

# Performance Guarantees on Machine-Learning-based Overtaking Strategies for Autonomous Vehicles

Balázs Németh, Tamás Hegedűs and Péter Gáspár

**Abstract**—The control of autonomous vehicles in overtaking scenarios is an important challenge, in which an autonomous vehicle in a multiple vehicle environment must be safely driven. Due to the complexity of vehicle scenarios, several machine-learning-based design strategies have been developed, which provide outstanding results. However, in most of these methods it is difficult to provide a theoretical guarantee on the most important performance of the overtaking strategy, i.e., the avoidance of collisions with the surrounding vehicles. This paper proposes a design architecture with which this performance can be guaranteed. The method is based on the robust control framework and it is independent from the structure of the machine-learning-based agent. The effectiveness of the method is illustrated through simulation examples.

## I. INTRODUCTION AND MOTIVATION

Machine-learning techniques are important components in the control of automated and autonomous vehicle systems. The advantage of the learning methods is that several decision processes, perceptions and the characteristics of human driving interventions can be effectively integrated in their agents. Machine-learning techniques have a special importance at the handling of the complex multi-vehicle scenarios.

Although there exist several optimal control solutions to the problem of overtaking maneuvers [1], [2], [3], [4], machine-learning-based methods have also been successfully applied. A reinforcement-learning-based overtaking control strategy is proposed in [5], [6]. In [7] a Q-learning strategy is used in the design of driving algorithms for multi-lane environments. An analysis method of the robust properties of the machine-learning-based overtaking decision strategies is found in [7]. A deep-reinforcement-learning method is used in [8], which is applied to decision making in different manoeuvres. The applied method provides the possibility of behaviour adaptation without re-training and thus, the agent is capable of adhering to traffic rules and learns to drive safely in a variety of situations.

In spite of the promising results on the application of machine learning methods in the overtaking control strate-

B. Németh, T. Hegedűs and P. Gáspár are with Systems and Control Laboratory, Institute for Computer Science and Control, Kende u. 13-17, H-1111 Budapest, Hungary. E-mail: [balazs.nemeth;tamas.hegedus;peter.gaspar]@sztaki.hu

This work has been supported by the GINOP-2.3.2-15-2016-00002 grant of the Ministry of National Economy of Hungary and by the European Commission through the H2020 project EPIC under grant No. 739592. The work of Balázs Németh was partially supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and the ÚNKP-19-4 New National Excellence Program of the Ministry for Innovation and Technology. The work of Tamás Hegedűs was partially supported by the ÚNKP-19-3 New National Excellence Program of the Ministry for Innovation and Technology.

gies, a crucial problem is the lack of performance guarantees, see [9]. A current issue is how it is possible to quantify and guarantee performance levels of a machine-learning-based agent in the sense of the control theory. The challenge is that the mathematical structure of the machine-learning-based algorithms and the formulation of the dynamic control systems are different. This complexity makes it difficult to examine the system which is yielded via their interconnections. Moreover, the conventional control systems are generally designed based on simplified control-oriented models, which are also used in the evaluation of closed-loop stability and performances [10]. It means that the machine-learning-based control systems can be complex for the evaluation of the closed-loop system through the conventional analysis methods. Although there are some novel results in the research on guarantees, e.g. [11], [12], the problem is still open.

This paper focuses on trajectory design and control design methods for machine-learning-based overtaking strategies with which the performance of collision avoidance can be guaranteed. The motivation of the paper is to provide a design framework, which is able to provide the advantages of a machine-learning agent (e.g. self-learning, achievement of human expectations through samples), while its drawback about the lack of performance guarantee is eliminated. As a novelty of the method, the generated trajectory of the machine-learning method is verified using a guaranteed safe trajectory, which is computed through an optimization task. Formally, it results in a measured disturbance, which is incorporated in the robust design of the local trajectory tracking control. The effectiveness of the method is illustrated through simulations using CarMaker. It is illustrated that the emergency, which results from an unexpected event for the machine learning algorithm, can be avoided.

The paper is organized as follows. The machine-learning-based trajectory design is presented in Section II. The design method of the guaranteed safety trajectory is formed in Section III. Section IV proposes the LPV-based control design, which guarantees the minimum performance level of the system. Simulation results are shown in Section V and the contributions are summarized in Section VI.

