Design for Active Driver Assistance Systems*; Springer: Berlin/Heidelberg, Germany, 2017.
2. Pham, T.P.; Sename, O.; Dugard, L. Real-time Damper Force Estimation of Vehicle Electrorheological Suspension: A NonLinear Parameter Varying Approach. Part of special issue: 3rd IFAC Workshop on Linear Parameter Varying Systems LPVS 2019: Eindhoven, Netherlands, 4–6 November 2019. *IFAC-PapersOnLine* **2019**, *52*, 94–99. [CrossRef]
3. Rosolia, U.; Borrelli, F. Learning How to Autonomously Race a Car: A Predictive Control Approach. *IEEE Trans. Control. Syst. Technol.* **2020**, *28*, 2713–2719. [CrossRef]
4. Nemeth, B.; Gaspar, P.; Peni, T. Nonlinear analysis of vehicle control actuations based on controlled invariant sets. *Int. J. Appl. Math. Comput. Sci.* **2016**, *26*, 31–43. [CrossRef]
5. Ribeiro, A.M.; Fioravanti, A.R.; Moutinho, A.; de Paiva, E.C. Nonlinear state-feedback design for vehicle lateral control using sum-of-squares programming. *Veh. Syst. Dyn.* **2020**, 1–27. [CrossRef]
6. Yang, G.; Zhao, Y. Motion Stability Analysis of Vehicle with Four Wheel Steering System Considering Tire Nonlinearity. In Proceedings of the 2010 3rd International Congress on Image and Signal Processing (CISP2010), Yantai, China, 16–18 October 2010; pp. 3433–3437.
7. Sadri, S.; Wu, Q. Lateral Stability Analysis of On-road Vehicles Using Lyapunov's Direct Method. In Proceedings of the 2012 Intelligent Vehicles Symposium, Alcala de Henares, Spain, 3–7 June 2012; pp. 821–826.
8. Huijun, G.; Weichao, S.; Shen, Y.; Okyay, K. Stability Control for Lateral Vehicle Motion with Uncertain Parameters and External Nonlinearities. In Proceesings of the 32nd Chinese Control Conference, Xi'an, China, 26–28 July 2013.
9. Corno, M.; Panzani, G.; Roselli, F.; Giorelli, M.; Azzolini, D.; Savaresi, S.M. An LPV Approach to Autonomous Vehicle Path Tracking in the Presence of Steering Actuation Nonlinearities. *IEEE Trans. Control. Syst. Technol.* **2020**, *24*, 956–970. [CrossRef]
10. Gaspar, P.; Szabo, Z.; Bokor, J. A grey-box identification of an LPV vehicle model with side slip angle estimation. In Proceedings of the 2007 American Control Conference, Zurich, Switzerland, 4–7 September 2007.
11. Rodonyi, G.; Bokor, J. Uncertainty Identification for a Nominal LPV Vehicle Model Based on Experimental Data. In Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain, 15 December 2005; pp. 2682–2687.
12. Kuutti, S.; Bowden, R.; Jin, Y.; Barber, P.; Fallah, S. A Survey of Deep Learning Applications to Autonomous Vehicle Control. *IEEE Trans. Intell. Transp. Syst.* **2020**, 1–22. [CrossRef]
13. Hubschneider, C.; Bauer, A.; Doll, J.; Weber, M.; Klemm, S.; Kuhnt, F.; Zollner, J.M. Integrating end-to-end learned steering into probabilistic autonomous driving. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–7.
14. Rausch, V.; Hansen, A.; Solowjow, E.; Liu, C.; Kreuzer, E.; Hedrick, J.K. Learning a deep neural net policy for end-to-end control of autonomous vehicles. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 4914–4919.
15. Pomerleau, D. Knowledge-Based Training of Artificial Neural Networks for Autonomous Robot Driving. *Robot. Learn.* **1993**, *233*, 13–43.
16. Cavanini, L.; Ferracuti, F.; Longhi, S.; Monteriu, A. LS-SVM for LPV-ARX Identification: Efficient Online Update by Low-Rank Matrix Approximation. In Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 1–4 September 2020; pp. 1590–1595.
17. Romano, R.A.; dos Santos, P.L.; Pait, F.; Perdicoulis, T.; Ramos, J.A. Machine learning barycenter approach to identifying LPV state-space models. In Proceedings of the 2016 American Control Conference (ACC), Boston, MA, USA, 6–8 July 2016; pp. 6351–6356. [CrossRef]
18. Bao, Y.; Velni, J.M. Data-Driven Linear Parameter-Varying Model Identification Using Transfer Learning. *IEEE Control. Syst. Lett.* **2021**, *5*, 1579–1584. [CrossRef]
19. Abdufattokhov, S.; Muhiddinov, B. Stochastic Approach for System Identification using Machine Learning. In Proceedings of the 2019 Dynamics of Systems, Mechanisms and Machines (Dynamics), Omsk, Russia, 5–7 November 2019; pp. 1–4.
20. Biagetti, G.; Crippa, P.; Falaschetti, L.; Turchetti, C. Machine learning regression based on particle bernstein polynomials for nonlinear system identification. In Proceedings of the 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP), Tokyo, Japan, 25–28 September 2017; pp. 1–6. [CrossRef]
21. Rosolia, U.; Borrelli, F. Learning Model Predictive Control for Iterative Tasks. A Data-Driven Control Framework. *IEEE Trans. Autom. Control.* **2018**, *63*, 1883–1896. [CrossRef]
22. Fliess, M.; Join, C. Model-free control. *Int. J. Control.* **2013**, *86*, 2228–2252. [CrossRef]
23. Formentin, S.; De Filippi, P.; Corno, M.; Tanelli, M.; Savaresi, S.M. Data-Driven Design of Braking Control Systems. *IEEE Trans. Control. Syst. Technol.* **2013**, *21*, 186–193. [CrossRef]
24. Palmieri, G.; Baric, M.; Glielmo, L.; Borrelli, F. Robust vehicle lateral stabilisation via set-based methods for uncertain piecewise affine systems. *Veh. Syst. Dyn.* **2012**, *50*, 861–882. [CrossRef]
25. Fenyes, D.; Nemeth, B.; Gaspar, P. Analysis of autonomous vehicle dynamics based on the big data approach. In Proceedings of the European Control Conference, Limassol, Cyprus, 12–15 June 2018; pp. 219–224.
26. Rajamani, R. *Vehicle Dynamics and Control*; Springer: Berlin/Heidelberg, Germany, 2005.

27. Wang, Y.; Witten, I.H. *Pace Regression*; (Working Paper 99/12); University of Waikato, Department of Computer Science: Hamilton, New Zealand, 1999.
28. Shibata, R. An optimal selection of regression variables. *Biometrika* **1981**, *68*, 45–54. [CrossRef]
29. Gill, P.E.; Murray, W.; Wright, M. *Practical Optimization*; Academic Press: London, UK, 1981.
30. Coleman, T.F.; Li, Y. A Reflective Newton Method for Minimizing a Quadratic Function Subject to Bounds on some of the Variables. *SIAM J. Optim.* **1996**, *6*, 1040–1058. [CrossRef]