Decision and control methods for overtaking strategies of autonomous vehicles

Theses of Ph.D. dissertation

by

TAMÁS HEGEDŰS

Supervisor

Dr. Balázs Németh

Budapest University of Technology and Economics
Faculty of Transportation Engineering and Vehicle Engineering
Department of Control for Transportation and Vehicle System

Budapest, Hungary

SZTAKI

Eötvös Loránd Research Network
Institute for Computer Science and Control
Systems and Control Laboratory

Budapest, Hungary

2022.

1 Motivation of data-driven vehicle control

Nowadays, one of the main challenges for the automotive industry is the development of autonomous vehicles, which involves several problems, such as motion-planning, decision-making, trajectory planning and control design. The accurate motion prediction is one of the main aspects of the efficient operation of autonomous vehicles. In the literature, various methods can be found, by which the motion of the other traffic participants can be predicted during the decision-making process. Physics-based methods [1] and machine-learning-based, [2] are widely used for the accurate prediction. The decision-making algorithm uses the results of the prediction layer, which aims to guarantee the safe and efficient operation of the autonomous vehicle in various traffic scenarios. Rule, probabilistic or optimization-based methods are widely used methods [3, 4], although, nowadays reinforcement learning-based solutions are one of the most commonly used approaches [5]. Based on the results of the decision-making layer, a trajectory for the vehicle is designed, which must satisfy several specifications such as the continuity of the curvature of the trajectory [6]. One of the widely used methods is the curve fitting method [7], and also optimization-based and machine learning-based solutions can be found in the literature [8]. The last step is to fulfill accurate trajectory tracking along the dynamically feasible trajectory. In the previous decades, several approaches have been developed for the controlling problems of the dynamic systems, such as robust control ($\mathcal{H}_\infty$) [9], Linear-Quadratic Regulator (LQR), [10], model predictive approaches (MPC, NMPC) [11] and polytopic system-based algorithms, such as Linear Parameter-Varying framework (LPV) [12]. With the increase in the computation capacity of CPUs and GPUs, new fields have come into a view such as data-driven and machine-learning-based approaches. Several methods have been developed using the combination of the classical and machine learning-based approaches [13]. It can be concluded during the review of the methods for the decision-making and trajectory planning problems of autonomous vehicles, that several different methods have been investigated. Nevertheless, the design process of an algorithm, which is capable of decision-making even in changing traffic situations, still hides many challenges and not straightforward tasks. Moreover, the difficulty of the control design arises from the fact that the parameters of the vehicle can change in a wide range, and also many environmental effects must also be taken into account. The aim of this dissertation is to propose own decision-making and trajectory planning algorithms for autonomous vehicles, with which both the safety and comfort requirements can be guaranteed. Moreover, an improved lateral control design methods are introduced, with which the performances can be increased.

2 Forming lateral dynamics of the vehicle

In this dissertation the one-track bicycle model is used during the modeling phase of the lateral vehicle dynamics [14]. The basic idea behind this model is that the front and rear wheels are replaced by one wheel each placed on the longitudinal axis of symmetry of the vehicle. The model consists of two main equations: lateral acceleration and yaw motion.

$$\ddot{\psi} I_z = C_f \left(\delta - \beta - \frac{\dot{\psi} l_f}{v_x} \right) l_f - C_r \left(-\beta + \frac{\dot{\psi} l_r}{v_x} \right) l_r, \tag{1a}$$

$$(\ddot{y} + v_x \dot{\psi}) m = C_f \left(\delta - \beta - \frac{\dot{\psi} l_f}{v_x} \right) + C_r \left(-\beta + \frac{\dot{\psi} l_r}{v_x} \right), \tag{1b}$$

where $\dot{\psi}$ denotes the yaw-rate and I_z is the yaw-inertia of the vehicle. Moreover, C_i gives the cornering stiffnesses of the tires and β is the side-slip angle, which can be computed using the longitudinal (v_x) and lateral velocity (v_y) as $\beta = \frac{v_y}{v_x}$. The distance from the axes to the center of the gravity is given by l_f, l_r and the lateral position of the vehicle is given by y. Finally, δ gives the steering angle. Since the dynamical equations of (1) describe the motion of the vehicle in a local coordinate system, it must be transformed into a road related coordinate system:

$$\dot{x}_g = v_x \cos(\psi) - v_y \sin(\psi), \tag{2a}$$

$$\dot{y}_g = v_x \sin(\psi) + v_y \cos(\psi), \tag{2b}$$

where x_g, y_g are the positions in global coordinate system. Through the formulation of the vehicle model the following further assumptions are taken:
A./ The lateral tire force characteristics are linearized around zero lateral slip α_f, α_r values, hence $F_{y,f} = C_f \alpha_f, F_{y,r} = C_r \alpha_r$, where $\alpha_f = \delta - \frac{v_y + l_f \dot{\psi}}{v_x}$ and $\alpha_r = \frac{-v_y + l_r \dot{\psi}}{v_x}$ are the slips on the front and rear axis.
B./ The steering angle has small value, which leads to the approximations $\cos\delta \approx 1$ and $\sin\delta \approx \delta$.
C./ It is assumed that the orientation of the vehicle is close to the orientation of the road, which leads to: $\cos\psi \approx 1$, $\sin\psi \approx \psi$. Using these, (2) equations are simplified as

$$\dot{x}_g = v_x - v_y \psi, \quad \dot{y}_g = v_x \psi + v_y. \tag{3}$$

The control-oriented formulation of the lateral vehicle model is based on the dynamical relationships (1) with the previous simplifications, the state space representation of the system can be formed:

$$\dot{x}_{lat,c} = A_{lat,c} x_{lat,c} + B_{lat,c} \delta, \tag{4a}$$

$$y_{lat,c} = C_{lat,c} x_{lat,c}, \tag{4b}$$

where the state vector is $x_{lat,c}^T = \begin{bmatrix} v_y & \psi & \dot{\psi} \end{bmatrix}$.

