Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-35 (2022) 97–102

Robust LPV control synthesis for learning-aided driver assistance systems

Balázs Németh *, **Attila Lelkó** *, **Péter Gáspár** *

* *Systems and Control Laboratory, Institute for Computer Science and Control (SZTAKI), Eötvös Loránd Research Network (ELKH), Kende u. 13-17, H-1111 Budapest, Hungary.*
E-mail: [balazs.nemeth;attila.lelko;peter.gaspar]@sztaki.hu

Abstract: This paper proposes a control design framework with guarantees for systems, which contains learning-based control elements. The framework is based on a supervisory control structure, which contains a supervisor, a robust Linear Parameter Varying (LPV) controller and the learning-based control elements. This paper presents the design of a lateral path following control for driver assistance systems, which is aided with a learning-based agent. In the paper the formulation of the supervisor, the design method of the robust LPV controller are provided, while the learning-based agent via an imitation learning process is considered to be given. The effectiveness of the method on driver-in-the-loop simulation scenario is demonstrated. It is shown that the proposed control system is able to provide guarantee on the limitation of the path following error, and to guarantee transition between the driver steering actuation and the automated control intervention.

Keywords: automated vehicles, learning and control, robust LPV design, driver assistance

1. INTRODUCTION AND MOTIVATION

Guaranteeing safe motion of automated vehicles has the highest priority over all of the control performance requirements. Nevertheless, the control design for complex systems, which contains learning-based control elements, e.g., learning-based agents, is a challenging task. Moreover, in case of automated vehicles, the elements in the control architectures during the lifetime of the vehicle can vary, e.g., through an update process. Thus, a further challenge is to develop control design frameworks, with which guarantees on the safety performances against the variation of some control elements can be achieved.

In case of these challenges, the concept of plug and play control has indicated a direction for research (Stoustrup [2009]). In vehicle control context various partial results have been published. Paper Gangadharan et al. [2016] has focused on safety features of vehicles, if the vehicle through different elements based on the consumer's request is composed. In Lin et al. [2021] a software architecture is proposed, in which a hypervisor manages the operating systems of different vehicle controls. Possibility of plug and play design in Qazi et al. [2020] from the viewpoint of transportation is evaluated, and in Li et al. [2021] the concept is developed for unmanned aerial vehicles. Although all of these methods are useful for solving special control problems, a comprehensive solution on the problem of providing safety performances guarantees for systems with varying control elements has not been published. Nevertheless, an actual problem of automated vehicles is to guarantee robust performance of the vehicle, even if the learning-based agent of the vehicle is modified, e.g., it is retrained or updated during service time.

In this paper a novel control design framework based on the Linear Parameter Varying (LPV) method is proposed, with which guarantees on primary, i.e., safety performances can be provided. In the proposed concept the control loop can contain learning-based control elements, and in the control design a priory knowledge on the internal structures of the learning-based elements is not used. The advantage of the proposed design framework is the it uses only the outputs of the learning-based control elements, and thus, the modifications of these elements have not impact on safety performance requirements. The proposed design framework is based on a supervisory control structure. The control input of the systems is computed by the supervisor based on the output signal of the learning-based control element and the output of a robust LPV control. This paper focuses on the design of the LPV control and the supervisor, while the learning-based element is considered to be given.

Although some preliminary results can be found in Németh and Gáspár [2021], this paper provides new achievements on the application of the method for steering control of automated vehicles. Furthermore, another preliminary work in the topic of vehicle control with learning-based

[1] The research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems (RRF-2.3.1-21-2022-00002). The paper was partially funded by the National Research, Development and Innovation Office (NKFIH) under OTKA Grant Agreement No. K 135512.

10.1016/j.ifacol.2022.11.296

approaches can be found, see Németh [2021]. Nevertheless, the methodology of that paper is different, i.e., the robust control is considered to be designed, and that work focuses on the learning-based designed. This paper provides solution on the problem, when the learning-based agent is given, and the problem is the design of the robust controller.

The paper is organized as follows. The fundamentals of the framework are presented in Section 2. In Section 3 the design method is developed for the path following problem. The effectiveness of the method through simulation examples is demonstrated in Section 4, and finally, the paper is concluded in Section 5.

