2020 American Control Conference
Denver, CO, USA, July 1-3, 2020

LPV-based autonomous vehicle control using the results of big data analysis on lateral dynamics

Dániel Fényes, Balázs Németh, Péter Gáspár

Abstract— **The paper presents a new big data based control design for autonomous vehicles. The main contribution of this work is the longitudinal velocity optimization process, which is based on the approximation of the reachability sets of a passenger vehicle by using a machine-learning approach. The data, which is used for the approximation, is provided by the high-fidelity car simulation software, CarSim. The approximation is performed by applying a well-known decision tree algorithm, C4.5. The reachability sets are computed for different longitudinal velocities. Moreover, a LPV technique based lateral control design is proposed, which is used to guarantee the trajectory tracking of the vehicle. To enhance the capability of the LPV controller, the control scheme is extended with the longitudinal velocity optimization process. Thus, the stable and safe motion of the vehicle is guaranteed.**

I. Introduction and motivation

The development of the autonomous vehicles has brought new challenges for the whole automotive industry and the control engineers. The automated vehicles requires more accurate sensors, decision making algorithms and advanced control solutions. Since the dynamics of the vehicle is highly nonlinear and cannot be modeled accurately due to the uncertainties, the conventional control methods cannot provide robust and reliable solutions for this problem. In the recent years, the new artificial intelligence (AI) based and other traditional machine learning (or big data) methods have become popular among control engineers. They can be used for a variety of problems including: state estimation, object recognition and even for control problems, such as tracking of a predefined signal, see e.g., [1], [2], [3]. Most of the machine learning and artificial intelligence methods require a lot of data for teaching their models. Fortunately, the modern vehicles are getting equipped with more and more sensors, which measure a lot of information about the car. From the measurements of the individual vehicles, gigantic databases can be created. These databases can provide good bases for machine learning algorithms and big data analyses, see [4].

Several papers utilize these huge amount of information. For example in [5], a pace regression based solution can found for the estimation of the side-slip angle of the vehicle. Another example is given in [6], which describes a machine learning algorithm based method for the determination of the road surface. The machine learning methods can also be used for computing the reachability sets of the vehicle, see [7]. Furthermore, [8] presents an MPC (Model Predictive Control) scheme, which is extended with the result of the big data analysis. This paper presents a method to determine the reachability sets of the vehicle, which are used in the computation of the optimal longitudinal velocity profile. Furthermore, a Linear Parameter Varying (LPV) based lateral control design is proposed for autonomous vehicles. The control structure is extended with a velocity profile optimization process, which can guarantee the safe and stable motion of the vehicle. The velocity optimization process utilizes the reachability sets of the vehicle, which are determined by using a machine learning method, more precisely the decision tree algorithm, called C4.5. The paper describes the acquisition of the data, which is the basis of the determination of the reachability sets. Furthermore, the prepocess steps of the acquired data are also presented. The main contributions of the paper are the LPV-based lateral control design and the longitudinal velocity optimization algorithm.

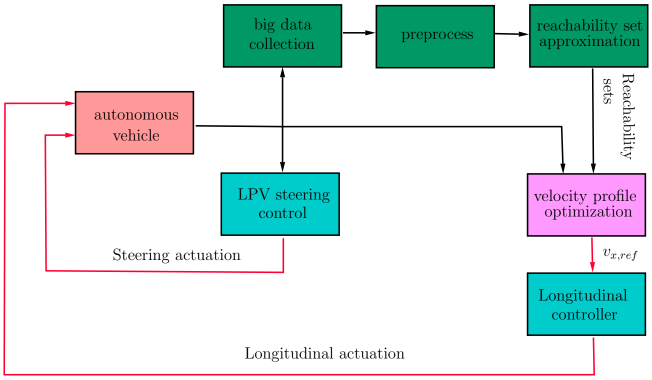


Fig. 1. Scheme of the proposed control structure

The structure of the proposed autonomous vehicle control solution is illustrated in Figure 1. As a first step, the collection of the dataset must be done. Then, the collected raw data has to be prepared for the further data analysis steps.