3 Theses of the dissertation

3.1 Optimal trajectory planning in overtaking strategies for autonomous vehicles

Overtaking is one of the most risky maneuvers for drivers due to the high velocity of the participants and many unexpected events and uncertainties of the maneuver. The main challenge during the decision-making and trajectory planning process is to define a maneuver that fulfills the requirements for safe driving. However, in most traffic scenarios several possible, collision-free routes can be defined and the main question is how can the optimal one chosen at low computational capacity. Another difficulty is the motion prediction of the traffic participants, which is necessary for the increase of the performances. Moreover, the result of the decision-making process is strongly connected with the performance level of the trajectory planning process. This means, that the parameters of the feasible trajectory should be considered during the decision-making process, however, in most cases, the two layers are arranged hierarchically. This chapter consists of the following two main parts:

- Firstly, a decision-making algorithm is presented, with which trajectory optimization is performed.
- Secondly, a method is proposed, in which the decision-making layer is augmented, with some information of the possible feasible trajectories.

Optimal trajectory design The first step of the decision-making process is to predict the motion of the surrounding vehicles. This process can be performed by the previously recorded dataset. Density functions are determined using the dataset, and the goal is to detect the typical behavior pattern of the drivers in the given traffic scenario. Based on the density function, the longitudinal motion of the surrounding vehicles is predicted. Then, the optimal trajectory design is performed, in which the geometrical information of the vehicle and other limitations are considered. The lateral motion of the vehicle is formulated based on the kinematic model of the vehicle, such as

$$\frac{dy(t)}{dt} = v_x \sin \psi(t), \quad \frac{d\psi(t)}{dt} = \frac{v_x}{D} \tan \delta(t), \tag{5}$$

where y is the lateral displacement, v_x is the longitudinal velocity in the vehicle coordinate system, ψ is the yaw angle, δ is the front steering angle of the front wheels and D is the distance between the front and the rear axle. After transforming the motion equation to space domain by making $v_x = ds(t)/dt$, the motion equations are discretized. Using the results, feasible trajectories can be designed, which are built up using clothoid segments. The constrained trajectory optimization problem is formed as

$$\min_{u_i \dots u_{i+n-1}} \frac{1}{2}\mathcal{C}^T \phi \mathcal{C} + \beta^T \mathcal{C} + \mathcal{C}^T \gamma, \tag{6}$$

subject to the following constraints are also guaranteed:

$$\mathcal{A} - Y^{min} \geq -\mathcal{B}\mathcal{C}, \tag{7a}$$

$$Y^{max} - \mathcal{A} \geq \mathcal{B}\mathcal{C}, \tag{7b}$$

$$\mathcal{C} \in \mathbf{C}, \tag{7c}$$

where $\mathcal{B}$ is built by the state matrices and $\mathcal{C}$ is built by the ratios of clothoid sections, $\mathbf{C}$ contains the achievable clothoid ratios. The minimum Y^{min} and maximum Y^{max} values are determined by the edges of the lanes and the surrounding vehicles. The optimization problem can be solved using standard quadratic programming methods, e.g., [15]. The solution of (6) leads to a series of clothoid ratios on the prediction horizon L_n.

Combining decision-making layer with trajectory design Previously, the overtaking strategy is formed in a hierarchical structure. The goal of this part is to merge the decision-making and the trajectory planning layers, with which several performances can be incorporated during the decision-making process. This means, that during the evaluation of the given traffic situation, the possible, feasible trajectories are considered and the optimal trajectory, which guarantees the safety and comfort requirements, can be chosen. As an initial step, the possible trajectories are arranged into a matrix, in which the start point of the overtaking maneuver is given by x_s, the end point is defined by x_f and y_i gives the lateral displacement. The matrix can be formed as:

$$\chi = \begin{bmatrix} x_{s,1}, x_{f,1}, y_0 & x_{s,1}, x_{f,2}, y_0 & \cdots & x_{s,1}, x_{f,n}, y_0 \\ x_{s,1}, x_{f,1}, y_1 & x_{s,2}, x_{f,2}, y_0 & \cdots & x_{s,2}, x_{f,n}, y_0 \\ \vdots & \vdots & \ddots & \vdots \\ x_{s,1}, x_{f,1}, y_{m-1} & x_{s,3}, x_{f,2}, y_{m-2} & \cdots & x_{s,m}, x_{f,n}, y_0 \end{bmatrix}, \tag{8}$$

where y_0 gives the center of the lane. In this case, when the vehicle does not necessarily need to start an overtaking maneuver, y_{m-1} is chosen. Using these information, clothoid segment-based trajectory design is performed for each element of the matrix. The feasible trajectories are compared to each other using potential fields, which are related to each performances. The selection of the trajectory is based on an optimization problem, whose cost function contains the weighted sum of the potential field functions. Using the potential field-based approach several effects are considered such as the center of the lane (P_{lane}), the longitudinal velocity (P_v), the acceleration (P_a) and the other participants ($P_{k,j}$). The cost function of the optimization is formed as

$$
\begin{aligned}
P(x_s, x_f, y, a_{long}) &= q_1 P_a(j(x_s, x_f, y)) + \\
&+ q_2 \sum_{k=1}^{N} P_{k,j}(x_s, x_f, y, a_{long}) + q_3 P_{lane}(y) + q_4 P_v(a_{long}),
\end{aligned} \tag{9}
$$

where q_1, q_2, q_3 and q_4 are design parameters to guarantee the priorities between the performances. The optimization problem based on (9) is formed as:

$$
\min_{x_s, x_f, y, a_{long}} P(x_s, x_f, y, a_{long}), \quad \text{subject to} \quad (x_s, x_f, y) \in \chi, \tag{10}
$$

The results of the optimization is a triad of the start and end point of the trajectory and also the lateral deviation.