2. ROBUST CONTROL FRAMEWORK WITH LEARNING-BASED AGENT IN THE CONTROL LOOP

The aim of the concept is to form the structure of the robust design framework with which requirements on the primary, i.e., safety performances can be guaranteed. The idea behind the framework is that the control input of the system is equivalent to the output of the learning-based agent, if the requirements on the primary performances can be guaranteed. But, if the primary performances are violated, the output of the learning-based control is overridden by the supervisor. The decision about the violation of the primary performances is performed through the comparison of the output of the learning-based agent and that of the robust LPV controller.

The output of the learning-based controller is vector u_L with n elements as $u_L = \mathcal{F}(y_L) = [u_{L,1}\ u_{L,2}\ \dots\ u_{L,n}]^T$, where y_L vector contains the inputs of the controller with m_L elements. $\mathcal{F}$ represents the learning-based controller itself, e.g., a deep-learning neural network, which is activated on its input layer. Moreover, the output of a robust LPV controller is u_K with n elements as $u_K = \mathcal{K}(\rho, y) = [u_{K,1}\ u_{K,2}\ \dots\ u_{K,n}]^T$, where $\mathcal{K}$ represents the LPV controller and y_K is the vector of the measured signals with m_K elements. Moreover, $\rho \in \varrho$ vector contains the scheduling variables of the controller, which is considered to have at least n elements.

The fundamental assumption of the design method is that the control input signal of the system $u = [u_1\ u_2\ \dots\ u_n]^T$ can be expressed as a function of u_K in a linear form, under predefined conditions. The parameters in the linear formulation are selected to guarantee $u = u_L$ if the requirements on the primary performances are guaranteed by u_L. Thus, the relationship between u, u_K and u_L with the conditions is formed as

$$\begin{aligned} u = & u_L = u_K + \Delta_L^*, \\ & \text{if} \quad \Delta_{L,i}^* \in \Lambda_{L,i}, \quad \forall i = 1 \dots n, \end{aligned} \tag{1}$$

where Δ_L^* is $n \times 1$ vector as

$$\Delta_L^* = \left[\Delta_{L,1}^*\ \Delta_{L,2}^*\ \dots\ \Delta_{L,n}^*\right]^T, \tag{2}$$

and $\Delta_{L,i}^*$, $i = 1 \dots n$ are time-dependent weighting signals, $\Lambda_{L,i} = [\Delta_{L,i,min}; \Delta_{L,i,max}]$ represent domains with $\Delta_{L,i,min}$, $\Delta_{L,i,max}$ scalars. The set of the domain is denoted by Λ_L.

If the condition of (1) for $\Delta_{L,i}^*$ is guaranteed, the control input of the system u is equal to u_L. But, if there exists at least one $i \in [1; n]$, where $\Delta_{L,i}^* \notin \Lambda_{L,i}$, the variables $\Delta_{L,i}^*$ are limited with the boundaries of $\Lambda_{L,j}$ during the computation of the control signal u_i. In this case $u \neq u_L$.

The general control rule, which contains both cases is formed as

$$u = u_K + \Delta_L, \tag{3}$$

where

$$\Delta_L = [\Delta_{L,1}\ \Delta_{L,2}\ \dots\ \Delta_{L,n}]^T, \tag{4a}$$

$$\Delta_{L,i} = \min\Big(\max\left(\Delta_{L,i}^*; \Delta_{L,i,min}\right); \Delta_{L,i,max}\Big), \quad \forall i = 1 \dots n. \tag{4b}$$

The relation (4b) guarantee that $\Delta_L \in \Lambda_L$. The minimum performance level is determined by the LPV controller in the entire operation domain of the system, while inside of the domain Λ_L the performance level is enhanced through learning-based control. Thus, the advantages of learning-based control can be achieved, while its drawback, such as performance degradation in some scenarios is eliminated through the minimum performance level.