## II. MACHINE-LEARNING-BASED TRAJECTORY DESIGN

In this section a machine-learning-based trajectory design algorithm for the overtaking problem is presented. The design of the agent has three stages.

1./ The training set of the supervised learning method is generated. It requires an off-line method, in which the

route and the acceleration outputs for various scenarios are computed. The method is based on the results of [13], [14]. In this solution a graph-based algorithm is applied, in which several factors can be incorporated, e.g. a probability map of the surrounding vehicle positions, the comfort in the route selection, the decision of the overtaking, etc. Although the graph-based optimization provides an acceptable route for the vehicle, the numerical complexity makes it difficult in an online implementation. Thus, optimization can be used only as an off-line process. The motivation of the learning is to ignore the necessity of the online solution of the graph-based optimization problem. Therefore, in the proposed solution the optimization is performed for numerous scenarios and their results are used as a training set.

2./The generated training set must be processed by a supervised learning method, which results in a neural network. The training is also an off-line process, but the resulting neural network can be used in an on-line process.

3./ During the cruising of an autonomous vehicle several multi-vehicle scenarios must be handled. The scenarios can be used to enlarge the training set. This requires the logging of the input data and an off-line solution of the graph-based optimization. If the training set is enlarged, the initial neural network based on the enlarged set can be retrained. Using this self-learning process the trajectory design method for the autonomous vehicle can be improved, which is an advantage of the method. Unfortunately, the self-learning capability has the drawback that the future structure and the numerical values of the neural network may be unknown.

In the rest of this section the training process of the neural network is briefly presented.

#### *Training of the machine learning agent*

The neural network is a member of the machine learning family. The artificial neural network is modeled after the human brain in such a manner that it is able to solve complex, nonlinear problems. The main advantage which distinguishes this technique from the conventional machine learning algorithms is its ability to deal with different kinds of optimization tasks, such as clustering, classification, prediction, fitting etc. A neural network contains weights and activation functions, which are called neurons. They are grouped into layers. A network has one input and one output layer and, at least, one hidden layer. The number of the hidden layers and the type of the activation function can be chosen freely, and they are the main parameters of the neural network [15].

In this paper the neural network is trained using a supervised learning technique, which requires a data set for training and testing purposes. The training set is a subset of the collected data set, which is used for training the network. Another subset is the test set, which is used for evaluating the neural network. The data set is generated through 5000 distinct scenarios, in which the proposed graph search algorithm has been performed. During the data collection the initial parameters of the scenarios are selected randomly, e.g., the positions or the velocity values of the vehicles. The results of

the graph-based decision-making algorithm are the computed trajectory and the velocity profile. For the reduction of the computation in the training, the resulting lateral position values are transformed into discrete values  $0; \pm 1$ . The value means that the lateral position is not modified concerning the previous computation.  $\pm 1$  represents that the trajectory is modified to one step right or left on the grid. In the computation of the trajectory the road horizon is divided into 10 steps in longitudinal, and 7 in lateral directions, which results in a grid with 70 points.

The structure of the network consists of one input, one output and 3 hidden layers. The hidden layers contain 8 – 10 – 12 neurons. The numbers of the hidden layers and the neurons are selected by using the so-called k-fold cross validation technique [16]. The input vector of the network on the input layer contains

- the velocity difference between the actual velocity of the vehicle and the current velocity regulations,
- the lateral position of the vehicle,
- the distances and the velocity differences between the controlled vehicle and the further vehicles in the overtaking scenario.

The results of the neural network are the trajectory in grid step representation and the velocity profile  $v_{x,i+1,l}$ . The trajectory is converted into position values using a further smoothing layer, which results in  $y_{i+1,l}$ .

A further crucial part of the network is the used activation functions. Although there exist numerous functions that can be used in the training process, the rectified linear unit (ReLU) and the log-sigmoid functions are used in this estimation problem, because they can be easily adjusted to nonlinear problems. For training the network, the Levenberg-Marquardt algorithm is used [17].

### III. FORMULATION OF SAFE REFERENCE TRAJECTORY

The generation of a safe trajectory for overtaking is designed through a predictive optimization strategy [18], whose minimum performance level is quantified. The following description provides a brief overview of the method. In the trajectory design problem the lateral motion of the vehicle is formulated based on the kinematic model of the vehicle and the trajectory of the vehicle is formed as clothoid segments [19].