Thesis 1 *I have developed two decision-making algorithms for overtaking strategy of autonomous vehicles. In the first method the motions of the surrounding vehicles using probability density functions are predicted. The proposed overtaking strategy includes decision-making and trajectory generation methods. The trajectory generation method is formed as a constrained optimization task, considering the results of the decision-making method. In the second method, the decision-making and trajectory generation methods in one layer are merged. As a result of the merging increased a number of performance criteria during the method can be guaranteed, e.g., the level of traveling comfort can be increased. The decision-making methods are implemented in high-fidelity vehicle dynamics simulation software in order to show the effectiveness of the proposed method.*

Related publications: [NGH18, NHG19, HNG19, HNG21a, HNG21b] For more details see Chapter 2. of the dissertation.

3.2 Multi-objective trajectory planning with performance guarantees

During the decision-making process of autonomous vehicles, several effects must be taken into account such as the width of the given road segment and the surrounding traffic participants. Many difficulties have to be overcome during the creation of these decision-making methods, such as the adaptation to different situations or efficient operation. Moreover, the motion prediction of the surrounding vehicles can increase the effectiveness of the algorithms. Due to the possible complex traffic scenarios, several cases may result in a complex optimization problem, which can be time-consuming. In order to reduce the computing capacity of the algorithm, machine-learning-based solutions can be used. However, the drawback of these solutions is that the safety requirements are hard to ensure since no theoretical guarantees can be given for several machine-learning-based solutions. In order to solve this problem, the results of the neural network, and a robust control framework can be combined, with which performances can be guaranteed. This chapter consists of the following two main parts:

- Firstly, an advanced graph-based route selection algorithm is presented for the decision-making process for autonomous vehicles.
- Secondly, using the results of the graph-based solution, a neural network-based solution is proposed. Moreover, an LPV-based control design is presented, which guarantees the minimum performance level of the system.

Graph-based decision-making algorithm The first step of the graph-based route selection algorithm is to predict the motion of the surrounding vehicles. Then, the whole prediction horizon is divided equidistantly both in longitudinal and lateral direction. Using this, a weighted, directed graph $G = (V; \bar{E})$ is built on the predicted road section, whose vertices (V) represent the possible route points and velocity profile of the vehicle. The probability of collision (P_c) is computed to discrete segments as for a given time interval $t \in [t_i, t_j]$:

$$P_c(t_i, t_j) = \sum_{l=1}^{N} \frac{P(\lambda, s_i(t_i), s_j(t_j), y_{min}, y_{max})}{N}, \tag{11}$$

where N is the number of human-driven vehicles in the region of interest and P gives the probability function. The weight of the edge between vertices $V_i, V_j \in E(V_i, V_j)$, $j > i$ is formed as follows

$$S(i,j) = P_c(t_i, t_j) + S_c(i,j) + S_v(i,j), \tag{12}$$

where S_c is a weight that represents the difference from the center of the lane, while S_v weights the difference from the reference velocity. The goal during the route selection is to find the route which guarantees the minimum sum of weights on the graph, which can be solved using a greedy algorithm. The algorithm yields the shortest path with the minimum distance between the initial and the target vertices, where the distance D is defined as

$$D = \sum_{d=1}^{M-1} S(d, d+1), \tag{13}$$

where M is the number of vertices in the route. The result of the graph search algorithm is the trajectory of the vehicle, regarding the route and the velocity on the predicted horizon.

Guaranteeing performances for overtaking scenarios Using the results of the graph-based decision-making algorithm, a machine-learning-based solution is performed, which provides the learning-based trajectory for the vehicle. The safe trajectory $(y_{i,s}, v_{x,i,s})$ comes from a predictive optimization strategy. In this approach, the final goal is to combine the results of the algorithms, and the maximum deviation between the two trajectories is given. Using these, the reference values can be computed as:

$$y_{ref} = y_{i+1,s} + \Delta_{l,1}^*, \text{ if } \Delta_{l,1}^* \in \Lambda_{l,1}, \tag{14a}$$

$$v_{ref} = v_{x,i+1,s} + \Delta_{l,2}^*, \text{ if } \Delta_{l,2}^* \in \Lambda_{l,2}, \tag{14b}$$

where $\Delta_{l,1}^*, \Delta_{l,2}^*$ are scalar design parameters and $\Lambda_{l,1}, \Lambda_{l,2}$ are domains. The goal of the LPV control is to perform the overtaking maneuver without a collision. This means that it is necessary to decide about the acceptability of the machine-learning-based reference signal and the reference signal must be tracked with a limited error, with which a predefined safe distance s from the objects is guaranteed. The robust controller is able to handle the deviation between the learning-based solution and the safe trajectory as a disturbance of the system. The state-space representation of the system can be written as:

$$\dot{x} = A(\rho)x + B_1 w + B_2 u, \tag{15a}$$

$$z = C_1 x + D_{11} w + D_{12} u, \tag{15b}$$

$$y_m = C_2 x + D_{21} w \tag{15c}$$

where $A(\rho), B_1, B_2$ and $C_1, C_2, D_{11}, D_{12}, D_{21}$ are matrices, 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.