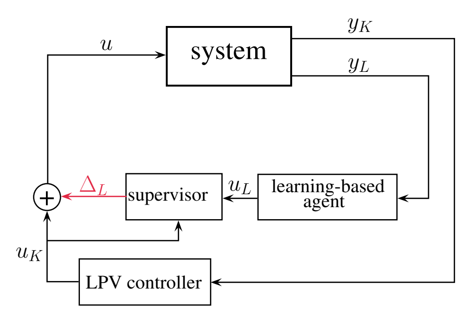


Fig. 1. Structure of the control architecture

In Figure 1 the structure of the given control architecture is presented. In the proposed concept the feedback loop contains the LPV controller, while the learning-based controller is in an auxiliary loop from the control aspect. The role of the supervisor block is to select Δ_L, with which the difference between u and u_L can be minimized, but the primary performance is guaranteed. Thus, the optimization task of the supervisor is $\min\limits_{\Delta_{L,i}} \sum\limits_{i=1}^{n} (u_{L,i} - u_i)^2$ subject to $\Delta_{L,i} \in \Lambda_{L,i}$ $\forall i \in n$ and to the primary performance requirements.

3. APPLICATION OF THE DESIGN FOR ACHIEVING PATH FOLLOWING FUNCTIONALITY

In this section the design of the robust control and the supervisor for the specific problem of path following is proposed.

3.1 Design of the robust control

The control design is based on the lateral model of the vehicle:

$$J\ddot{\psi} = C_1 l_1\left(\delta - \frac{v_y + \dot{\psi} l_1}{v}\right) - C_2 l_2\left(-\frac{v_y - \dot{\psi} l_2}{v}\right), \quad (5a)$$

$$m a_y = C_1\left(\delta - \frac{v_y + \dot{\psi} l_1}{v}\right) + C_2\left(-\frac{v_y - \dot{\psi} l_2}{v}\right), \quad (5b)$$

$$v_y = \dot{y}, \quad (5c)$$

where $\dot{\psi}$ is yaw-rate, v_y is lateral velocity, δ is front wheel steering angle, v is longitudinal velocity, a_y is lateral acceleration of the vehicle and y is its lateral position. Moreover, C_1, C_2 are cornering stiffness on the front and on the rear axle, l_1, l_2 are distances of the front or rear axles from the vehicle center of gravity. The dynamics is reformulated to state-space representation $\dot{x} = A(\rho)x + B_2 u$, where the state vector is $x = \begin{bmatrix} v_y & \dot{\psi} & y \end{bmatrix}^T$ and the control input is $u = [\delta]$, $\rho = v$ as scheduling variable is selected.

The steering control input u from the candidate control input from the learning-based agent u_L and from the output of the robust controller u_K are composed as follows

$$u = u_K + \Delta_L, \quad (6)$$

see (3). Thus, in this application example $n = 1$ due to the single control input of system. The transformation of the state-space equation through (6) results in the system representation

$$\dot{x} = A(\rho) + B_2\Delta_L + B_2 u_K. \quad (7)$$

The primary, i.e., safety performance of the system is to guarantee the limitation of the lateral error of the vehicle from the centerline of the road:

$$z_1 = y_{ref} - y; \qquad |z_1| \to min, \quad (8)$$

where y_{ref} is the reference lateral position for the vehicle. Moreover, the limitation of the steering angle is requested to avoid the unwanted effect of actuator saturation, which leads to the further performance:

$$z_2 = \delta; \qquad |z_2| \to min. \quad (9)$$

The performance vector $z_K = \begin{bmatrix} z_1 & z_2 \end{bmatrix}^T$ through the state-space equation (5) can be expressed as

$$z_K = C_2 x + D_{21} y_{ref} + D_{22} u, \quad (10)$$

which can be reformulated through (6), such as

$$z_K = C_2 x + D_{22} w + D_{22} u_K, \quad (11)$$

where $w = \begin{bmatrix} y_{ref} & \Delta \end{bmatrix}^T$. Similarly, the formulation of measurement $y_K = y_{ref} - y$ is expressed as

$$y_K = C_1 x + D_{11} w + D_{12} u_K. \quad (12)$$

The control-oriented state-space representation of the system from the dynamics, performances, measurements on the system is composed, such as

$$\dot{x} = A(\rho) + B_2\Delta_L + B_2 u_K, \quad (13a)$$
$$y_K = C_1 x + D_{11} w + D_{12} u_K, \quad (13b)$$
$$z_K = C_2 x + D_{22} w + D_{22} u_K. \quad (13c)$$