The curvature of the clothoid is a continuous function, which means that the relation between the curve in section  $i$  and  $i + 1$  is  $\kappa_{i+1} = \kappa_i + c_i L_i$ , where  $L_i$  is the distance between two section points and  $c_i$  is the ratio of the clothoid section. Moreover, during the cruising of the vehicle the velocity  $v_{x,i}$  may vary with constant acceleration  $a$ , which leads to the relation  $v_{x,i+1} = v_{x,i} + aT$ , where  $T$  is sampling time. The kinematic motion equations using the representation of the curvature and the velocity profile is transformed into a state-space representation

$$x_{i+1} = A(a)x_i + B(a)c_i, \quad (1)$$

where  $x_i = [y_i \ \psi_i \ \kappa_i]^T$  is the state vector of the system and  $c_i$  is its input and  $A(a), B(a)$  are acceleration-dependent matrices.

The goal of the design is to provide a trajectory which guarantees the avoidance of the objects on the road with minimum lateral displacement. The objective of the optimization is to minimize  $|y_i|$  for all  $i = 1 \dots n$ , which represents the prediction horizon. The criterion is formed in an objective function

$$J(C) = \frac{1}{2} Y^T(C) Q Y(C) + C^T R C, \quad (2)$$

where  $Y$  contains the predicted lateral errors,  $C$  involves the clothoid ratios,  $Q$  and  $R$  are weighting matrices.

During the optimization the actual positions of the objects and the borders of the lanes must be considered. The positions are incorporated in constraint relations on  $y_{i+j}, \forall j \in \{1, n-1\}$ . The role of the constraints is to limit the minimum and the maximum lateral values of the designed trajectory, such as  $y_{i+j} \geq y_{i+j}^{min}, y_{i+j} \leq y_{i+j}^{max}, \forall j \in \{1, n-1\}$ . where

$$y_{i+j}^{min} = y_{i+j}^{min,o} + d, \quad (3a)$$

$$y_{i+j}^{max} = y_{i+j}^{max,o} - d. \quad (3b)$$

In the computation of  $y_{i+j}^{min}, y_{i+j}^{max}$  two components are involved.  $y_{i+j}^{min,o}$  and  $y_{i+j}^{max,o}$  are determined by the positions of the objects and the borders of the lane.  $d$  is safe distance, which has an important role in the performance specification of the trajectory design algorithm.  $d$  represents the smallest distance between the safe trajectory and the objects or the borders of the lane. It has impact on the guaranteed minimum performance level of the vehicle control system.

The trajectory design leads to a constrained optimization problem, in which the objective function (2) must be minimized subject to the constraints on  $y_{i+1}$ . During the optimization the constrained optimization is solved for various fixed  $a$  acceleration values and the minimum of  $J(C)$  is requested depending on  $C, a$ . The variables in  $C$  and the parameter  $a$  are also bounded by constraint. The optimization leads to a series of clothoid ratios on the horizon from which the safe trajectory  $y_{i+1,s}, v_{x,i+1,s}$  in the next step  $i+1$  is computed.

#### IV. ROBUST CONTROL DESIGN OF THE AUTONOMOUS OVERTAKING STRATEGY

In this section the tracking control design method is proposed. The goal of the control is to perform the overtaking maneuver without a collision. It contains two tasks:

- it is necessary to decide about the acceptability of the machine-learning-based reference signal,
- the reference signal must be tracked with a limited error, with which a predefined safe distance  $s$  from the objects is guaranteed.

In the following the solutions of these tasks are presented.

#### A. Examinations on the reference signals

The reference signals for the vehicle control are the designed trajectory  $y_{ref}$  and the desired longitudinal velocity  $v_{ref}$ . In the generation of these signals  $y_{i+1,l}, v_{x,i+1,l}$  and  $y_{i+1,s}, v_{x,i+1,s}$  are considered.

The calculation of  $y_{ref}$  and  $v_{ref}$  are based on the relationships

$$y_{ref} = y_{i+1,s} + \Delta_{l,1}^*, \text{ if } \Delta_{l,1}^* \in \Lambda_{l,1}, \quad (4a)$$

$$v_{ref} = v_{x,i+1,s} + \Delta_{l,2}^*, \text{ if } \Delta_{l,2}^* \in \Lambda_{l,2}, \quad (4b)$$

where  $\Delta_{l,1}^*, \Delta_{l,2}^*$  are scalar design parameters and  $\Lambda_{l,1}, \Lambda_{l,2}$  are domains.

