

# Lateral Control for Automated Vehicles Based on Model Predictive Control and Error-Based Ultra-Local Model

Tamas Hegedus<sup>1</sup> <sup>a</sup>, Daniel Fenyes, Vu Van Tan<sup>2</sup> and Peter Gaspar<sup>1,3</sup>

<sup>1</sup>*Institute for Computer Science and Control (SZTAKI), Eötvös Loránd Research Network (ELKH),  
Kende u. 13-17, H-1111 Budapest, Hungary*

<sup>2</sup>*Department of Automotive Mechanical Engineering, Faculty of Mechanical Engineering,  
University of Transport and Communications, 3 Cau Giay Street, 100000 Hanoi, Vietnam*

<sup>3</sup>*Department of Control for Transportation and Vehicle Systems, Budapest University of Technology and Economics,  
Stoczek u. 2, H-1111 Budapest, Hungary*

Keywords: Ultra-Local Model, Lateral Control, Automated Vehicles.

Abstract: The paper proposes a combined control design framework using Model Predictive Control (MPC) and ultra-local model-based methods. The main idea behind the control algorithm is to exploit the advantage of both approaches. During the control input computation, a simplified model is used, which has a significant impact on the computational cost. Moreover, the simplified model does not contain hardly measurable or varying vehicle-specific parameters, which makes the whole control design process easier. The ultra-local model is used to deal with the unmodeled dynamics of the vehicle, by which the performance of the control system can be increased. The effectiveness of the proposed control structure is demonstrated through trajectory tracking problem of autonomous vehicles. The whole algorithm is implemented in a high-fidelity vehicle dynamics simulation software, whose results are compared to an accurate model-based MPC in terms of computational cost and tracking accuracy.

## 1 INTRODUCTION

In the last decades, several model-based control design techniques have been developed and successfully implemented, such as PID, LQR,  $\mathcal{H}_\infty$  (Batista et al., 2019; Zhou and Doyle, 1998). However, for the effective application of these techniques, an accurate model is essential, especially when high-performance level must be guaranteed. One of the main difficulties of the modeling process is that the parameters of the system may change during its operation. Another issue connected to accurate modeling of the system can be challenging due to high nonlinearities and uncertainties. On the other hand, an inaccurate model of the system can lead to performance degradation, instability, or constraint violation, which is not suitable for safety-critical systems such as automated vehicles.

In addition to the mentioned control design methods, the Model Predictive Control (MPC) is a widely used technique due to its advantages such as constraining specific state(s) of the system (Schwenzer et al., 2021). The optimization-based control signal

calculation provides, for example, the possibility to limit the input signal and guarantee constraints for the given states of the system. Moreover, nonlinearities and changing parameters can also be efficiently taken into account during the control signal calculation (e.g., LTV-MPC, NMPC) (Allgöwer et al., 2004; Katriniok and Abel, 2011). However, this control design process still requires an accurate model or parameter estimation to provide high-performance level. Another difficulty of the optimization-based methods is that the complexity of the model and the length of the control horizon have a high impact on the computational capacity.

In summary, simplifying the modeling process or reducing the complexity of the model can make the control design process easier. In order to avoid the performances degradation caused by the simplified model, the ultra-local model-based approach can be applied, which is also called in some literature a Model-Free Control (MFC) structure (Fliess and Join, 2013). The ultra-local model-based approach has been successfully implemented for control design of many systems such as micro air vehicles

<sup>a</sup>  <https://orcid.org/0000-0001-8466-1102>

(Barth et al., 2020) or active suspension (Ghazally I. Y. Mustafa, 2019). However, this approach cannot guarantee steady-state zero error. Therefore, a nominal feedback controller is applied, which is hard to appropriately tune without a prior knowledge of the system. Moreover, in (Wang and Wang, 2020) a hybrid method is presented, with which both the advantages of the ultra-local model and the MPC-based solution can be exploited.

This approach can handle the parameter change of the vehicle. The main advantage of the combined structure is that it does not require an accurate model of the system, which makes the whole control design process easier. Furthermore, the parameters of a vehicle may change during its operation, which can be also effectively handled by the ultra-local model. Another aspect of this control structure, is that several constraints can be applied to the states of the vehicle, with which stable motion can be achieved. In addition, the computational time can be decreased using the low-complexity model. In the paper, a kinematic model-based solution is compared to another MPC algorithm, which uses a dynamic, accurate model of the vehicle. Finally, it is shown, that using the combined solution, the same performance level can be reached during the control of the vehicle. The simulations are performed in high fidelity vehicle dynamics simulation software, CarMaker.