Thesis 2 *I have developed a graph-based decision-making algorithm, by which the several effects (e.g., position of the center-line, different sizes of the traffic participants) can be taken into account effectively during the evaluation of the given traffic scenario. In the first step the method predicts the motions of the surrounding vehicles, and second, it generates a graph on the given prediction horizon, for the evaluation of the scenario. Finally, a machine-learning-based decision-making method is developed to reduce the complexity of the graph-based solution. In this method, a design architecture is proposed, by which the minimum performance level of the machine-learning-based solution can be*

guaranteed through the combination of robust control design and a learning-based agent. The effectiveness of the method in high-fidelity vehicle dynamics simulation software is illustrated.
Related publications: [HNG20b, HNG20c, NHG20, HNG20a] For more details see Chapter 3. of the dissertation.

3.3 Modeling and synthesis methods via closed-loop matching

Most of the modern control approaches are based on a physical model, which describes the motion, and dynamical behavior of the considered system. In general, these models are only an approximation of the system with some severe limitations, especially in the case of highly nonlinear systems. The models have a high impact on the performances of the control system. It means that to reach a high-performance level, an accurate model is required, which is a challenging task to acquire. In order to simplify the modeling method, a neural-network-based model matching algorithm is proposed. Moreover, using the resulted system, the linear control design tools can be used effectively since the nonlinearities are handled during the modeling process. This chapter consists of the following two main parts:

- Firstly, a neural network-based model matching algorithm is proposed.
- Secondly, the resulted neural network is evaluated and the reliability of the closed-loop system is computed.
- Finally, the closed-loop is analyzed and the uncertainties are determined, by which a robust control algorithm can be designed.

Neural network-based model-matching The goal is to compute an additional input signal Δu, by which the output signals (y_{nom}, y_{mat}) become identical for the given input signal u. The structure of the closed-loop matching is illustrated in Figure 1.

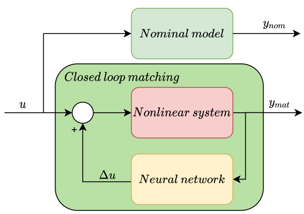


Figure 1: Structure of the closed- loop matching

The determination of the additional control input is not a straightforward task since the nonlinear system is not accurately known. In order to solve this issue, an iterative algorithm is proposed, in which the prior knowledge of the system is used. The iterative method is not suitable for the online application, therefore, the additional control input sequence is used as a training set of the neural network.

Reliability computation Using the neural-network-based closed-loop matching algorithm the performances of the control algorithm can be increased. Nevertheless, it is necessary to neglect the inappropriate neural network results, which may destabilize the closed-loop. This comes from two main reasons: unstable regions of the nonlinear system and high fitting errors. Using the resulted neural network, the given closed-loop system can be evaluated in a manner of stability and performances. And the whole dataset is sorted to two sets such as '*acceptable*' and '*not acceptable*' sets. During the reliability analysis for the actual state of the nonlinear system, the probability of the acceptable results is determined using the Bayesian rule [16] based on the sorted dataset:

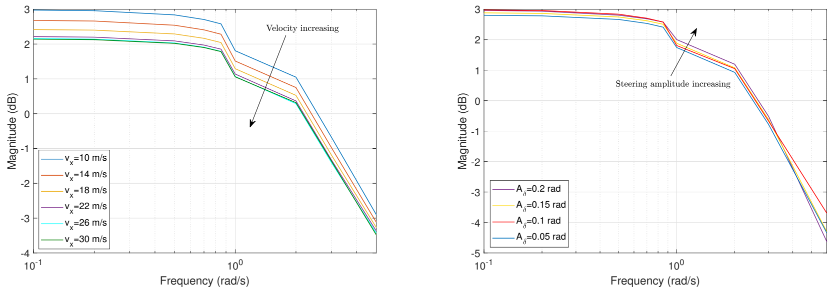
$$P(\mathcal{R}|\mathcal{X}) = P(\mathcal{R})\frac{P(\mathcal{X}|\mathcal{R})}{P(\mathcal{X})}, \tag{16}$$

where $P(\mathcal{R}|\mathcal{X})$ gives the probability of the acceptable results, if the states are $\mathcal{X} \in [x_{act} - x_l, x_{act} + x_u]$. The goodness of the neural network can be given by $P(\mathcal{R})$ term, which gives the rate of the acceptable data in the saved dataset. Finally, $P(\mathcal{X})$ specifies a probability value that expresses how often the system is within the given operating range. The computation of the probabilities in a real-time application cannot be achieved due to the high amount of data. In order to meet this criterion, based on the results of (16), a piecewise linear function is determined:

$$f_P(x) = \begin{cases} \frac{P(\mathcal{X}_i|\mathcal{R}_i)}{P(\mathcal{X}_i)}, & x \in \mathcal{X}_i \\ 0, & otherwise, \end{cases} \tag{17}$$

where, $\mathcal{X}_i$ gives the i^{th} subset of the dataset.

Modeling the uncertainty of the system In this example, the whole model matching process is carried out in terms of the yaw-rate of the vehicle. In order to quantify $\dot{\psi}_n - \dot{\psi}$, several simulations have been performed using the nonlinear vehicle model, which is implemented in CarMaker. During the simulations, the linearized system is excited with different steering angles and the vehicle runs at different longitudinal velocities. Using the results of the simulations, each points of the transfer functions from the measured yaw-rate ($\dot{\psi}$) to the yaw-rate error ($\Delta\dot{\psi} = \dot{\psi}_n - \dot{\psi}$) are calculated. The resulted transfer functions from the numerical computations are illustrated in Figure 2.