The values of  $\Delta_{l,1}^*, \Delta_{l,2}^*$  are selected in such a way that if the conditions of (4) are guaranteed, then  $y_{ref} = y_{i+1,l}$  and  $v_{ref} = v_{x,i+1,l}$ , such as

$$\Delta_{l,1}^* = y_{i+1,l} - y_{i+1,s}, \quad (5a)$$

$$\Delta_{l,2}^* = v_{x,i+1,l} - v_{x,i+1,s}. \quad (5b)$$

But, if the conditions of each relation are not guaranteed in (4), then the design parameters are limited with the boundaries of their domains. It means that the general rule of the reference signal formulation is

$$y_{ref} = y_{i+1,s} + \Delta_{l,1}, \quad (6a)$$

$$v_{ref} = v_{x,i+1,s} + \Delta_{l,2}, \quad (6b)$$

where

$$\Delta_{l,1} = \min(\max(\Delta_{l,1}^*; \Delta_{l,1,max}); \Delta_{l,1,min}), \quad (7a)$$

$$\Delta_{l,2} = \min(\max(\Delta_{l,2}^*; \Delta_{l,2,max}); \Delta_{l,2,min}), \quad (7b)$$

where  $\Delta_{l,1,min}, \Delta_{l,1,max}, \Delta_{l,2,min}, \Delta_{l,2,max}$  are the boundaries of  $\Lambda_{l,1}, \Lambda_{l,2}$ .

The formulations (6)-(7) show that  $y_{ref}, v_{ref}$  must be inside of a limited neighbourhood of the safe trajectory  $y_{i+1,s}, v_{x,i+1,s}$ . In practice, it is suggested to select  $|\Delta_{l,i,max}| = |\Delta_{l,i,min}| = \Delta_{l,i,m}, i = \{1; 2\}$ , which leads to symmetric domains. If the domains are small, then the machine learning based trajectory is often overridden, with which the benefits of the learning can be degraded. But, if the domains are broad, then an incorrect reference signal  $y_{i+1,l}, v_{x,i+1,l}$  is not improved in time. Thus, it is recommended to select the initial value of  $\Lambda_{l,1}, \Lambda_{l,2}$  boundaries through test experiences on the trajectory generation algorithms, e.g. simulations or real experiments. Moreover, there are further conditions, which must be considered during the selection, as it is proposed below.

The selection of  $\Lambda_{l,1}$  has high importance on the guaranteed minimum performance level of the entire overtaking control system, because the distance between the safe reference trajectory  $y_{i+1,s}$  and the object or the lane border can decrease through  $\Delta_{l,i,m}$ . The minimum performance level of the system is influenced by  $d$  (see (3)),  $\Delta_{l,i,m}$  and  $e_y, e_v$  tracking errors of  $y_{ref}, v_{ref}$ .

The performance specification of the overtaking control is defined that the minimum distance between the controlled vehicle or the lane border is  $s_{safe}$ . In a worst case scenario

the following constraint must be guaranteed, see an illustration in Figure 1:

$$d - \Delta_{l,1,m} - e_y - e_{y,v} - H/2 \geq s_{safe}, \quad (8)$$

where the components are the following. The value  $d$  is selected in the design of the safe reference trajectory. The value  $\Delta_{l,1,m}$  is selected and built in the control design through the scaling of the reference signal. The value of  $e_y$  is built in the control design directly through the scaling of the control performances. Since  $v_x$ , influences the lateral motion of the vehicle,  $e_v$  can cause a lateral position error  $e_{y,v}$ . It is computed through kinematic relations as

$$\begin{aligned} y_{i+1} &= y_i + T v_{x,i} + \frac{v_{ref} - v_{x,i}}{2} T + \frac{e_v T}{2} = \\ &= y_{i+1,s} + e_{y,v}, \end{aligned} \quad (9)$$

where  $e_{y,v} = \frac{e_v T}{2}$ . The value  $e_v$  is built in the control design directly through the scaling of the control performances, which leads to  $e_{y,v}$ . Moreover, the value  $H$  represents the width of the vehicle.