The paper is structured as follows: The nominal models are demonstrated, which are used during the control of the vehicle in Section 2. Then, in Section 3 a description of the error-based ultra-local model can be found. The MPC formalism can be found in Section 4. The simulation example and the comparison of the controllers are located in Section 5. Finally, the whole paper and its contribution are summarized in Section 6.

## 2 LATERAL MODELS OF THE VEHICLE

In this section, two lateral models are presented for the lateral modeling purposes of the vehicle. Firstly, the two-wheeled bicycle model is described, while the second model is created based on the kinematic model of the system. The advantage of the first lateral model is that it takes into account the dynamic effects of the real vehicle. However, the second model has less complexity and does not need dynamic parameters of the controlled vehicle, which makes the modeling process easier.

## Dynamic Lateral Vehicle Model

In this subsection, the accurate lateral model is presented, which considers several effects of the dynamics. For modeling purposes the two-wheeled lateral vehicle model is used, which consists of the following two main equations (Rajamani, 2005):

$$I_z \ddot{\psi} = \alpha_f C_f l_f - \alpha_r C_r l_r, \quad (1a)$$

$$m(\ddot{y} + v_x \dot{\psi}) = \alpha_f C_f + \alpha_r C_r, \quad (1b)$$

where the side-slip of the front ( $\alpha_f$ ) and rear ( $\alpha_r$ ) tires can be computed, using the lateral velocity ( $v_y$ ) and the yaw-rate ( $\dot{\psi}$ ) of the vehicle, as:  $\alpha_f = \delta - \frac{v_y + \dot{\psi} l_f}{v_x}$  and  $\alpha_r = \frac{-v_y + \dot{\psi} l_r}{v_x}$ . Moreover,  $m$  gives the mass and the yaw inertia is given by  $I_z$ . The distance between the center of gravity and the axles is given by  $l_f$  and  $l_r$ . The control input is the steering angle of the vehicle ( $\delta$ ), while the longitudinal velocity is expressed with  $v_x$ . Based on the dynamical equations, the following state space representation can be formed:

$$\dot{x}_{veh} = A_{veh}(v_x)x_{veh} + B_{veh}(v_x)\delta, \quad (2a)$$

$$A_{veh} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & 1 & 0 & v_x \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad B_{veh} = \begin{bmatrix} b_1 \\ b_2 \\ 0 \\ 0 \end{bmatrix}, \quad (2b)$$

where  $a_{11} = -\frac{l_f^2 C_f + l_r^2 C_r}{I_z v_x}$ ,  $a_{12} = -\frac{l_f C_f - l_r C_r}{I_z v_x}$ ,  $a_{21} = -\frac{l_f C_1 + l_r C_2}{m v_x} - v_x$ ,  $a_{22} = -\frac{C_f + C_r}{m v_x}$ ,  $b_1 = \frac{l_f C_f}{I_z}$ , and  $b_2 = \frac{C_f}{m}$ . The state vector is the following:  $x_{veh} = [\dot{\psi}, v_y, y_g, \psi]^T$ , which means the states of the system are the yaw-rate, the lateral velocity of the vehicle in the local coordinate system ( $v_y$ ), the global lateral position ( $y_g = v_x \sin(\psi) + v_y \cos(\psi)$ ), which is approximated for small yaw angles  $\sin(\psi) \approx \psi$ ,  $\cos(\psi) \approx 1$ , and the yaw-angle. The presented model is augmented with the steering dynamics of the system, in order to increase the performances of the control. The steering system is modeled as a first-order term, which can be written to the following state space representation:  $A_{st} = [\frac{-1}{T_{st}}]$ ,  $B_{st} = [\frac{1}{T_{st}}]$ ,  $C_{st} = [1]$ . In this paper, the parameter of the steering system is set to  $T_{st} = 0.25$ , which is a reasonable value for the modeling process. The state space representation model of the system is augmented with the steering system, which leads to the following matrices:

$$A_{veh,st} = \begin{bmatrix} A_{st} & 0^{1 \times 4} \\ B_{veh} C_{st} & A_{veh} \end{bmatrix}, \quad B_{veh,st} = \begin{bmatrix} B_{st} \\ 0^{4 \times 1} \end{bmatrix}, \quad (3)$$

$$C_{veh,st} = \begin{bmatrix} 0^{4 \times 1} \\ C_{st} \end{bmatrix}.$$

The computation of the steering angle is performed by an MPC method, which requires discrete time model. Using (3) the discrete model computed as:  $A_{dyn} = e^{A_{veh,st}T_s}$  and  $B_{dyn} = \int_{kT_s}^{(k+1)T_s} e^{A_{veh,st}((k+1)T_s-\tau)} B_{veh,st} \tau$ , where  $T_s$  is the sampling time of the system, which is set to  $T_s = 0.05s$ .