B. Németh, P. Gáspár and D. Fényes are with Systems and Control Laboratory, Institute for Computer Science and Control, Hungarian Academy of Sciences, Kende u. 13-17, H-1111 Budapest, Hungary. E-mail: [daniel.fenyes;balazs.nemeth;peter.gaspar]@sztaki.mta.hu

The work was supported by the research program titled ”Exploring the Mathematical Foundations of Artificial Intelligence (2018-1.2.1-NKP-00008).

The research was supported by the Hungarian Government and co-financed by the European Social Fund through the project ”Talent management in autonomous vehicle control technologies” (EFOP-3.6.3-VEKOP-16-2017-00001).

The work of B. Németh was partially supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and the UNKP-19-4 New National Excellence Program of the Ministry for Innovation and Technology. The work of D. Fényes was partially supported by the UNKP-19-3 New National Excellence Program of the Ministry for Innovation and Technology.

978-1-5386-8266-1/$31.00 ©2020 AACC

The prepared dataset is used for computing the reachability sets of the vehicle, which will serve as the basis of the velocity optimization algorithm. The block 'LPV steering control' represents the design of a lateral tracking control algorithm, which is based on the LPV framework. Whilst, the 'Longitudinal controller' is responsible for guaranteeing the realization of the computed optimal velocity profile. The longitudinal controller is based on a simple PID structure, therefore it's not detailed in this paper.

The remainder of the paper is structured as follows. Section II presents the data acquisition, the preprocess of the data and the computation of the reachability sets. The next section details the LPV-based lateral control design and the longitudinal velocity optimization process. Finally, the last section shows a comprehensive simulation to show the operation and effectiveness of the proposed control scheme.

II. Big data analysis on lateral dynamics

In this section, the determination of the reachability sets of the vehicle is presented by using a machine learning approach. For the computation of the reachability sets a well-known decision tree algorithm is applied, which is called C4.5. Furthermore, the acquisition of the datasets and the prepocess of the raw data are also detailed in this section.

Acquisition of dataset

The first step of any data-based analysis is the acquisition of the appropriate dataset. In this research, the dataset is provided by the high fidelity simulation software, CarSim. In the simulations the test vehicle has been controlled by the CarSim's built-in driver model. Several simulations have been conducted using different parameter sets. During the simulations the following parameters have been changed: longitudinal velocity, adhesion coefficient, simulation track. From the simulations, only those variables/measurements have been save, which could be measured in all modern cars, such as:

- Longitudinal velocity (v_x)
- Yaw-rate ($\dot{\psi}$)
- Accelerations (a_x, a_y)
- Wheel speeds ($\omega_{fl}, \omega_{fr}, \omega_{rl}, \omega_{rr}$)
- Side-slip angle (β) (for illustration purposes)
- Adhesion coefficient μ (Simulation parameter)

In this manner, more than 10 million instances have been collected and saved.

Prepocessing of dataset

The measurements and other information, which are provided by car's sensors, require some preprocess steps before using them in any machine learning algorithms. These steps, for example, includes the selection of the relevant signals (variable). Since not all of the measured attributes must be used (or necessary) for a given machine learning based problem, the selection of the most significant ones is a crucial step. Another problem is the selection of the stable instances. Since the reachability sets contain only the stable states of a given system, an appropriate criterion must be found to split the measured instances into stable and unstable groups. In the literature, several criteria can be found that can give information on the stability of the vehicle e.g. [9], [10]. Unfortunately, they do not fit for this classification problem.