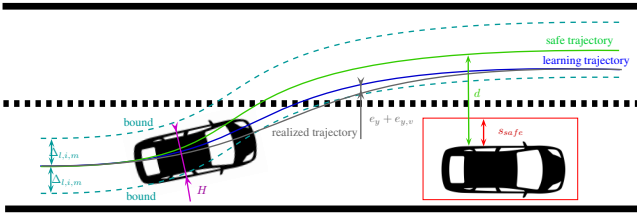


Fig. 1. Illustration of the overtaking trajectories

The inequality (8) shows that there are several parameters which must be selected during the design. In practice, it is recommended to select  $d$  and  $\Delta_{l,i,m}$  at high values, while  $e_y$  and  $e_v$  must be selected at a small value in the design.

### B. Design process of the robust controller

The design of the tracking control requires the model of the vehicle, which is described through the following dynamical model [20]:

$$m\ddot{y} = C_f \left( \delta - \frac{\dot{\psi} l_f + \dot{y}}{v_x} \right) + C_r \left( \frac{\dot{\psi} l_r - \dot{y}}{v_x} \right), \quad (10a)$$

$$J\ddot{\psi} = C_f l_f \left( \delta - \frac{\dot{\psi} l_f + \dot{y}}{v_x} \right) - C_r l_r \left( \frac{\dot{\psi} l_r - \dot{y}}{v_x} \right), \quad (10b)$$

$$m\dot{v}_x = F_{long} - F_{dist} \quad (10c)$$

where  $J$  is the yaw inertia of the vehicle,  $m$  is the vehicle mass,  $C_f, C_r$  are the cornering stiffness coefficients and  $l_f, l_r$  are geometric parameters. The signal  $\dot{y}$  is the lateral velocity and  $\dot{\psi}$  is the yaw rate. The longitudinal dynamics is described by  $v_x$  and the longitudinal traction force  $F_{long}$  and the disturbances  $F_{dist}$ , e.g. rolling resistance, air drag etc.

The designed controlled system must perform the following control performances

$$z_1 = y_{ref} - y, \quad |z_1| \rightarrow \min, \quad (11a)$$

$$z_2 = v_{ref} - v_x, \quad |z_2| \rightarrow \min, \quad (11b)$$

$$z_3 = \delta, \quad |z_3| \rightarrow \min, \quad (11c)$$

$$z_4 = F_{long}, \quad |z_4| \rightarrow \min. \quad (11d)$$

The performance vector is compressed as:  $z = [z_1 \ z_2 \ z_3 \ z_4]^T$ . The performances  $z_1, z_2$  can be reformed using (6) as

$$z_1 = y_{i+1,s} + \Delta_{l,1} - y, \quad (12a)$$

$$z_2 = v_{x,i+1,s} + \Delta_{l,2} - v_x, \quad (12b)$$

in which  $y_{i+1,s}, v_{x,i+1,s}$  and  $\Delta_{l,1}, \Delta_{l,2}$  are measured disturbances. There is an important difference between these signals

- $y_{i+1,s}, v_{x,i+1,s}$  are generated by the safe trajectory design method. Their characteristics (i.e. bounds, rates) are determined by the optimization algorithm.
- $\Delta_{l,1}, \Delta_{l,2}$  are the result of various signals, such as the machine learning algorithm, the safe trajectory design algorithm and the selected values  $\Delta_{l,1,m}, \Delta_{l,2,m}$ .

Since the sources of disturbances are different it is recommended to handle them separately.

The measured signals of the systems are the tracking errors

$$y_{m,1} = y_{ref} - y, \quad (13a)$$

$$y_{m,2} = v_{ref} - v_x, \quad (13b)$$

where  $y_m = [y_{m,1} \ y_{m,2}]^T$  vector contains the measured signals.

The dynamic equations (10), the performance (11)-(12) and the measurement equations (13) are transformed into a LPV state-space representation

$$\dot{x} = A(\rho)x + B_1 w + B_2 u, \quad (14a)$$

$$z = C_1 x + D_{11} w + D_{12} u, \quad (14b)$$

$$y_m = C_2 x + D_{21} w \quad (14c)$$

where  $A(\rho), B_1, B_2$  and  $C_1, C_2, D_{11}, D_{12}, D_{21}$  are matrices.  $x = [\dot{y} \ \dot{\psi} \ v_x]^T$  is the state vector, the disturbance is  $w = [F_{dist} \ y_{i+1,s} \ v_{x,i+1,s} \ \Delta_{l,1} \ \Delta_{l,2}]^T$  and the control input vector is  $u = [\delta \ F_{long}]^T$ .  $\rho = v_x$  is selected as a scheduling variable of the LPV system.