## 2.1 Kinematic Vehicle Model

After the description of the dynamic vehicle model, the kinematic lateral model is detailed. The main difference between this, and the previously described model, is that many effects related to the vehicle dynamics are neglected during the modeling process such as the tires, and the steering system. Moreover, vehicle-specific parameters are not necessary for the modeling process and the changing parameters also do not need to consider. However, neglecting the dynamic effect may cause low performance or unstable motion. During the MPC design, this model is extended with the effect of the error-based ultra-local model to increase tracking performances. Two main equations can be formulated for the lateral and angular motion of the vehicle (Lima et al., 2015):

$$\frac{dy_g(t)}{dt} = v_x \sin \psi(t), \quad (4a)$$

$$\frac{d\psi(t)}{dt} = \frac{v_x}{l} \tan \delta(t), \quad (4b)$$

where the distance between the two axes is represented with  $l = l_f + l_r$ . The longitudinal velocity can be expressed as  $v_x(t) = ds(t)/dt$ , with which (4) can be transformed into space-domain as:

$$\frac{dy_g(s)}{ds} = v_x \sin \psi(s), \quad (5a)$$

$$\frac{d\psi(s)}{ds} = \frac{\delta(s)}{l}. \quad (5b)$$

The curvature of the trajectory ( $\kappa(s)$ ) can be calculated using (5b) as  $\kappa(s) = \delta(s)/l$ . One of the main specifications for lateral control of a vehicle is related to comfort requirement, which is fulfilled through a smooth trajectory design. To meet this criterion, the whole trajectory can be built up using clothoid segments, which leads to the following expression:  $\kappa(s) = \kappa_0 + cs$ , where  $c$  provides the sharpness of the given segment.

In the followings, the continuous equations are transformed into discrete form, with the assumption, that the vehicle travels with a constant velocity between two sampling time steps. Thus, the arc length between two points can be computed as  $L_k = v_x(k)T_s$ ,

and using the clothoid segments, the curvature can be formulated  $\kappa(k+1) = \kappa(k) + c(k)L_k$ . The lateral position and the yaw-angle of the vehicle can be also transferred into discrete time:

$$\psi(k+1) = \psi(k) + \kappa(k)L_k + \frac{1}{2}c(k)L_k^2, \quad (6)$$

$$y_g(k+1) = y_g(k) + L_k \sin(\psi(k)). \quad (7)$$

Using the kinematic equation for the vehicle motion description, and assuming small angles, the following discrete state-space representation can be formed:

$$\begin{aligned} x_{kin}(k+1) &= A_{kin}x_{kin}(k) + B_{kin}c(k) \\ \begin{bmatrix} y_g(k+1) \\ \psi(k+1) \\ \kappa(k+1) \end{bmatrix} &= \underbrace{\begin{bmatrix} 1 & L_k & 0 \\ 0 & 1 & L_k \\ 0 & 0 & 1 \end{bmatrix}}_{A_{kin}} \begin{bmatrix} y_g(k) \\ \psi(k) \\ \kappa(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ \frac{L_k^2}{2} \\ L_k \end{bmatrix}}_{B_{kin}} c(k), \end{aligned} \quad (8)$$

where the states ( $x_{kin}(k)$ ) are the lateral position, the yaw-angle, and the curvature of the trajectory. Moreover, the control signal is the sharpness of the clothoid segment. The output vector is computed as  $y_{kin}(k) = C_{kin}^T x_{kin}(k)$ ,  $C_{kin}^T = [1, 0, 0]$ .

During the computation of the ultra-local model-based part of the control algorithm an additional control input, which is a steering angle in this paper, is determined. However, in order to achieve the stable motion of the vehicle, the effect of the additional control input is taken into account during the determination process of the bounds. In this paper, the effect of the ultra-local-based part is considered through the approximation of the lateral acceleration ( $a_y$ ), which is limited during the optimization process. The lateral acceleration is also approximated by a kinematic assumption (Polack et al., 2017). The turning radius is given as:

$$R = \frac{l}{\tan(\delta)}. \quad (9)$$

The lateral acceleration is approximated as:  $a_y = \frac{v^2}{R}$ . Thus, the lateral acceleration at a given time step is given by  $a_y(k) = \kappa(k)v^2(k)$ . Moreover, the yaw-rate is also can be approximated from the turning radius and the actual longitudinal velocity  $\dot{\psi} = v_x/R$ . During the computation of the lateral acceleration, the modeling process of the steering system has been neglected.