Therefore, a new criterion is used, which is based on the linearized one-track lateral model, see [11], [12], [13]. The basic idea behind this criterion is that the motion of the vehicle is considered to be stable, when its dynamical behavior is close to the linear region. Thus, the criterion is formed as the deviation of the actual vehicle motion and the linear model, which can be expressed as:

$$-\varepsilon_1 < \frac{|1+\alpha_1|}{|1+\delta-\beta-\frac{l_1\dot{\psi}}{v_x}|} - 1 \leq \varepsilon_1, \tag{1}$$

where α_1 is the averaged side-slip angle of the front wheels, δ is the averaged steering angle of the front wheels, β denotes the side-slip angle of the CoG of the vehicle, $\dot{\psi}$ represents the yaw-rate of the vehicle, l_1 is the distance between the front axis and the CoG and v_x is the longitudinal velocity of the vehicle. Moreover, ε_1 is an experimentally defined parameter. An instance is considered to be stable, if it satisfies the inequality (1), while in the other case, it said to be unstable.

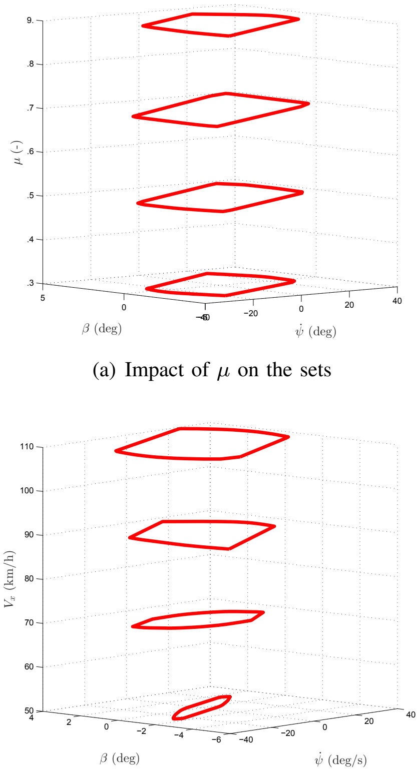
Approximation of the reachability sets

As mentioned earlier, the so-called C4.5. decision tree algorithm is used to determine the reachability sets of the vehicle. In this subsection a brief introduction is given to this method, see [14], [15]. The result of the algorithm is a decision tree, which can determine that whether an instance belongs to the group, which contains the stable instances, or not. A decision tree has three main components:

1) **Nodes:** They represents a condition e.g.: the value of an variable is bigger or smaller than a given value. The outcome of a node can lead to another node or to a leaf.
2) **Branches:** The nodes and the leaves are connected by branches. A branch connects the outcome of a node to another node or to a leaf.
3) **Leaves**: Finally, the leaves determine the class of the instances. In this case, the leaves separate the instances into stable and unstable groups.

The C4.5 algorithm was developed by R. Quinlan in 1993, detailed descriptions can be found in [15] and [16]. The decision tree algorithm C4.5 uses the metric called information gain to optimize the generated decision tree. Furthermore, a crucial parameter of the algorithm is the 'number of minimal objects per leaf'. This parameter influences the size of the generated tree. If the value of this parameter is set to a small number, in general, the resulted tree consists of a huge number of nodes and leaves. In the other case, the number of the components can be reduced by setting this parameter to a small value. Therefore, the size of the decision tree can be controlled by this parameter, which is especially useful when the decision trees are used in any control system.

The results of the decision tree are illustrated in Figures 2. Only the instances, which are classified as stable by the decision tree, are depicted. Moreover, in the figures the stable instances are approximated by a convex hull, which is a conservative inner approximation of the reachability sets. As the figures shows the reachability sets, which are illustrated in the plane of $\dot{\psi}$ (yaw-rate) and β (side-slip angle) with respect to the adhesion coefficient (μ) in the top figure and to the longitudinal velocity (v_x) in the second figure. It can be observed that the shapes and the sizes of the reachability sets can change along with the velocity and the adhesion coefficient. This means that the intervention of the vehicle must be fitted to the actual circumstances.