The system (2) is parameter-dependent with disturbance vector w, whose impact on the performance vector z must be minimized. Therefore, the robust LPV design method the control synthesis is selected, which is able to provide the stability of the closed loop system together with disturbance attenuation (Bokor and Balas [2005]). Scaling of disturbances and performances is requested for the robust LPV design, and thus, the plant (13) is augmented with weighting functions, see Figure 2. The system (13)

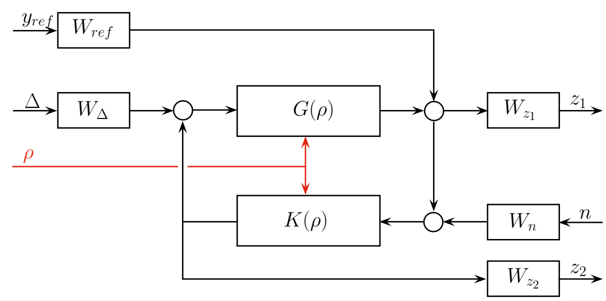


Fig. 2. Illustration of the augmented plant

is represented as $G(\rho)$ and the controller is $K(\rho)$. The reference signal y_{ref} is scaled with the function $W_{ref} = \frac{y_{ref,max}}{T_{ref}s+1}$. It represents that for steady-state scenario the maximum of the reference signal is $y_{ref,max}$, and moreover, its variation can have a dynamics with T_{ref} time constant. Performance z_1 is also scaled with a transfer function to represent the allowed dynamics of the tracking error. In the function of $W_{z_1} = \frac{1/e_{max}}{T_e s+1}$, e_{max} represents maximum lateral error in steady-state and T_e represents the time constant of the tracking error dynamics. Moreover, in the augmented plant the further weights are selected to be scalar values. $W_\Delta = \Delta_{max}$ is related to the bound of Δ, W_n is the weighting function of the sensor noise. Finally, $W_{z_2} = \frac{1}{\delta_{max}}$ scales the control input, whose maximum is allowed to be δ_{max}.

The goal of the design method of the LPV control is to provide the quadratically stability of the closed-loop system and the induced $\mathcal{L}_2$ norm from the disturbance vector w to z is less than the scalar $\gamma > 0$. 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. Gridding reflects the qualitative changes in the performance weights. The stability and the performance level of the closed-loop system are guaranteed by the design procedure (Wu et al. [1996]). The quadratic LPV performance problem is to choose the parameter-varying controller $\mathcal{K}(\rho, y)$ 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:

$$\inf_{\mathcal{K}(\rho,y)} \sup_{\rho\in\varrho} \sup_{\substack{\|w\|_2 \neq 0,\\ w \in \mathcal{L}_2}} \frac{\|z\|_2}{\|w\|_2}. \quad (14)$$

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. Finally, the state-space representation of the LPV control $\mathcal{K}(\rho, y)$ is constructed, see Wu et al. [1996], Sename et al. [2013]. The optimization problem (14) is solved offline and the resulted controller is $\mathcal{K}(\rho, y)$ implemented for online control input computation. It leads to the control input u_K, which is incorporated in the computation of u (3) together with the selection of Δ_L. The control rule results in that the minimum performance level of the closed-loop system is determined by $\mathcal{K}(\rho, y)$.

A challenge of the control design is the relationship between the design of $\mathcal{K}(\rho, y)$ and the selection of Λ_L. The scaling of the performance level through the design of $\mathcal{K}(\rho, y)$ requires the preliminary selection of Λ_L. Similarly, the scaling of the performance level through the selection of Λ_L requires the preliminary design of $\mathcal{K}(\rho, y)$. A possible solution to the problem is to create an iterative design process in which the control design, the selection of Λ_L and the optimization in the supervisor are ordered and performed in an iterative method.