### 3 ERROR-BASED ULTRA-LOCAL MODEL

This section describes the error-based ultra-local model, which approximates the deviation between the nominal model and the real system. Using this, the non-modeled dynamics of the system can be also taken into account and the performances of the control algorithm can be increased. For the computation of the error-based ultra-local model, the ultra-local model, and the nominal ultra-local model is needed. The differences between them are considered as an error-based ultra-local model (Hegedűs et al., 2023):

$$y^{(v)} = F + \alpha u, \quad (10a)$$

$$y_{ref}^{(v)} = F_{nom} + \alpha u_{nom}, \quad (10b)$$

$$\underbrace{y^{(v)} - y_r^{(v)}}_{e^{(v)}} = \underbrace{F - F_{nom}}_{\Delta} + \underbrace{\alpha u - \alpha u_{nom}}_{\alpha \tilde{u}}, \quad (10c)$$

$$e^{(v)} = \Delta + \alpha \tilde{u}, \quad (10d)$$

where  $F_{nom}$  is the nominal ultra-local model,  $u_{nom}$  is the nominal control input of the system. Similarly to the original structure, zero error can be achieved only for the derivative of the error signal. This means, that the augmented structure also requires a classical controller for accurate tracking performances:

$$u = -\frac{\Delta}{\alpha} - \mathcal{K}(x, y_{ref}). \quad (11)$$

The main concept of the error-based ultra-local model is introduced briefly in this paper, however, it is detailed in (Hegedűs et al., 2023). In this case, the classical control algorithm  $\mathcal{K}$  is chosen to an MPC, which uses the actual states of the system ( $x$ ) and also the reference signal. The advantage of the MPC-based extension is that this structure does not require additional signal computation such as the nominal control input ( $u_{nom}$ ). During the design process, the nominal model of the system is used, which is described in Subsection 2.1. The parameter  $\alpha$  aims to scale the derivative of the output to the control input. In this paper,  $\alpha$  is set to a constant value and it is selected to  $\alpha = 100$ . Moreover, for vehicle control-related problems, the value  $v$  can be selected to  $v = 2$ . More details regarding the choice of the parameters can be found in More details can be found in (Hegedűs et al., 2022). Figure 1 shows briefly the combined control structure of the control algorithm.

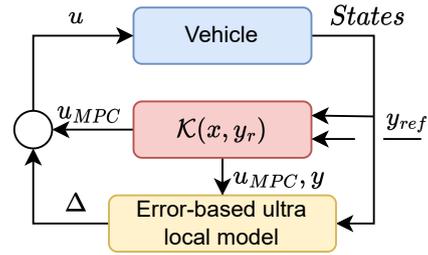


Figure 1: The structure of the control algorithm.

### 4 LATERAL CONTROL DESIGN

The goal of the paper is to compare the performances and computational capacity of the two different lateral vehicle models. The nominal control algorithm is selected for an MPC-based solution, with which several constraints can be taken into account. The stable motion of the vehicle is guaranteed through these constraints and also the effect of the additional control input can be taken into account. The goal during the input signal calculation is to achieve the following performances:

- The tracking error:  $y - y_{ref} \rightarrow \min!$
- The interventions:  $\delta \rightarrow \min!$

In the followings, a brief introduction is presented for the Model Predictive Control design.

#### 4.1 Motion Prediction and the Cost Function

In the first step, based on the system model and the actual states of the vehicle, the states of the system are predicted along the prediction horizon, which can be made computed in a general form as:

$$\begin{aligned} y(k+2) &= C^T (Ax(k+1) + Bu(k+1)) = \\ &= C^T (A(Ax(k) + Bu(k)) + Bu(k+1)), \end{aligned} \quad (12)$$

where  $A, B, C^T$  are the system matrices. In the second step, the error between the actual output ( $y$ ) and the reference value ( $y_{ref}$ ) can be predicted as  $\varepsilon(k+i) = y(k+i) - y_{ref}(k+i)$ . Using this, and the vector of the reference values ( $\mathcal{R}$ ), the error is computed for the whole prediction horizon as:

$$\begin{bmatrix} \varepsilon(k+1) \\ \varepsilon(k+2) \\ \vdots \\ \varepsilon(k+N_p) \end{bmatrix} = \begin{bmatrix} C^T A \\ C^T A^2 \\ \vdots \\ C^T A^n \end{bmatrix} x(k) - \begin{bmatrix} y_{ref}(k+1) \\ y_{ref}(k+2) \\ \vdots \\ y_{ref}(k+N_p) \end{bmatrix} + \begin{bmatrix} C^T B & 0 & \cdots & 0 \\ C^T AB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C^T A^{N_p-1} B & C^T A^{N_p-2} B & \cdots & C^T B \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_p-1) \end{bmatrix} = \mathcal{A}x(k) - \mathcal{R} + \mathcal{B}\mathcal{U} \quad (13)$$