(a) Impact of μ on the sets

(b) Impact of v_x on the sets

Fig. 2. Illustration of the reachability set approximations

III. LPV-based Vehicle Control Design and Velocity Selection Strategy

This section focuses on the lateral control design and longitudinal velocity optimization for autonomous vehicles. The main goal of these algorithms is to create an enhanced control scheme, which is able to guarantee the safe and stable motion of the vehicle. The lateral control design is based on the LPV (Lateral Parameter Varying) approach, in which the longitudinal velocity is used as a scheduling parameter. Whilst, in the velocity optimization process, the presented reachability sets are utilized to compute the optimal velocity profile.

Design of lateral LPV control

In the lateral control design, the one-track lateral vehicle model is used, whose detailed description can be found in [11]. The model consists of the following three equations:

$$mv_x(\dot{\psi}+\dot{\beta}) = F_y(\alpha_1)+F_y(\alpha_2), \tag{2}$$

$$J\ddot{\psi} = F_y(\alpha_1)l_1 - F_y(\alpha_2)l_2, \tag{3}$$

$$\dot{v}_y = v_x(\dot{\psi}+\dot{\beta}), \tag{4}$$

where β denotes the side-slip angle of the vehicle, $\dot{\psi}$ represents the yaw-rate of the vehicle, v_x and v_y are the longitudinal and lateral velocities of the vehicle, l_1, l_2 are the distances between the axes and CoG of the vehicle, while m is the mass of the vehicle and J is the yaw inertia. Furthermore, F_y denotes the lateral force of the tires, see [17]. They depend the side slips of the tires (α_1, α_2). In this case, this relationship is considered to be linear, therefore it can be approximated as $F_y = C\alpha_i$. C is the cornering stiffness.

Using these equations, the following state-space representation can be derived:

$$\dot{x} = A(\rho_1)x + B(\rho_1)u, \tag{5}$$

whose state-vector consists of $x = \begin{bmatrix}\dot{\psi} & \beta & v_y & y\end{bmatrix}^T$, the actuation is $u = \delta$ and $\rho_1 = v_x$ is the scheduling variable. Moreover, $A(\rho_1)$, $B(\rho_1)$ are matrices of system (5). The lateral control design has two main goals: 1./ To ensure the safe and the stable trajectory tracking of the vehicle, 2./ To minimize the actuation of the system. These goals form the following performances:

- *Minimization of the lateral error:*
  One of the main goals of the controller is to guarantee the trajectory tracking of the vehicle, which means that the controller must minimize the error between the actual lateral position (y) and reference signal (y_{ref}):

$$z_1 = y_{ref} - y, \qquad |z_1| \rightarrow min. \tag{6}$$

- *Minimization of the actuation:*
  The mentioned trajectory tracking must be reached by using minimal energy (intervention):

$$z_2 = \delta, \qquad |z_2| \rightarrow min. \tag{7}$$

These performances are summed up by the performance vector: $z = \begin{bmatrix}z_1 & z_2\end{bmatrix}^T$, which forms the performance equation:

$$z = C_1x + D_{11}r + D_{12}u, \tag{8}$$

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

The LPV-based control design aims to find a controller that can ensure the balance between the presented performances and the external noises. The optimal balance between the performances can be reached by using scaling or weighting functions. The structure of the closed-loop system is illustrated in Figure 3. As it can be seen, several weighting functions are used to guarantee the predefined performances of the system. The goal of $W_{ref,1}$ is to weight the reference signal in such a manner to ensure the smooth tracking. W_{z_1} represents the weight of the first performance, which means it is responsible for the tracking of the lateral position.

The weighting function W_{z_2} is to minimize the intervention (steering angle) of the system. Finally, $W_{\omega,1}$ attenuates the external noise on the measured signal.