The control design is based on a weighting strategy, which is formulated through the closed-loop interconnection structure, see Figure 2. The interconnection structure contains several weighting functions, whose roles are to guarantee the trade-off between the performances and to scale the signals. The weights  $W_{n,1}, W_{n,2}$  are related to the sensor characteristics on the lateral and the velocity error measurements, where  $n_1, n_2$  represent noises.  $W_{dist}$  scales the longitudinal disturbance force.

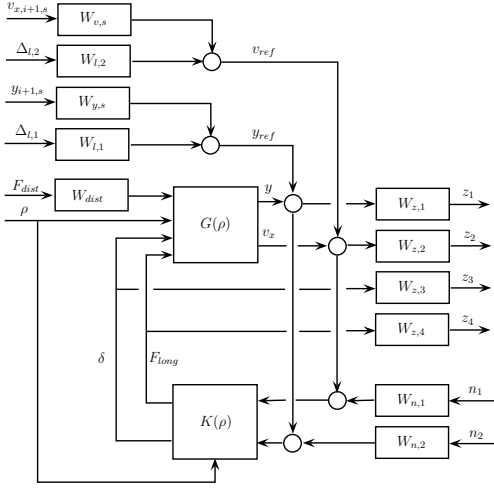


Fig. 2. Closed-loop interconnection structure

The role of weights  $W_{y,s}$ ,  $W_{v,s}$  is to consider the dynamics of the reference signals  $y_{i+1,s}$ ,  $v_{x,i+1,s}$ . Similarly,  $W_{l,1}$ ,  $W_{l,2}$  scale the disturbances  $\Delta_{l,1}$ ,  $\Delta_{l,2}$ . The weights are selected as

$$W_{l,1} = \frac{\Delta_{l,1,m}}{T_{12}s^2 + T_{11}s + 1}, \quad W_{l,2} = \frac{\Delta_{l,2,m}}{T_{22}s^2 + T_{21}s + 1}, \quad (15)$$

where  $T_{i,j}$  are design parameters, which represent the dynamics of the signal.  $\Delta_{l,i,m}$  are selected based on (8). The selected forms guarantee that the values of the disturbances are  $\Delta_{l,1,m}$ ,  $\Delta_{l,2,m}$  in steady state.

$W_{z,1}$ ,  $W_{z,2}$  and  $W_{z,3}$ ,  $W_{z,4}$  are the weights for the performances, which represent the minimization of them. Weights  $W_{z,1}$ ,  $W_{z,2}$  have important role from the aspect of the overall performance of the overtaking control, because they scale the tracking errors  $e_y$ ,  $e_v$  (8). The forms of the weights are

$$W_{z,1} = \frac{e_y}{T_{10}s + 1}, \quad W_{z,2} = \frac{e_v}{T_{20}s + 1}, \quad (16)$$

where  $T_{10}$ ,  $T_{20}$  are design parameters and  $e_y$ ,  $e_v$  are the expected tracking errors. The selected forms guarantee that the tracking errors are  $e_y$ ,  $e_v$  in steady state.

The design of the control is based on robust LPV method [21]. The problem is set up by gridding the parameter space and solving the set of LMIs that hold on the subset of  $\mathcal{F}_P$ . The induced  $\mathcal{L}_2$  norm of parameter-dependent stable LPV systems with zero initial conditions is defined as

$$\inf_K \sup_{\rho \in \mathcal{F}_P} \sup_{\|w\|_2 \neq 0, w \in \mathcal{L}_2} \frac{\|z\|_2}{\|w\|_2}. \quad (17)$$

The result of the optimization are the LPV controller  $K\rho$ , which which steering angle  $\delta$  and the longitudinal force  $F_{long}$  are computed.

Finally, Figure 3 shows the architecture of the entire control system, which is incorporated in the robust LPV controller and the generation blocks of the learning and safe reference signals. The role of the measured disturbance generator is to perform the computations (5)-(7).

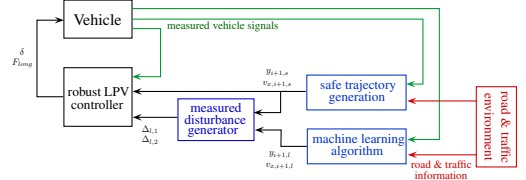


Fig. 3. Architecture of the entire control system

## V. SIMULATION RESULTS

The effectiveness of the proposed method is illustrated through simulation examples, in which the CarMaker vehicle dynamic software is used. In the scenario the machine-learning-based algorithm designs a trajectory of the vehicle, which is not safe enough and the vehicle may leave the road. This results in a scenario, which has been rare in the training set and thus, the machine-learning-based algorithm has a low performance. Nevertheless, the proposed robust control algorithm is able to guarantee the safety of the vehicle owing to the modification of the trajectory.