The goal is to define a cost function, with which the optimization process is performed. Using (13) the following function can be defined:

$$J(\mathcal{U}) = (\mathcal{A}x(k) - \mathcal{R} + \mathcal{B}\mathcal{U})^T Q (\mathcal{A}x(k) - \mathcal{R} + \mathcal{B}\mathcal{U}) + \mathcal{U}\lambda\mathcal{U}. \quad (14)$$

where  $Q$  gives the weighting matrix, with which the tracking performances can be influenced and the intervention is weighted by  $\lambda$ . During the choice of the weight values, the balance should be found between the accurate tracking and the energy consumption (Camacho and Bordons, 2007). Using the cost function (14), the following quadratic optimization problem can be formed:

$$\min_{\mathcal{U}} \frac{1}{2} (\mathcal{U}^T \gamma \mathcal{U} + \omega^T \mathcal{U} + \mathcal{U}^T \alpha + \varepsilon), \quad (15a)$$

such that

$$\phi \mathcal{U} < b \quad \text{and} \quad l_b \leq \mathcal{U} \leq l_u, \quad (15b)$$

where (15b) provides the constraints for the control input, where  $l_b$  gives the lower bound, while  $l_u$  is the upper bound. During the computation of the control signal, constraints can be applied for the states of the vehicle ( $b$ ). Moreover,  $\phi$  serves to select or compute the constrained states of the system. Finally,  $\gamma, \omega, \alpha, \varepsilon$  can be computed from (14) and the prediction horizon is selected to  $N_p = 30$ .

## 4.2 Constraints of the Vehicle

The additional control input ( $\Delta/\alpha$ ) has an impact on the states of the vehicle, which can influence the stable motion. Therefore, constraints are defined for the given states such as the yaw-rate, and lateral acceleration to satisfy the stable motion requirements. However, taking into account the additional control input is challenging since it cannot be considered directly as a disturbance on the input signal, since it has impact on the results of the optimization process and the performances may be decreased. Thus, the additional

control input is built into the computation of the constraints during the optimization, with which the stable motion of the vehicle is achieved.

On the other hand, the additional control input cannot be computed for the whole optimization horizon since measured signals of the system are also needed. The idea to address this problem is to use the computed value at the  $k^{th}$  time step and to suppress the value of the additional control input along the prediction horizon. The prediction process is carried out using the following expression:

$$G_{pred} = \frac{\Delta}{\alpha} \left( 1 - \frac{1}{T_{\Delta}^2 s^2 + 2T_{\Delta} s + 1} \right) \quad (16)$$

where  $T_{\Delta}$  gives the time constant of the given lateral dynamics of the vehicle, which can be determined by the analysis of the dynamics see e.g. (Mondek and Hromčík, 2017). The additional control input prediction is built up of two main parts. The first part is the actual, computed ultra-local model ( $\frac{\Delta}{\alpha}$ ), while the prediction is made using the transfer function ( $G_{\Delta}$ ), which is a second-order term. In this paper, the time constant is set to  $T_{\Delta} = 0.2$ . The prediction process of the additional control input is performed after the discretization of  $G_{\Delta}$ , which results in the matrices  $A_{\Delta}, B_{\Delta}, C_{\Delta}$ .

Since the formulation of the kinematic model does not contain the lateral acceleration or the yaw-rate of the vehicle, it is approximated by (9). Based on (13), the predicted lateral acceleration can be formed as  $a_{y,p} = \mathcal{B}(A_{kin}, B_{kin}, C_a) \mathcal{U}$ , where  $C_a = [0, 0, v_x^2]^T$ . Using the approximated lateral acceleration and the predicted value of the additional control input, the following constraints can be defined along the whole prediction horizon:

$$a_{y,p} + \mathcal{E} \frac{\Delta}{\alpha} \left( 1 - \mathcal{B}(A_{\Delta}, B_{\Delta}, C_{\Delta}, \mathcal{U}_{\Delta}) \right) < b_{ay} \quad (17a)$$

$$a_{y,p} - \mathcal{E} \frac{\Delta}{\alpha} \left( 1 - \mathcal{B}(A_{\Delta}, B_{\Delta}, C_{\Delta}, \mathcal{U}_{\Delta}) \right) > -b_{ay} \quad (17b)$$

where,  $\mathcal{E} = v^2/IE$ , and  $E = [1, 1, \dots, 1]^T$ ,  $b_{ay} \in \mathbb{R}^{N_p \times 1}$  gives the maximum lateral acceleration values.  $\mathcal{B}(A_{\Delta}, B_{\Delta}, C_{\Delta}, \mathcal{U}_{\Delta})$  can be computed as it is described in (13) and the input ( $\mathcal{U}_{\Delta}$ ) is the Heaviside step function. In this paper, the maximum value of the lateral acceleration is set to  $|a_{y,max}| = 7m/s^2$ , with which  $b_{ay} = E a_{y,max}$ .