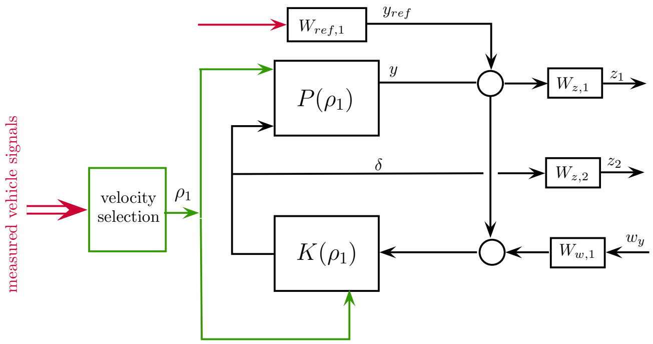


Fig. 3. Closed-loop interconnection structure

The quadratic LPV performance problem is to choose the parameter-varying controller $K(\rho_1)$ 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:

$$\inf_{K(\rho_1)} \sup_{\rho_1,\in F_\rho} \sup_{\|w\|_2\neq 0, w\in\mathcal{L}_2} \frac{\|z\|_2}{\|w\|_2}, \tag{9}$$

where F_ρ bounds the scheduling variables. The optimization problem is solved by using the LPVTools in MATLAB environment, see [18]. The yielded controller $K(\rho_1)$ is formed as

$$\dot{x}_K = A_K(\rho_1)x_K + B_K(\rho_1)y_K, \tag{10}$$
$$u = C_K(\rho_1)x_K + D_K(\rho_1)y_K, \tag{11}$$

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

$$y_K = C_2 x + D_{21} r, \tag{12}$$

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 look-up-tables, see [19], [20].

Design of the optimal velocity profile

The goal of the velocity profile optimization is to guarantee the safe and stable motion of the vehicle. It means that the state of the vehicle remain close to the linear region of its dynamics. This condition can be ensured by keeping the states of the vehicle inside the reachability sets. Of course, the ultimate goal is to maximize the longitudinal velocity, while keeping the states inside the reachability sets.

For guaranteeing this criterion, the prediction of the motion of the vehicle, especially the states $\dot{\psi}$ (yaw rate) and β (side-slip angle) must be computed. This can be made by forming the closed-loop system using the nominal LPV model (2) and the controller (10).

$$\dot{x}_{cl} = A_{cl}(\rho_1)x_{cl} + B_{cl}(\rho_1)r, \tag{13}$$

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

$$A_{cl}(\rho_1) = \begin{bmatrix} A_{cl,11} & A_{cl,12} \\ A_{cl,21} & A_{cl,22} \end{bmatrix}, \tag{14}$$

$$B_{cl}(\rho_1) = \begin{bmatrix} B(\rho_1)D_K(\rho_1)D_{21} \\ B_K(\rho_1)D_{21} \end{bmatrix}. \tag{15}$$

with the following matrices

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

In order to predict the motion of the vehicle, the presented closed-loop system must be discretized using the sample time T, detailed explanation can be found in [21]:

$$x_{cl}(k+1) = A_{cl}(k)x_{cl}(k) + B_{cl}(k)r(k), \tag{16}$$
$$y_{cl}(k) = C_{cl}x_{cl}(k), \tag{17}$$

where $A_{cl}(k), B_{cl}(k)$ are used as compact notations of $A_{cl}(\rho_1(k))$ and $B_{cl}(\rho_1(k))$. Moreover, $y_{cl}(k)$ contains $\dot{\psi}$ and β, and C_{cl} is the corresponding matrix. The prediction of $y_{cl}(k)$ can be computed on a n-steps long horizon as:

$$\boldsymbol{y_{cl}(k,n)} = \begin{bmatrix} y_{cl}(k+1) \\ y_{cl}(k+2) \\ \vdots \\ y_{cl}(k+n) \end{bmatrix} = \mathcal{A} + \mathcal{B}R, \tag{18}$$

where

$$\mathcal{A} = \begin{bmatrix} C_{cl}A_{cl}(k) \\ C_{cl}A_{cl}(k)A_{cl}(k+1) \\ \vdots \\ C_{cl}\prod_{i=k}^{k+n} A_{cl}(i) \end{bmatrix} x_{cl}(k) \tag{19}$$