In the example the performance is guaranteed based on the inequality (8). The required safe distance is  $s_{safe} = 0.5m$ . The width of the vehicle is  $H = 2m$ . In the design of the LPV control  $e_y = 0.005m$  and  $e_v = 0.05m/s$  tracking errors are selected. Since  $T = 0.05s$  sampling time is selected, the lateral error from the velocity tracking error is  $e_{y,v} = e_v T/2 = 0.05 = 0.00125m$ . It means that the selection of  $d = 2.1m$  and  $\Delta_{l,1,m} = 0.5m$  guarantees the required performance, such as

$$2.1m - 0.5m - 0.005m - 0.00125m - 1m = 0.59375m \geq 0.5m.$$

The trajectory and velocity profile results are presented in Figure 4. They show that the motion of the vehicle is smooth and the maximum velocity regulation is kept through the tracking of the safe velocity profile generation.

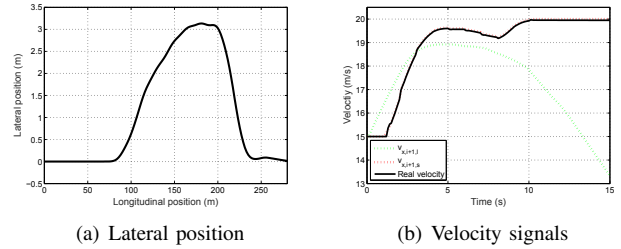


Fig. 4. Motion of the vehicle

Figure 5 compares results of the learning and the safe trajectories at three periods of the overtaking manoeuvre. The results show that  $y_{i+1,l}$  can lead to the departure of the road, which is avoided through the proposed method. Moreover, in the second segment the safe distance between the vehicles is preserved, see after 8.8 s. As a result, the reference trajectory is inside of the safe trajectory during the entire scenario. It guarantees the most important performance of the overtaking strategy.

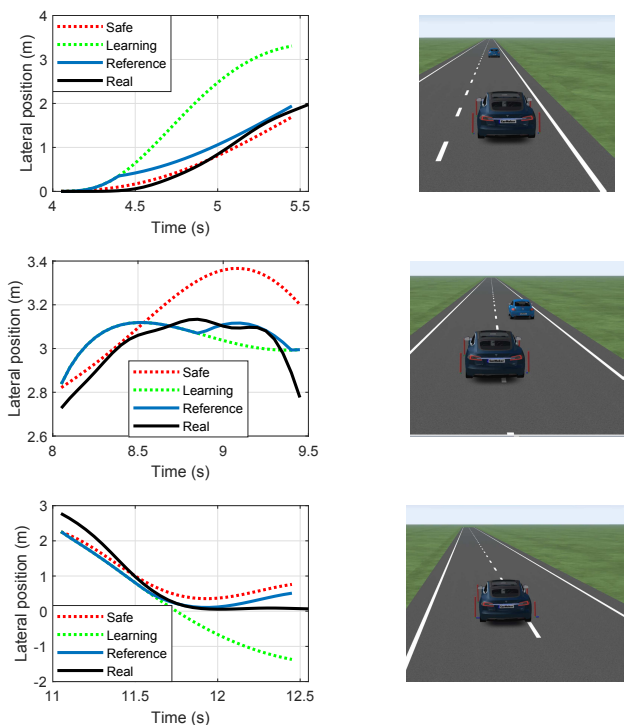


Fig. 5. Trajectories of the vehicle in various segments of the road

Finally, the control inputs of the system are illustrated in Figure 6, which are actuated on the vehicle. Through the steering angle (Figure 6(a)) and  $F_{long}$  (Figure 6(b)) the tracking of the reference trajectory (Figure 5) and the velocity (Figure 4(b)) are successfully performed.