Furthermore, other states of the vehicle and the maximum values of the steering angle are limited during the optimization process. The constraints for the yaw-rate of the vehicle are set to  $|\dot{\Psi}_{max}| = 0.6rad/s$ . Using the approximation  $\dot{\Psi} = v_x/R$  and (17), the constraints can be determined for the yaw-rate value sim-

ilarly to the lateral acceleration. The steering angles are bounded ( $l_u, l_b$ ) at  $0.4rad$ . The maximum lateral error is defined as:  $y_{max,i} = y_{ref,i} + R_i$  and  $y_{min,i} = y_{ref,i} - R_i$ . The  $R_i$  is chosen in such a way, that the first element ( $R_1$ ) of the  $R$  vector is the highest and the last value ( $R_{Np}$ ) is the lowest. Between the two elements, the values decreased equidistantly, with a step of  $d = (R_1 - R_{Np})/Np$ . The highest value is set to  $R_1 = 0.6m$  and the lowest is  $R_{Np} = 0.05m$ . Using these constraints, the vector of the bounds ( $b$ ) can be created. Finally,  $\phi$  is determined based on the appropriately chosen matrix  $C$ . These constraints serves to guarantee the stable motion of the vehicle and applied for both of the MPC controllers during the simulations.

## Structure and the Calculation of the Derivatives

Finally, in Figure 2 the control structure is presented, which contains the MPC with the kinematic model and uses also the results of the error-based ultra-local model. The MPC computes the input sequence for the system. Using the first value of the input signal, the mathematical formulation of the system, and the actual states, the output of the system can be approximated for the following time step ( $y_{MPC}(k+2)$ ). Moreover, for the computation of the nominal ultra-local model (10b), the derivative signals are also needed, which is illustrated with D.A. in Figure 2. On the other hand, the measurable signals, and the real input of the system are used for the computational process of the ultra-local model. The deviation between these models gives the error-based ultra-local model, which is an additional control input of the system. Finally,  $z^{-1}$  aims to match between the input and the output signals and between the two ultra-local models.

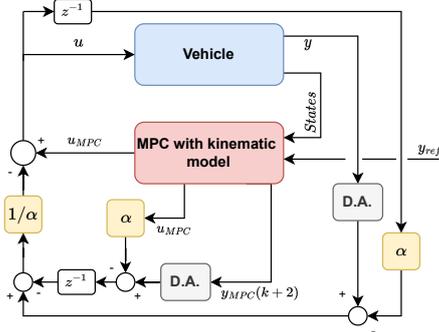


Figure 2: Structure of the combined control algorithm.

It is described in (10), that the derivative of the measured, and the approximated outputs are required.

The accelerations, which are equivalent to the 2<sup>nd</sup> derivative of the outputs, can be calculated as (Polack et al., 2019):

$$a_{y,est} = -\frac{5!}{2T^5} \int_0^T (-T^2 + 6T\tau - 6\tau^2)y(\tau)d\tau \quad (18)$$

where  $T > 0$  must be chosen small and  $x(t)$  denotes the longitudinal position of the vehicle and  $T$  gives the time window of the filter. However, the proposed equations cannot be implemented in practice. Thus, the numerical solution is approximated using the Simpson's rule (Polack et al., 2019):

$$\int_a^b f(x)dx \approx \frac{b-a}{90} \left( 7f(a) + 32f\left(\frac{a+b}{4}\right) + 12f\left(\frac{2(a+b)}{4}\right) + 32f\left(\frac{3(a+b)}{4}\right) + 7f(b) \right) \quad (19)$$

During the implementation of the proposed algorithm the output and the approximated output of the system are derivated using (18), (19).

## 5 SIMULATION

In this section, the algorithm is tested on a vehicle dynamic simulation software, CarMaker. This section aims to show the performances and the effectiveness of the combined solution. Moreover, the control performances using the different models are also compared to each other. The following simulations are performed:

- Kinematic model-based MPC with the error-based ultra-local model
- MPC with the dynamic model of the vehicle
- Kinematic model-based MPC without the error-based ultra-local model

During the simulations, the vehicle is selected to a Tesla Model S. The accurate vehicle parameters can be found in the simulation software, with which the lateral model (3) can be tuned properly. On the other hand, the kinematic lateral model does not require hardly determinable vehicle-related parameters. The whole algorithm is tested through trajectory tracking, which is selected to a lane-change-like reference path. The maximum lateral deviation of the lane changes varies randomly, and also the velocity varies. In Figure 3 the measured velocity profile of the vehicle can be seen.