(a) At various velocities

(b) At various steering angles

Figure 2: Transfer functions of the neural network fitting error

As it can be seen, the amplitude of the yaw-rate error ($\Delta\dot{\psi}$) decreases along with the velocity and with the amplitude of the steering angle as well. The error of the neural network is handled as an uncertainty in the control design, which is described by the presented transfer functions.

Thesis 3 *I have developed a novel neural-network-based closed-loop matching method, whose goal is to eliminate the effects of nonlinearities and uncertainties. In the control structure, the neural network in an inner loop can be found. Using the measurable signals of the nonlinear system, the neural network computes an additional control input, with which the operation of the inner loop is matched to the operation of the nominal, linear model. Furthermore, a robust $\mathcal{H}_\infty$-based reference tracking control is designed, which handles the error of the neural network matching and guarantees accurate tracking performance. Finally, the effectiveness of the proposed method through a trajectory tracking problem of autonomous vehicles is illustrated.*

Related publications: [HFNG21, FHNG21b, HNG21a] For more details see Chapter 4. of the dissertation.

3.4 Control design using error-based ultra-local model

Besides the neural network-based closed-loop matching there can be found other approaches in the literature, which can improve the performances of a controller such as the Model Free Control (MFC). The basic idea behind this control structure is to compute an ultra-local model of the system, which is valid for a short period of time. By using this ultra-local model, the complex modeling process of the system can be neglected. Therefore, this method is suitable for systems with highly nonlinear dynamics and uncertainties. However, the implementation of this control algorithm is a challenging task due to the estimation errors and the time-delays. Another problem of the MFC algorithm is the tuning of the baseline controller, which still requires a model. The MFC structure has a free parameter, which is generally denoted by α. This parameter has a significant impact on the performances of the closed-loop system. An appropriately chosen improves the performances, however, it can also destabilize the controlled motion of the system. Since there is no elaborate method to select this parameter, the tuning of it is still an open question.

- Firstly, the modified MFC structure is proposed.
- Secondly, a tuning method is shown for the design parameter α.
- Finally, an extension of the state-space representation is presented, which serves the basis of a LPV control design.

Modified structure Using the original structure of the control algorithm, the ultra-local model is computed for the nominal system, and for the controlled system as well. Moreover, the goal is to reach the state that the error between the predefined reference signal and the measured output becomes zero. The computation of the modified error-based ultra-local model is the following:

$$y^{(\nu)} = F + \alpha u, \tag{18a}$$

$$y_{ref}^{(\nu)} = F_{nom} + \alpha u_{nom,ref}, \tag{18b}$$

$$\underbrace{y^{(\nu)} - y_{ref}^{(\nu)}}_{e^{(\nu)}} = \underbrace{F - F_{nom}}_{\Delta} + \underbrace{\alpha u - \alpha u_{nom,ref}}_{\alpha \tilde{u}}, \tag{18c}$$

where F is a parameter of the real system, while F_{nom} is a parameter of the nominal model. The error-based ultra-local model is computed using the deviation between the real system and the nominal model. Moreover, u denotes the input signal of the system. The measured output of the system is given by y and α is a design parameter. y_{ref} is the reference signal, $u_{nom,ref}$ is the calculated reference input signal, which is computed by using the nominal model. Moreover, ν denotes the ν^{th} derivative of the measured output y.

Tuning method for α Although the use of the nominal model makes the design process simpler, similar to the original structure tuning for the parameter α still remains challenging. If $\alpha \rightarrow \infty$ the effect of the ultra-local model-based part decreases. However, when $\alpha \rightarrow 0$, the controller input comes from mainly the ultra-local model. The goal is to develop a method, with which this parameter can be adjusted to the given driving situations. Practically, α should be determined using the states of the system, which should be directly measurable. The lateral acceleration of the vehicle is well correlated to the deviation of the nominal and the real system. Therefore, α is calculated using the following form:

$$\alpha_{act} = \alpha_0 - \phi a_y, \tag{19}$$

The selection of the parameters (α_0,ϕ), by which the smallest error can be reached, is not a straightforward task. In order to solve this issue, several simulations are performed, and using data-driven analysis the optimal values are chosen. The vehicle is driven along randomly generated overtaking trajectories. Using the saved dataset, the following optimization can be formed:

$$\arg\min_{\alpha_0,\phi} \left(\frac{\beta}{n} \sum_{j=1}^{n} (y_{ref,j,i} - y_{ms,j,i})^2 + \max(y_{ref,i} - y_{ms,i}) \right), \tag{20}$$

where i denotes the i^{th} test scenario. The reference trajectory is given by y_{ref} and the measured position is y_{ms}. During the determination of α the average error value and the maximum error value for the given test scenario are considered. The main role of β is to scale the two factors during the optimization process.

Extension of state-space The goal of the extension of the state-space representation is to include the effect of the error-based ultra-local model in the LPV control design. The components of the error-based ultra-local model are handled in the following way: $\ddot{y}_{ref} = \ddot{y}_{p,ref}$, and $u_{nom,ref} = \delta_{ref}$ are considered to be external, measurable disturbances of the controlled system. The inclusion of the signals $\ddot{y}_e$ and u are more challenging, because $y_{p,ref}$ is the predicted error and the second derivative of that signal is taken into account, which is solved using a filtered derivative term is applied. The transfer function can be transformed into a one dimensional state-space representation, whose matrices are: $A_{f,1} = [\frac{-1}{T_1}]$, $B_{f,1} = [\frac{1}{T_1}]$, $C_{f,2} = [1]$. The filtering term of the signal $\ddot{y}$ is of the form, which means: $A_{f,2} = [\frac{-1}{T_2}]$, $B_{f,2} = [\frac{1}{T_2}]$, $C_{f,2} = [1]$. Note that the robust controller uses only the input signal δ, the other intervention is handled by the error-based ultra-local model, by using the mentioned signals: $[\ddot{y}_p, \ddot{y}_{p,ref}, u, u_{nom,ref}]$.