The goal of the iteration is to minimize the difference between u, u_L and to minimize the path following error $y_{ref} - y$. The minimization of $u_L - u$ leads to increased $\Delta_{L,max}$, because u_L can significantly differ from u_K, i.e., due to the increased robustness requirement a more conservative controller $\mathcal{K}(\rho, y)$ is achieved. The minimization of the path following error can lead to reduced Δ_L, because in this case u is close to u_K and thus, high value for $\Delta_{L,max}$ is unnecessary. The minimization of the differences is achieved through an iterative process, in which a balance between the differences is also achieved. The following optimization task is formed

$$\min_{\Delta_{L,max}} \sum_{k=0}^{N} \Big(D|u_L(k) - u(k)| + |y_{ref}(k) - y(k)| \Big), \quad (15)$$

where N reflects to the horizon, on which the solution of the minimization problem is searched. $\bar{E}$ notes path tracking error. Moreover, $D > 0$ scalar is design parameter. The role of D parameter is to scale $|u_L - u|$ and to guarantee a balance between the two terms of the cost. Since high $\Delta_{L,max}$ can result in increased robustness requirement, it can result in problem in the feasibility of the LMIs in the design of the LPV control. Therefore, it is necessary to limit D.

The solution of the optimization problem (15) begins with domains with high ranges, which are reduced through the following iteration process.

(1) The domain $\Lambda_L = [-\Delta_{L,max}; \Delta_{L,max}]$ is selected high in the first step. Initially, $\Delta_{L,max}$ is selected high, which results in a conservative LPV control. The goal of the iterative design process is to reduce the conservativeness through the appropriate selection of the boundaries.
(2) The LPV control with the selected domains is designed using (14).
(3) The closed-loop system with the incorporation of the designed $\mathcal{K}(\rho, y)$ and the domain Λ_L is analyzed through various scenarios. It yields the signals u_L and u_K.
(4) Due to the results of the scenarios the boundaries are modified to reduce the cost function of the optimization problem (15). The new value of $\Delta_{L,max}$ is selected by the optimization algorithm, e.g., through simplex search or trust-region-reflective methods (Lagarias et al. [1998], Coleman and Li [1996]).
(5) The LPV design, the scenarios and the evaluation (steps 2-4) are performed until the minimum of (15) is reached. If the minimum performance level of the designed control is not suitable, or the range of the domain results in frequent control intervention on the bounds, the parameter D must be modified (step 1) and the iteration must be performed again.

The results of the iteration process are the robust LPV controller and the domain Λ_L.

3.2 Formulation of the supervisory optimization

The objective of the optimization is to minimize $(u - u_L)^2$, i.e., $(\delta_K + \Delta_L - \delta_L)^2$ using (6). The constraint of the optimization reflects to the primary performance criteria, i.e., predicted lateral error through a predefined e_{max} scalar value must be limited. The prediction of the lateral error at preview time T_p is formed through the motion prediction of the vehicle:

$$\psi(k+1) = \psi(k) + v(k)\frac{\tan\delta(k)}{L}T_p, \quad (16a)$$
$$X(k+1) = X(k) + v(k)\cos(\psi(k+1))T_p, \quad (16b)$$
$$Y(k+1) = Y(k) + v(k)\sin(\psi(k+1))T_p, \quad (16c)$$

where k reflects to the actual signals and $k+1$ to the predicted vehicle states, such as X, Y position. In practice, the predicted lateral error $e(k+1)$ can be calculated through a search method, where the goal is to find minimum difference between $X(k+1), Y(k+1)$ and the coordinates of the set of forthcoming waypoints:

$$e(k+1) = \min_{i \in I_p} \sqrt{(X(k+1) - X_i)^2 + (Y(k+1) - Y_i)^2}, \quad (17)$$

where I_p represents the set of candidate waypoints. The predicted lateral error depends on $\delta(k)$, i.e., on $\rho\delta_K + \Delta_L$ through $X(k+1), Y(k+1)$.