Figure 4 presents the measured and the reference lateral position of the vehicle. It can be concluded that

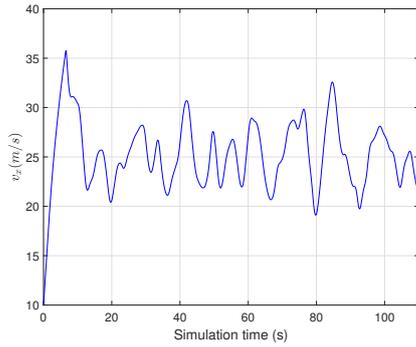


Figure 3: Longitudinal velocity of the vehicle.

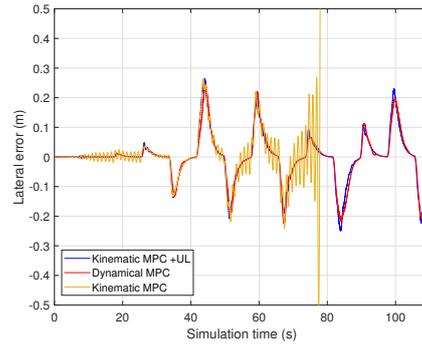


Figure 5: Lateral errors during the test scenarios.

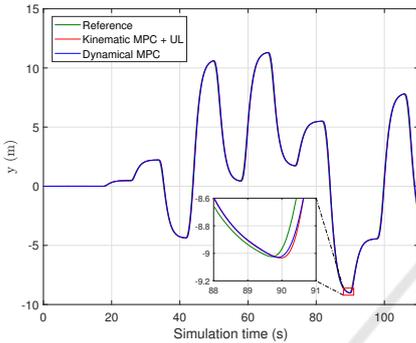


Figure 4: Lateral position of the vehicle.

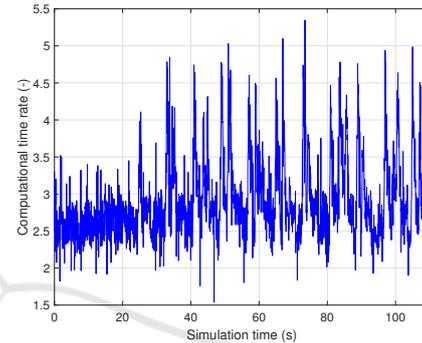


Figure 6: Rate of the computational capacities.

both of the control algorithms can steer the vehicle along the given reference trajectory accurately.

The kinematic MPC, with the error-based ultra-local model, reaches the performances of the dynamical MPC. In Figure 5 the computed lateral errors are presented to make it easier to compare the control algorithms. However, in this case, the results of the kinematic MPC, without the additional control input, are also demonstrated with the yellow line. It can be examined that this algorithm cannot control the vehicle along the given trajectory and it loses its stability. Since the dynamical effects are not modeled and also the steering system is neglected, an oscillation occurs and finally the constraints cannot be met and the MPC is not capable to compute a feasible solution with the given limitations for the states.

Another important aspect is to compare the results in terms of computational capacity. The optimization time saved for the same simulation case using the dynamical MPC ( $T_{dyn,MPC}$ ) and the kinematic model-based MPC with the ultra-local model-based part. Then the rate of the two algorithms is computed as  $C_r = T_{dyn,MPC}/T_{kin,MPC}$ . To eliminate external factors as much as possible, the same simulation case is simulated 10 times and the results are averaged. The results are shown in Figure 6.

Figure 6 depicts that the computational capacity of the kinematic model-based MPC with the ultra-

local model is much lower than the complex, dynamic model-based MPC. In the worst case, it is 1.5 times faster and the highest difference is more than 5x. In the following table, the maximum, minimum, mean, and standard deviation (Std.) of the rate of the computational times are presented.

Highest	Lowest	Mean	Std.
5.35	1.53	2.93	0.547

The computed steering angles can be seen in Figure 7. The blue line represents the results of the MPC, and the red lines show the computed error-based ultra-local model. Using these signals, the vehicle is driven along the predefined lane-change maneuvers.

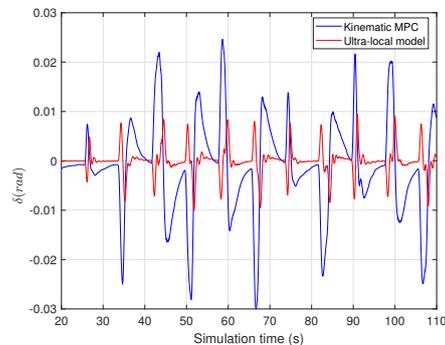


Figure 7: Control inputs during the test scenarios.