$$\mathcal{B} = \begin{bmatrix} C_{cl}B_{cl}(k) & \cdots & 0 \\ C_{cl}A_{cl}(k)B_{cl}(k) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ C_{cl}\prod_{i=k}^{k+n-1} A_{cl}(i)B_{cl}(k) & \cdots & C_{cl}B_{cl}(k) \end{bmatrix} \tag{20}$$

$$R = \begin{bmatrix} r(k+1) \\ \vdots \\ r(k+n) \end{bmatrix}. \tag{21}$$

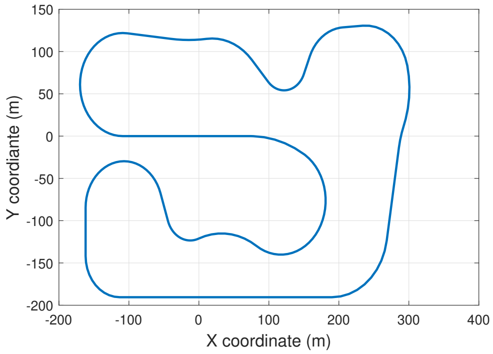


Fig. 4. Michigan track

Using these notations, the optimization problem can be formed. The main goal of the velocity profile design is to maximize the elements of the vector $\boldsymbol{\rho} = \left[\rho_1(k+1)\dots\rho_1(k+n)\right]^T$, which contains the reference longitudinal velocity on the forthcoming road segments. Of course, the elements of this vector have upper bounds, which can be handled as constraints during the optimization process

$$\boldsymbol{\rho} \leq \boldsymbol{\rho_{max}}. \tag{22}$$

$\boldsymbol{\rho_{max}}$ contains the upper bounds of the longitudinal velocity on the forthcoming road horizon. The stability of the vehicle can be guaranteed by ensuring that the predicted states of the vehicle $\boldsymbol{y_{cl}(k,n)}$ remain inside the reachability sets, presented in Section II. This criterion can be formed as:

$$\boldsymbol{y_{cl}(k,n)} \in \mathcal{R}\big(\rho_1(i)\big), \qquad \forall k \leq i \leq n. \tag{23}$$

Then, the final optimization task is:

$$\max_{\rho_1(k+1)\dots\rho_1(k+n)} \boldsymbol{\rho} \tag{24}$$

subject to the constraints (22), (23):

$$\boldsymbol{\rho} \leq \boldsymbol{\rho_{max}} \tag{25}$$
$$\boldsymbol{y_{cl}(k,n)} \in \mathcal{R}\big(\rho_1(i)\big), \qquad \forall k \leq i \leq n. \tag{26}$$

The solution of the optimization is a vector, which contains the optimized velocity profile for n steps ahead.

IV. Simulation examples

In this section a comprehensive simulation examples is presented to show the operation and the effectiveness of the proposed control algorithm. The analysis has been performed using the Weka data mining software [22], while the dataset has been collected through the CarSim vehicle dynamic simulator.

The vehicle, which is used in the simulation, a D-class passenger car, whose mass is $m = 2015kg$. During the simulation, the vehicle is driven along a segment of Michigan Waterford Hills Road Racing track twice, see IV. In the first run, the vehicle uses the optimized velocity profile, while in the second simulation, the car uses the nominal velocity profile.

The results of the simulations are illustrated in the following figures. Figure 5 shows the lateral error for both cases. As it can be seen the vehicle, which uses the original velocity profile, leaves the road at $120m$. At that station, a sharp bend begins, which cannot be tracked by the vehicle with the predefined velocity profile. In the first case, the car is able to track the road at the sharp bend since the original velocity profile is overwritten by the velocity optimization process.

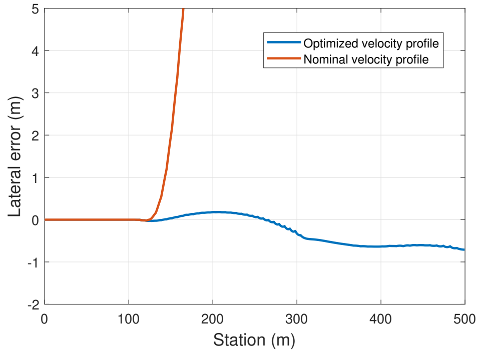


Fig. 5. Lateral error of the vehicles on Waterford Hills track

The modified velocity profile is illustrated in Figure 6. It can be seen that the algorithm reduces the longitudinal velocity after $6s$, which reflects on the mentioned sharp bend.