The optimization problem of the supervisor is formed as

$$\min_{\Delta_L} (\delta_K + \Delta_L - \delta_L)^2, \quad (18a)$$
$$\text{s.t.} \quad e(k+1) \le e_{max}, \ \Delta_L \in \Lambda_L. \quad (18b)$$

The solution of (18) requires the solution of two optimization process, which are in a hierarchical structure. In the outer optimization loop the task is to minimize the objective $(u - u_L)^2$, and in the inner optimization loop, for all candidate Δ_L the minimization task (17) must be solved.

4. SIMULATION RESULTS

In the example the steering inputs of the neural network and the driver are combined. The contribution of the simulation examples is that the limitation of the lateral error ($e_{max} = 4m$) through the proposed design framework under various configurations can be guaranteed.

4.1 Driver-in-the-loop simulation environment

In this simulation input u_L is provided by a neural network. The goal of the neural network is to imitate driver steering characteristics, which has been learned from measured data. In some papers the problem of imitation learning for autonomous vehicles have already been studied. The advantage of imitation learning is to use neural networks, which which the characteristics of the driver can be achieved. Through the fitting of neural networks it is possible to handle control problems, in which nonlinearities have high impacts. For example, an end-to-end learning method in Codevilla et al. [2018] has been proposed, where a vision-based steering control through conditional imitation learning has been achieved. Paper Kebria et al. [2020] has proposed a general framework for the selection of convolutional neural network parameters in case of deep imitation learning problems. Moreover, in Pan et al. [2020] imitation learning has been applied for agile autonomous driving, which provides special challenges under extreme driving situations.

In this simulation example a medium-size passenger vehicle in CarMaker is selected. The driving of the vehicle through the driver model of CarMaker has been performed, and thus, data on the vehicle motion and on the steering intervention has been collected. For the fitting of the neural network, the following input signals on the neural network have been used:

- actual lateral error of the vehicle,
- actual longitudinal velocity of the vehicle,
- actual and forthcoming curvature of the road in $1s$ horizon with $0.1s$ consecutive steps.

Moreover, the output of the neural network is the generated steering angle of the driver, which must be fitted to the steering angle of the measured dataset. In this example 2 hidden layers with 20 neurons in each layers have been selected. The learning process has been performed through Levenberg-Marquardt algorithm, see Demut et al. [1997], Xu and Chen [2008].

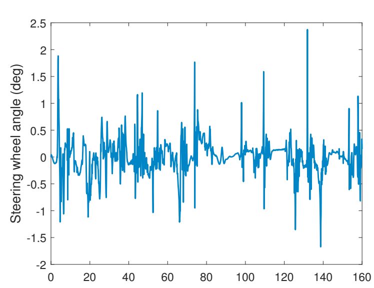


Fig. 3. Result on steering wheel fitting error

In the simulation setup the neural network and the driver intervention are simultaneously involved. The goal of this setup is to handle the driving situations, when the driver's hands on the steering wheel are not hold. Thus, the vehicle control provides steering angle, but this steering angle through the driver can be modified. The advantage of this control solution compared to the previous setup is that the driver has the ability to modify the trajectory of the vehicle, if the driver is not satisfied with its motion. Moreover, the functionality of driving transfer between the driver and the automated system through this setup can be tested, which is an important aspect of automated driving Kaustubh et al. [2016], Molnar et al. [2018].

The structure of the simulation setup is illustrated in Figure 4. The steering angle of the front wheels depends on the rotation angle of the steering wheel. Thus, in this setup δ is not realized on the front wheels of the vehicle directly, but the electric motor of the steering wheel is rotated to achieve δ on the wheels. Nevertheless, if the driver can decide to modify the angle of steering wheel, and thus, the driving is transferred. Torque T for the steering wheel actuator through a PID control from the difference of steering wheel angle and δ is computed.