Finally, the lateral accelerations are presented. It can be seen, the maximum value of the acceleration reaches  $3m/s^2$ , which is an appropriate value for everyday traffic situations.

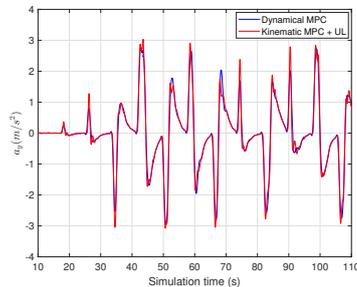


Figure 8: Lateral accelerations during the test scenarios.

## 6 CONCLUSIONS

The paper has presented a novel control design approach, which took the advantages of low computational cost MPC and error-based ultra-local model. The proposed control algorithm was able to guarantee a high-performance level and to take into account the state constraints of the system. The efficiency and the operation of the proposed method have been demonstrated through a vehicle-oriented control problem, trajectory tracking. The designed controller has been compared to a high-computational cost MPC to show the performance level of the presented algorithm. The comparison has been carried out in the high-fidelity simulation software, CarMaker.

## ACKNOWLEDGEMENTS

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 under OTKA Grant Agreement No. K 143599. The research was also supported by the National Research, Development and Innovation Office through the project "Cooperative emergency trajectory design for connected autonomous vehicles" (NKFIH: 2019-2.1.12-TÉT\_VN).

## REFERENCES

Allgöwer, F., Findeisen, R., and Nagy, Z. (2004). Nonlinear model predictive control: From theory to application. *Journal of The Chinese Institute of Chemical Engineers*, 35:299–315.

- Barth, J. M., Condomines, J.-P., Bronz, M., Moschetta, J.-M., Join, C., and Fliess, M. (2020). Model-free control algorithms for micro air vehicles with transitioning flight capabilities. *International Journal of Micro Air Vehicles*, 12.
- Batista, J. G., Souza, D. A., dos Reis, L. L., Filgueiras, L. V., Ramos, K. M., Junior, A. B., and Correia, W. B. (2019). Performance comparison between the PID and LQR controllers applied to a robotic manipulator joint. In *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 479–484.
- Camacho, E. F. and Bordons, C. (2007). Model predictive control. *Springer London*.
- Fliess, M. and Join, C. (2013). Model-free control. *International Journal of Control*, 86(12):2228–2252.
- Ghazally I. Y. Mustafa, Haoping Wang, Y. T. (2019). Model-free adaptive fuzzy logic control for a half-car active suspension system. *Studies in Informatics and Control*, 28.
- Hegedűs, T., Fényes, D., Németh, B., Szabó, Z., and Gáspár, P. (2022). Design of model free control with tuning method on ultra-local model for lateral vehicle control purposes. pages 4101–4106.
- Hegedűs, T., Fényes, D., Szabó, Z., Németh, B., Lukács, L., Csikja, R., and Gáspár, P. (2023). Implementation and design of ultra-local model-based control strategy for autonomous vehicles. *Vehicle System Dynamics*, 0(0):1–24.
- Katriniok, A. and Abel, D. (2011). LTV-MPC approach for lateral vehicle guidance by front steering at the limits of vehicle dynamics. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6828–6833.
- Lima, P. F., Trincavelli, M., Martensson, J., and Wahlberg, B. (2015). Clothoid-based model predictive control for autonomous driving. In *2015 European Control Conference (ECC)*, pages 2983–2990.
- Mondek, M. and Hromčík, M. (2017). Linear analysis of lateral vehicle dynamics. In *2017 21st International Conference on Process Control (PC)*, pages 240–246.
- Polack, P., Altché, F., d'Ándrea Novel, B., and de La Fortelle, A. (2017). The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 812–818.
- Polack, P., Delprat, S., and d'Ándrea Novel, B. (2019). Brake and velocity model-free control on an actual vehicle. *Control Engineering Practice*, 92:104072.
- Rajamani, R. (2005). Vehicle dynamics and control. *Springer*.
- Schwenzer, M., Ay, M., and Abel, D. (2021). Review on model predictive control: an engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117:1327–1349.
- Wang, Z. and Wang, J. (2020). Ultra-local model predictive control: A model-free approach and its application on automated vehicle trajectory tracking. *Control Engineering Practice*, 101:104482.
- Zhou, K. and Doyle, J. C. (1998). *Essentials of Robust Control*. Prentice-Hall.