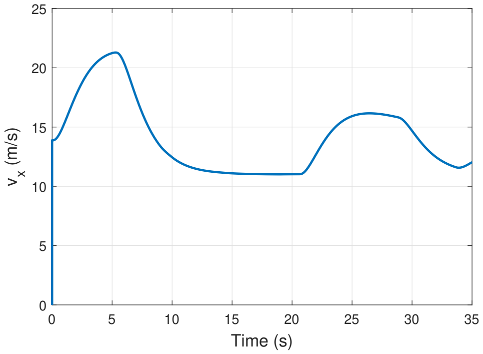


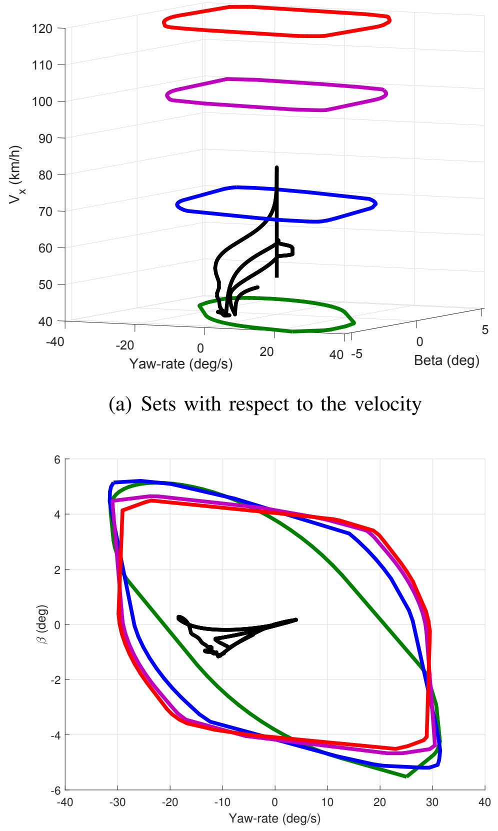
Fig. 6. Velocity profiles

The Figures 7 show the states of the vehicle in the plane of $\dot{\psi}$ (yaw-rate) and β (side-slip angle) for that case when the modified velocity profile is used. As the figures indicate, the states of the vehicle do not leave the stable region of its states, therefore the stability of the vehicle is guaranteed throughout the simulation. Furthermore, this is the reason why the vehicle does not leave the road in the first case.

Finally, the Figure 8 shows the steering actuation in the simulation. The steering angle is between $[-4, 1deg]$, which is a reasonable range for this system.

V. Conclusions

The paper has presented a LPV bases lateral control design for autonomous vehicles. The control system has been



(a) Sets with respect to the velocity

(b) Sets in the plane of $\dot{\psi}$ and β

Fig. 7. Reachabaility sets and the trajectory of the vehicle

extended with a longitudinal velocity profile optimization process. This algorithm utilized the reachability sets of the vehicle, which had been determined by using a decision tree algorithm, called C4.5. Furthermore, a comprehensive simulation example has been presented to show the operation and the effectiveness of the proposed control algorithm. In the simulations, two cases have been compared to each other. In the first case, the vehicle has used the original velocity profile, while in the second case the car has been driven by the modified profile. It has been shown that without the modified velocity the vehicle was not able to follow the predefined path.

References

[1] J. Jeon, W. Lee, H. J. Cho, and H. Lee, "A big data system design to predict the vehicle slip," in *15th Int. Conf. Control, Automation and Systems (ICCAS)*, 2015.