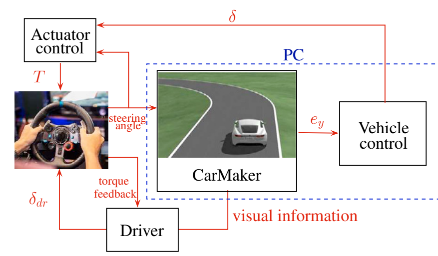
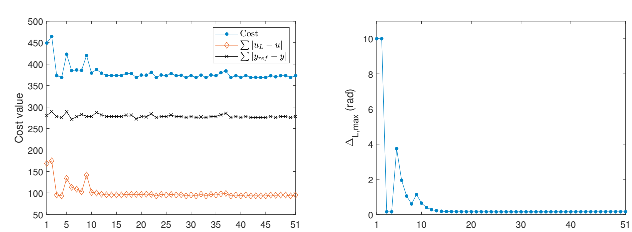


Fig. 4. Scheme of driver-in-the-loop with neural network simulation setup

4.2 Results of the iteration

In Figure 5 the results of the iteration in the control design process are shown. The value of the cost, together with its components are illustrated in Figure 5(a). The initial value of $\Delta_{L,max}$ is 10, which is a high value, as it is recommended in Section 3. During the iteration, in one iteration step the vehicle performs a $1km$ long road section with varying velocity and with various curves. In the simulation example D scalar parameter is selected as 1. It can be seen that the cost decreases during the iteration process, until the minimum value is achieved. Similarly, the variation of $\Delta_{L,max}$ is also reduced, see Figure 5(b), where the achieved value is $0.15rad$.

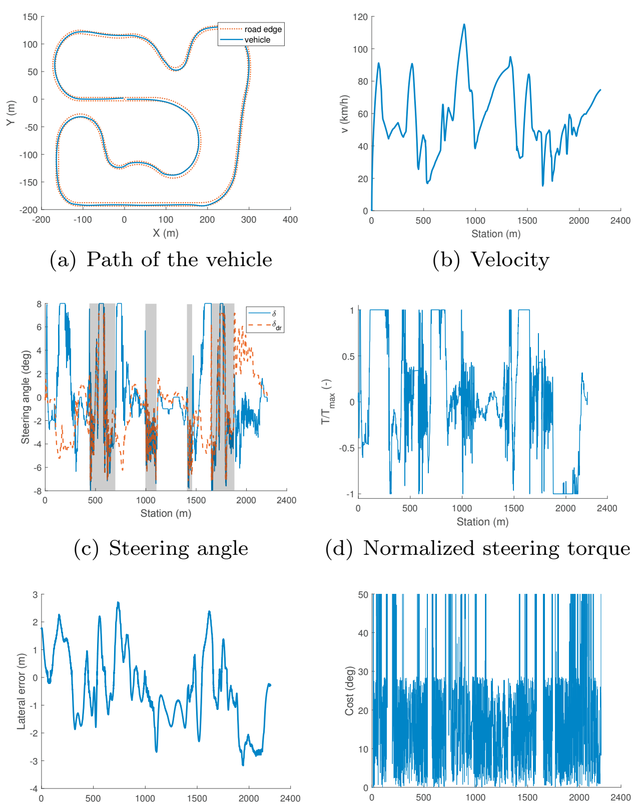


(a) Iteration results of the cost (b) Iteration results of $\Delta_{L,max}$

Fig. 5. Results of the iteration

4.3 Results of the simulation

Some results of the simulation are found in Figure 6. The path of the vehicle during the Race Track is illustrated in Figure 6(a) The velocity selection of the driver through the pedals of the simulator is illustrated in Figure 6(b). The steering angle values δ and δ_{dr} are shown in Figure 6(c). The shaded parts of the figure reflects to the sections of the vehicle path, where $\delta_{dr} = \delta$, i.e., the hands of the driver have not been on the steering wheel. On the further road sections the driver modifies the turning of the steering angle, and thus, δ differs from δ_{dr}. The normalized value of the steering torque on the steering wheel T is illustrated in Figure 6(d). The characteristics of T is close to δ, because the goal of T is to realize δ on the steering wheel. The lateral error during the vehicle motion is illustrated in Figure 6(e). It can be seen that the lateral error is under e_{max}, independently from holding the hand of the driver on the steering wheel or not. The value of the cost function of the supervisor is illustrated in Figure 6(f). The results show that the cost, i.e., the difference between δ and δ_L is small, if the driver steering intervention is close to δ_L (e.g., between $1100m \dots 1400m$), or if the driver does not actuate steering (e.g., between $450m \dots 650m$).