$$\dot{x}_e = A_e(\rho)x_e + B_e(\rho)u_e + B_{e,w}(\rho)w_e, \tag{21a}$$

$$A_e(\rho) = \left[\begin{array}{c|c|c} A_v & B_v & -B_v/\alpha \\ \hline 0_{1\times 3} & A_{f,1} & -B_{f,1}/\alpha \\ \hline B_{f,2}A_v^{1\times 3} & 0_{1\times 1} & A_{f,2} \end{array}\right], \tag{21b}$$

$$B_e(\rho) = \left[\begin{array}{c} B_{v,1} \\ \hline B_{f,1} \\ \hline 0_{1\times 1} \end{array}\right], B_{e,w}(\rho) = \left[\begin{array}{c|c} B_v/\alpha & -B_v \\ \hline B_{f,1}/\alpha & -B_{f,1} \\ \hline 0_{1\times 1} & 0_{1\times 1} \end{array}\right], \tag{21c}$$

where $u_e = [\delta]$, $x_e^T = [\dot{\psi},\ \dot{y}_p,\ y_p,\ u,\ \ddot{y}_p]$, $w_e^T = [\ddot{y}_{p,ref}, u_{nom,ref}]$ and $A_v^{2\times 3} = e^T A_v$, $e^T = [0,1,0]$. $B_v = [\frac{l_1 C_1}{I_z},\ \frac{C_1}{m},\ 0]^T$, $\rho = [v_x, \alpha]$. The extended state-space representation serves as the basis of the robust control design, which is detailed in the following section.

Thesis 4 *I have developed a new ultra-local model-based formulation, which is extended with a nominal model, and thus, an error-based ultra-local model is formed. A tuning method has been proposed for setting design parameters, by which the performances of the control system is increased. An extended state-space representation of the error-based ultra-local model is formulated, which serves as the basis for Linear Parameter Varying (LPV) control design. The control algorithm has been validated on a test vehicle through various maneuvers.*

Related publications: [HFNG22, HFS+22c, FHS+22, HFS+22b]. For more details see Chapter 5. of the dissertation.

3.5 Guaranteed design of observers for autonomous vehicle control systems

In several cases, some of the states, which is required for the control algorithm, are cannot be measured directly or the sensors are not affordable. In order to solve this problem, generally, estimator and observer algorithms are used. However, the model-based observers can provide minimum performance level, the machine learning-based techniques provide possibility for the improvement of the performance level. The goal is to bridge the gap between the model-based and the learning-based observer design methods in order to exploit the advantages of both techniques. The advantage of the method is that it is independent of the internal structure of the learning-based observer, and thus, it can be used to provide guarantees for various learning-based solutions. This chapter can be divided into the following two main parts:

- In the first solution a design framework is proposed, in which the learning-based observer is merged with a $\mathcal{H}_\infty$-based robust observer.
- The second solution is based on an LPV-based observer design.

Robust $\mathcal{H}_\infty$-based observer design In the proposed methods the machine-learning-based solution is selected to a neural network, which is trained using the actual, measurable states of the system and the output is the hard-to-measure state. The goal of the observer is to minimize the difference between the states of the system and the estimated states, such as

$$\lim_{t\to\infty} (x(t) - \hat{x}(t))^2 \to min. \tag{22}$$

Thus, it is requested to find an observer matrix L which is able to minimize the objective (22). The structure of the observer, which contains L and the model of the systems is formed as

$$\dot{\hat{x}} = A(\hat{x} + \Delta) + B_2 u + L(y_m - C_2 \hat{x}), \tag{23}$$

where Δ vector is the improvement based on the learning-based estimation $\hat{x}_L$. The values in Δ is formed as follows. The values in $\Delta = [\Delta_1 \dots \Delta_i \dots \Delta_n]^T$ are bounded by predefined values $\Delta_{min,i}, \Delta_{max,i}$, such as $\Delta_i = \max\big(\min(\hat{x}_i - \hat{x}_{L,i}; \Delta_{max,i}); \Delta_{min,i}\big), \forall i \in n$, and index i for $\hat{x}_{L,i}, \hat{x}_i$ represents the elements of the state vector. It means that Δ can be interpreted as a state correction from $\hat{x}_L$, which is bounded to avoid the degradation of $\hat{x}$ is $\hat{x}_L$ is degraded. The main idea behind the observer design is that the vector of Δ and u can be handled as known disturbances.

LPV-based observer design In the previous method, the maximum deviation of the learning-based observer is integrated to the model-based observer design. In this solution an LPV-based solution is presented for the estimation problem, which is independent from the learning-based part. Moreover, a state update process is shown, by which it can be decided whether the output of the learning-based estimator is acceptable or not. The structure of a the whole algorithm is briefly shown in Figure 3. The state estimate at $t-n$