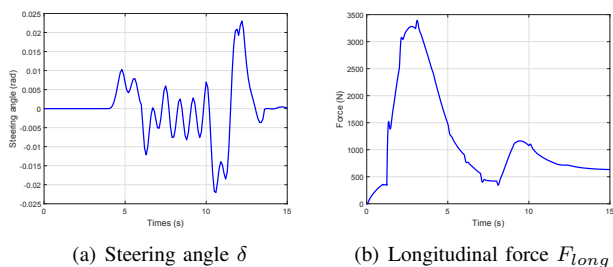


Fig. 6. Control interventions

## VI. CONCLUSIONS

The paper has proposed a design architecture for machine-learning-based trajectory design methods, with which the avoidance of collisions with the surrounding vehicles during the overtaking strategy can be guaranteed. During the simulation examples the effectiveness of the method is illustrated as follows. The proposed neural-network-based trajectory design method has been able to provide an appropriate trajectory. However, it has also been presented that in some special scenarios the generated trajectory may cause hazardous situations. Using the proposed robust design framework it has been possible to avoid these situations: the trajectory has been limited to vary in a neighbourhood of a safe trajectory.

The neighbourhood has been represented by the domain of a known disturbance, which has been incorporated in the robust control framework. The result of the paper has been a control design method which satisfies the safety requirements of overtaking manoeuvres.

## REFERENCES

- [1] K. Berntorp, "Path planning and integrated collision avoidance for autonomous vehicles," in *American Control Conference, Seattle*, 2017, pp. 4023–4028.
- [2] N. Murgovski and J. Sjöberg, "Predictive cruise control with autonomous overtaking," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 644–649.
- [3] N. A. Nguyen, D. Moser, P. Schragl, L. del Re, and S. Jones, "Autonomous overtaking using stochastic model predictive control," in *2017 11th Asian Control Conf. (ASCC)*, Dec 2017, pp. 1005–1010.
- [4] P. Petrov and F. Nashashibi, "Modeling and nonlinear adaptive control for autonomous vehicle overtaking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1643–1656, 2014.
- [5] D. C. K. Ngai and N. H. C. Yung, "A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 509–522, 2011.
- [6] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robotics and Autonomous Systems*, vol. 114, pp. 1 – 18, 2019.
- [7] R. Tami, B. Soualmi, A. Doufene, J. Ibanez, and J. Dauwels, "Machine learning method to ensure robust decision-making of AVs," in *IEEE Intelligent Transportation Systems Conference, Auckland, New Zealand*, October 2019.
- [8] P. Wolf, K. Kurzer, T. Wingert, F. Kuhnt, and J. M. Zollner, "Adaptive behavior generation for autonomous driving using deep reinforcement learning with compact semantic states," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 993–1000.
- [9] S. Shafaei, S. Kugele, M. Osman, and A. Knoll, "Uncertainty in machine learning: A safety perspective on autonomous driving," in *Computer Safety, Reliability, and Security. SAFECOMP 2018. Lecture Notes in Computer Science*, vol. 11094. Springer, 2018.
- [10] K. Zhou, J. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice Hall, 1996.
- [11] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. a data-driven control framework," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1883–1896, July 2018.
- [12] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 543–548, July 2018.
- [13] B. Németh, T. Hegedűs, and P. Gáspár, "Model predictive control design for overtaking maneuvers for multi-vehicle scenarios," in *2019 18th European Control Conference (ECC)*, June 2019, pp. 744–749.
- [14] T. Hegedűs, B. Németh, and P. Gáspár, "Graph-based multi-vehicle overtaking strategy for autonomous vehicles," *IFAC-PapersOnLine*, vol. 52, no. 5, pp. 372 – 377, 2019, 9th IFAC Symposium on Advances in Automotive Control AAC 2019.
- [15] M. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.
- [16] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statist. Surv.*, vol. 4, pp. 40–79, 2010.
- [17] M. Hagan, H. Demuth, and M. Beale, *Neural Network Design*. Boston: PWS Publishing, 1996.
- [18] B. Németh, T. Hegedűs, and P. Gáspár, "Optimal control of overtaking maneuver for intelligent vehicles," *Journal of Advanced Transportation*, 2018.
- [19] P. F. Lima, M. Trincavelli, J. Martensson, and B. Wahlberg, "Clothoid-based model predictive control for autonomous driving," in *European Control Conference*, 2015, pp. 2983–2990.
- [20] P. Gaspar, Z. Szabo, J. Bokor, and B. Nemeth, *Robust Control Design for Active Driver Assistance Systems. A Linear-Parameter-Varying Approach*. Springer Verlag, 2017.
- [21] 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.