(a) Path of the vehicle (b) Velocity

(c) Steering angle (d) Normalized steering torque

(e) Lateral error (f) Cost value

Fig. 6. Simulation results with driver intervention

5. CONCLUSIONS

In the paper the design method and the operation of the proposed control framework have been presented. Through simulation examples with learning-based agent and with driver-in-the-loop scenarios the effectiveness of the control has been demonstrated, i.e., the limitation of the lateral error in each scenarios is guaranteed.

The future challenge of the research is to develop the integration of steering control and cruise control within the proposed framework. It can provide to utilize the relationships between longitudinal and lateral dynamics, with which the safety level of the automated vehicle can be improved.

REFERENCES

J. Bokor and G. Balas. Linear parameter varying systems: A geometric theory and applications. *16th IFAC World Congress, Prague*, pages 1–6, 2005.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4693–4700, 2018.

T.F. Coleman and Y. Li. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6:418–445, 1996.

H. Demut, M. Hagan, and M. Beale. *Neural network design*. PWS Publishing Co, 1997.

Deepak Gangadharan, Jin Hyun Kim, Oleg Sokolsky, BaekGyu Kim, Chung-Wei Lin, Shinichi Shiraishi, and Insup Lee. Platform-based plug and play of automotive safety features: Challenges and directions. In *2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 76–84, 2016.

Mani Kaustubh, Dehlia Willemsen, and Manuel Mazo. The modeling of transfer of steering between automated vehicle and human driver using hybrid control framework. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 808–814, 2016.

Parham M. Kebria, Abbas Khosravi, Syed Moshfeq Salaken, and Saeid Nahavandi. Deep imitation learning for autonomous vehicles based on convolutional neural networks. *IEEE/CAA Journal of Automatica Sinica*, 7(1):82–95, 2020.

J.C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.

Peng Li, Di Liu, and Simone Baldi. Plug-and-play adaptation in autopilot architectures for unmanned aerial vehicles. In *47th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–6, 2021.

Chung-Wei Lin, BaekGyu Kim, and Shinichi Shiraishi. Hardware virtualization and task allocation for plug-and-play automotive systems. *IEEE Design Test*, 38(5):65–73, 2021.

Lisa J. Molnar, Lindsay H. Ryan, Anuj K. Pradhan, David W. Eby, Renée M. St. Louis, and Jennifer S. Zakrajsek. Understanding trust and acceptance of automated vehicles: An exploratory simulator study of transfer of control between automated and manual driving. *Transportation Research Part F: Traffic Psychology and Behaviour*, 58:319–328, 2018.

Balázs Németh. Coordination of lateral vehicle control systems using learning-based strategies. *Energies*, 14(5), 2021.

Balázs Németh and Péter Gáspár. *Guaranteed Performances for Learning-Based Control Systems Using Robust Control Theory*, pages 109–142. Springer International Publishing, Cham, 2021.

Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos A Theodorou, and Byron Boots. Imitation learning for agile autonomous driving. *The Int. J. of Robotics Research*, 39(2-3):286–302, 2020.

Sameer Qazi, Farah Sabir, Bilal A. Khawaja, Syed Muhammad Atif, and Muhammad Mustaqim. Why is internet of autonomous vehicles not as plug and play as we think? lessons to be learnt from present internet and future directions. *IEEE Access*, 8:133015–133033, 2020.

O. Sename, P. Gáspár, and J. Bokor. *Robust Control and Linear Parameter Varying Approaches*. Springer Verlag, Berlin, 2013.

Jakob Stoustrup. Plug and play control: Control technology towards new challenges. In *2009 European Control Conference (ECC)*, pages 1668–1683, 2009.

F. Wu, X.H. Yang, A. Packard, and G. Becker. Induced L_2 norm controller for LPV systems with bounded parameter variation rates. *Int. J. of Robust and Nonlinear Control*, 6:983–988, 1996.

Shuxiang Xu and Ling Chen. A novel approach for determining the optimal number of hidden layer neurons for fnn's and its application in data mining. *5th International Conference on Information Technology and Applications (ICITA 2008)*, pages 683–686, 2008.