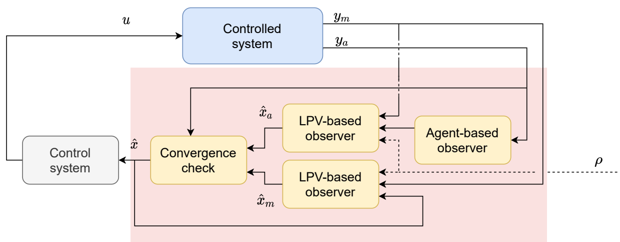


Figure 3: Structure of the observer

timestamp is denoted by the $\hat{x}_m(t-n)$, and $\hat{x}_a(t-n)$ is the updated state estimate using the outcome of the agent-based observer. Furthermore, y_a denotes the measured state variable and y_m is a measurement vector, which contains those measurable variables, which are required by the agent-based observer. Using the estimates $\hat{x}_a(t-n)$ and $\hat{x}_m(t-n)$ as initial condition the estimates $\hat{y}_a$ and $\hat{y}_m$ of the measurable output are computed for the time horizon n by the LPV-based observer. Whilst y_m is the measured output of the given system. In order to evaluate the accuracy of the agent-based observer at $t-n$, the convergence of $\hat{y}_a$ and $\hat{y}_m$ are checked in the following way:

$$e_a = y_m - \hat{y}_a, \quad e_m = y_m - \hat{y}_m, \tag{24}$$

where e_a and e_m are the error vectors of the measured output y_a and the estimated ones $\hat{y}_a$ and $\hat{y}_m$.

The state update is based on these error functions using a weighting function such as $W = [\omega_1, ..., \omega_n]$, which represents the importance of the errors during the time horizon. In this solution, the state-vector, using the output of the agent, is updated if it results in reduced estimation error on n compared to the estimation error on n without the agent. Formally, the state vector can be updated when the following condition is satisfied:

$$\begin{aligned} &\text{if} && |e_a^T|W < |e_m^T|W, \\ &\text{then} && \hat{x}(t) = \hat{x}_a(t). \end{aligned}$$

The choice of the weighting function W can be a crucial point of the algorithm, because it must guarantee the convergence of the estimation. In practice, it is recommended to use ascending weights ($\omega_1 \leq \omega_2 \leq ... \leq \omega_n$), which results in a more strict constraint on the convergence.

Thesis 5 *I have developed two novel observer design frameworks on state estimation problems of dynamical systems. In the first method, the neural-network-based observer is combined with a robust $\mathcal{H}_\infty$-based observer, which guarantees a minimum performance level on the estimation error. The combined observer design method is extended with a model-based controller, which leads to a joint robust $\mathcal{H}_\infty$ controller-observer design. Moreover, in the second solution, an observer design method is proposed, whose novelty is that the learning-based observer and an LPV-based observer in a joint observer design structure are incorporated. Through vehicle-oriented examples, the effectiveness of the proposed solutions is illustrated.*

Related publications: [NHG21, FHNG21a, HN20, HFS$^+$22a]. For more details see Chapter 6. of the dissertation.

4 Further challenges

Using the proposed methods and algorithms, several possible future research directions can be suggested:

- The decision-making algorithms are hard to validate in a real test vehicle since they cannot be driven autonomously in everyday traffic. Using the RC cars of SZTAKI, the goal is to validate and compare the performances of the methods presented in Thesis 1 and 2. The advantage of this system is that not only physical vehicles can be taken into account in a given traffic situation but also virtual vehicles, with which several traffic situations can be effectively investigated. Moreover, an interesting research topic could be the investigation how the methods affect human-driven vehicles.

- The graph-based decision making algorithms can be used for another vehicle-oriented control problems, such as the cooperative motion planning of ground and aerial vehicles. The Autonomous Systems National Laboratory, in SZTAKI has testing platform and environment, which can be a basis of this future research.

- Regarding the contributions of Thesis 3, the future challenge of the research is to take into account the impact of external disturbances and uncertainties (e.g., variation of the mass) of the closed-loop matching process, with which the robustness of the proposed method can be improved. Moreover, the training process of the neural network can also be improved by using another learning methods such as transfer learning.

- An important aspect can be the validation process of the lateral control algorithms using a real test vehicle. Although the ultra-local model-based controller has been tested on a test vehicle, the other control and observer methods have been validated using only the simulation software, CarMaker. During future research, the focus will be placed on the validation and implementation-related challenges.

Publications of the author

[FHN+20] Dániel Fényes, Tamás Hegedüs, Balázs Németh, Péter Gáspár, Damien Koenig, and Olivier Sename. LPV control for autonomous vehicles using a machine learning-based tire pressure estimation. *2020 28th Mediterranean Conference on Control and Automation (MED)*, pages 212–217, 2020.

[FHNG21a] Dániel Fényes, Tamás Hegedűs, Balázs Németh, and Péter Gáspár. Observer design with performance guarantees for vehicle control purposes via the integration of learning-based and LPV approaches. *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, pages 122–127, 2021.

[FHNG21b] Dániel Fényes, Tamás Hegedűs, Balázs Németh, and Péter Gáspár. Robust control design for autonomous vehicles using neural network-based model-matching approach. *Energies*, 14(21), 2021.

[FHS+22] Dániel Fényes, Tamás Hegedűs, Zoltán Szabó, Balázs Németh, and Péter Gáspár. Robust control design using ultra-local model-based approach for vehicle-oriented control problems. *European Control Conference (ECC)*, 2022.

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

[HFNG21] Tamás Hegedűs, Dániel Fényes, Balázs Németh, and Péter Gáspár. Improving sustainable safe transport via automated vehicle control with closed-loop matching. *Sustainability*, 13(20), 2021.

[HFNG22] Tamás Hegedűs, Dániel Fényes, Balázs Németh, and Péter Gáspár. Design of model free control with tuning method on ultra-local model for lateral vehicle control purposes. *American Control Conference, 2022*, 2022.

[HFS+22a] Tamás Hegedűs, Dániel Fényes, Zoltán Szabó, Balázs Németh, and Péter Gáspár. Combined observer design for road vehicles using LPV-based and learning-based methods. *11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes.*, 2022.