[2] H. A. Najada and I. Mahgoub, "Autonomous vehicles safe-optimal trajectory selection based on big data analysis and predefined user preferences," in *7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference*, 2016.

[3] M. Zhu, X. Y. Liu, M. Qiu, R. Shen, W. Shu, and M. Y. Wu, "Traffic big data based path planning strategy in public vehicle systems," in *24th Int. Symp. Quality of Service*, 2016.

[4] W. Xu, H. Zhou, N. Cheng, F. Lyu, W. Shi, J. Chen, and X. Shen, "Internet of vehicles in big data era," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 19–35, Jan 2018.

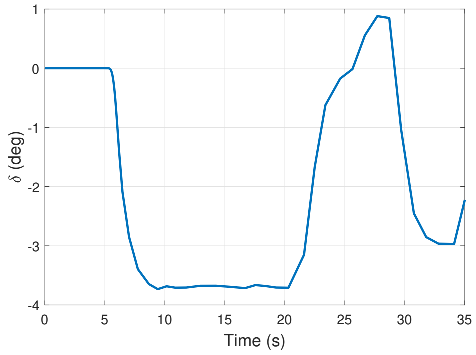


Fig. 8. Steering angle actuation

[5] D. Fényes, B. Németh, M. Asszonyi, and P. Gáspár, "Side-slip angle estimation of autonomous road vehicles based on big data analysis," in *26th Mediterranean Conference on Control and Automation*, 2018.

[6] D. Fenyes, B. Nemeth, P. Gaspar, and Z. Szabo, "Road surface estimation based lpv control design for autonomous vehicles," *Proc. 3rd IFAC Workshop on Linear Parameter Varying Systems, Eindhoven, Netherlands*.

[7] D. Fenyes, B. Nemeth, and P. Gaspar, "A predictive control for autonomous vehicles using big data analysis," *Proc. 9th IFAC Symposium on Advances in Automotive Control, Orleans, France*, 2019,.

[8] ——, "Impact of big data on the design of mpc control for autonomous vehicles," *Proc. 18th European Control Conference Napoli, Italy*, 2019.

[9] S. Sadri and C. Wu, "Stability analysis of a nonlinear vehicle model in plane motion using the concept of lyapunov exponents," *Vehicle System Dynamics*, vol. 51, no. 6, pp. 906–924, 2013.

[10] 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*, vol. 26, no. 1, 2016.

[11] R. Rajamani, "Vehicle dynamics and control," *Springer*, 2005.

[12] D. Fényes, B. Németh, and P. Gáspár, "Analysis of autonomous vehicle dynamics based on the big data approach," in *European Control Conference*, 2018.

[13] M. Masouleh and D. Limebeer, "Region of attraction analysis for nonlinear vehicle lateral dynamics using sum-of-squares programming," *Vehicle System Dynamics*, vol. 56, no. 7, 2018.

[14] E. B. Hunt, *Concept Learning: An information Processing Problem*. Wiley, 1962.

[15] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, California: Morgan Kaufmann Publishers, 1993.

[16] ——, "Bagging, Boosting, and C4.5," *Proceedings of the Thriteenth National Conference on Artificial Intelligence*, pp. 725–730, 1996.

[17] R. Ghandour, A. Victorino, M. Doumiati, and A. Charara, "Tire/road friction coefficient estimation applied to road safety," *18th Mediterranean Conference on Control and Automation*, 2010.

[18] G. Balas, A. Hjartarson, A. Packard, and P. Seiler, "A toolbox for modeling, analysis, and synthesis of parameter varying control systems," 2015.

[19] F. Wu, X. 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*, vol. 6, pp. 983–988, 1996.

[20] 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*, vol. 19, no. 1, pp. 104–117, 2011.

[21] R. Tóth, *Modeling and Identification of Linear Parameter-Varying Systems*, ser. Lecture Notes in Control and Information Sciences. Berlin, Heidelberg: Springer, 2010, vol. 403.

[22] I. Witten and E. Frank, *Data Mining Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, Elsevier, 2005.