[HFS+22b] Tamás Hegedűs, Dániel Fényes, Zoltán Szabó, Balázs Németh, and Péter Gáspár. Coordinated control design for steering and torque-vectoring in model-free control structure. *11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes.*, 2022.

[HFS+22c] Tamás Hegedűs, Dániel Fényes, Zoltán Szabó, Balázs Németh, and Péter Gáspár. Robust LPV design within the model free control framework for road vehicles. *10th IFAC Symposium on Robust Control Design*, 2022.

[HN20] Tamás Hegedűs and Balázs Németh. LPV irányítás gépi tanuláson alapuló abroncsnyomás-becsléssel autonóm járművek számára. *XIV.*

Innováció és fenntartható felszíni közlekedés konferencia, IFFK 2020, pages 1–7, 2020.

[HNG19] Támas Hegedűs, Balázs Németh, and Péter Gáspár. Graph-based multi-vehicle overtaking strategy for autonomous vehicles. *IFAC-PapersOnLine*, 52(5):372–377, 2019. 9th IFAC Symposium on Advances in Automotive Control AAC 2019.

[HNG20a] Tamás Hegedűs, Balázs Németh, and Péter Gáspár. Challenges and possibilities of overtaking strategies for autonomous vehicles. *Periodica Polytechnica Transportation Engineering*, 48(4):320–326, 2020.

[HNG20b] Tamás Hegedűs, Balázs Németh, and Péter Gáspár. Design of a low-complexity graph-based motion-planning algorithm for autonomous vehicles. *Applied Sciences*, 10(21), 2020.

[HNG20c] Tamás Hegedűs, Balázs Németh, and Péter Gáspár. Multi-objective trajectory design for overtaking maneuvers of automated vehicles. *IFAC-PapersOnLine*, 53(2):15566–15571, 2020. 21st IFAC World Congress.

[HNG21a] Tamás Hegedűs, Balázs Németh, and Péter Gáspár. Control design framework for automated vehicles using an advanced feedback linearization. *Advances in Dynamics of Vehicles on Roads and Tracks. IAVSD 2021*, 2021.

[HNG21b] Tamás Hegedűs, Balázs Németh, and Péter Gáspár. MPC based semi-active suspension control for overtaking maneuvers. *Periodica Polytechnica Transportation Engineering*, 49(3):224–230, 2021.

[NGH18] Balázs Németh, Péter Gáspár, and Tamás Hegedűs. Optimal control of overtaking maneuver for intelligent vehicles. *Journal of Advanced Transportation*, 2018.

[NHG19] Balázs Németh, Tamás Hegedűs, and Péter Gáspár. Model predictive control design for overtaking maneuvers for multi-vehicle scenarios. *2019 18th European Control Conference (ECC)*, pages 744–749, 2019.

[NHG20] Balázs Németh, Tamás Hegedűs, and Péter Gáspár. Performance guarantees on machine-learning-based overtaking strategies for autonomous vehicles. *2020 European Control Conference (ECC)*, pages 136–141, 2020.

[NHG21] Balázs Németh, Tamás Hegedűs, and Péter Gáspár. Design framework for achieving guarantees with learning-based observers. *Energies*, 14(8), 2021.

References

[1] Andreas Eidehall and Lars Petersson. Statistical threat assessment for general road scenes using Monte Carlo sampling. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):137–147, 2008.

[2] Florent Altché and Arnaud de La Fortelle. An LSTM network for highway trajectory prediction. *IEEE ITSC 2017*, 2018.

[3] Liting Sun, Wei Zhan, Ching-Yao Chan, and Masayoshi Tomizuka. Behavior planning of autonomous cars with social perception. *2019 IEEE Intelligent Vehicles Symposium (IV2019)*, 2019.

[4] Pengwei Wang, Song Gao, Liang Li, Shuo Cheng, and Hai xia Zhao. Research on driving behavior decision making system of autonomous driving vehicle based on benefit evaluation model. *Archives of Transport*, 53:21–36, 2020.

[5] Árpad Fehér, Szilárd Aradi, and Tamás Bécsi. Q-learning based reinforcement learning approach for lane keeping. *2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)*, pages 31–36, 2018.

[6] Emese Szádeczky-Kardoss and Bálint Kiss. Continuous-curvature paths for mobile robots. *Periodica Polytechnica Electrical Engineering*, 53(1-2):63–72, 2009.

[7] Ernő Horváth and Claudiu Radu Pozna. Clothoid-based trajectory following approach for self-driving vehicles. *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 251–254, 2021.

[8] Ferenc Hegedüs, Tamás Bécsi, Szilárd Aradi, and Péter Gápár. Model based trajectory planning for highly automated road vehicles. *IFAC-PapersOnLine*, 50(1):6958–6964, 2017.

[9] József Bokor, Péter Gáspár, and Zoltán Béla Szabó. *Robust control theory with automotive applications*. Typotex, 2012.

[10] Fitri Yakub and Yasuchika Mori. Comparative study of autonomous path-following vehicle control via model predictive control and linear quadratic control. *Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering*, 229:1695–1714, 2015.

[11] Daniel Fenyes, Balazs Nemeth, and Peter Gaspar. Impact of big data on the design of MPC control for autonomous vehicles. *2019 18th European Control Conference (ECC)*, pages 4154–4159, 2019.

[12] Olivier Sename, Peter Gaspar, and Jozsef Bokor. *Robust Control and Linear Parameter Varying approaches: Application to Vehicle Dynamics*, volume 437 of *Lecture Notes in Control and Information Sciences (LNCIS, volume 437)*. Spinger, 2013.

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

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

[15] P. E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press, London UK, 1981.

[16] William M. Bolstad. *Introduction to Bayesian Statistics*. John Wiley and Sons, Ltd, 2007.