LPV-Based Control Design using an Error-Based Ultra-Local Model

Tamás Hegedűs *, **Dániel Fényes** *, **Péter Gáspár** *,**

* *HUN-REN Institute for Computer Science and Control, Kende u. 13-17, H-1111 Budapest, Hungary.*
E-mail: [tamas.hegedus;daniel.fenyes;peter.gaspar]@sztaki.hun-ren.hu
** *Department of Control for Transportation and Vehicle Systems, Budapest University of Technology and Economics, Stoczek u. 2, H-1111 Budapest, Hungary.*
E-mail: [peter.gaspar]@kjk.bme.hu

Abstract:
In this paper, an LPV-based control design framework is presented, in which a reformulated ultra-local model is integrated. The limitations of the original ultra-local model-based control structure, are analyzed and a modified version of it is proposed. The key concept of this approach is to integrate the whole modified ultra-local model into the polytopic modeling framework. This results in an extended state-space model, the basis for the LPV control design. The model uncertainty is handled by the ultra-local model, while the stability requirements are guaranteed by the LPV controller. Moreover, the tuning parameter of the ultra-local model is handled as a scheduling parameter of the LPV controller. The effectiveness and operation of the proposed control algorithm are demonstrated through a vehicle-oriented application.

Keywords: ultra-local model, LPV control, trajectory tracking

1. INTRODUCTION AND MOTIVATION

In the field of control and system theory, one of the major challenges is the accurate modeling of the considered system. In the beginning, the systems were modeled as linear, and time-invariant, while the nonlinear behavior was neglected. This approach led to a conservative controller, which had a significant impact on their performance level. To improve the performance, polytopic and linear Parameter Varying (LPV) methods have been developed, as described in Toth (2010). This modeling framework allows the designer to capture the nonlinear dynamics using the set of linear systems. The LPV technique requires the measurement and observation of the scheduling parameters. However, in some cases, this requirement cannot be fulfilled.

In recent years, a new modeling approach started to take place in the field of nonlinear systems: The machine learning-based solutions, see Narendra and Parthasarathy (1990). Unlike the classical solutions, the machine learning techniques do not require a mathematical, physics law-based model of the considered system. These techniques, especially the neural networks, can learn any nonlinear functions, the only constraint is the number of the hidden layers and the neurons. The main drawback of these methods lies behind their black-box nature. The logic and the connections behind the provided model are hard to understand, thus its robustness and stability cannot be proven. This kind of method cannot be used in safety-critical systems, as detailed in Forsberg et al. (2020). To bridge this gap, the combined methods are taking over the place of pure machine learning-based solutions, see Lelko and Nemeth (2024). These techniques combine the advantages of the model-based and the data-driven algorithms. For example, in Fényes et al. (2022) a decision tree and LPV-based approach is presented. The decision tree provides the scheduling parameters for the LPV system based on the available measurements. A trained neural network can be converted into a set of linear systems, as detailed in Lelko et al. (2021). The drawback of this method is that the learning-based algorithm is highly dependent on the quality and quantity of the training data. This means that a lot of measurements are needed to cover the entire operational range of the system considered.

Other solutions could be model-free techniques. The Model-Free Control (MFC) method has been developed by Fliess and JOIN (2014). Basically, this algorithm uses an ultra-local model to capture the nonlinear behavior of the controlled system in real-time. The ultra-local model is computed from the applied control signal and the derivative of the output of the system. The control signal is computed from the ultra-local model and an additional con-

[1] The research was partially supported by the National Research, Development and Innovation Office (NKFIH) through the project 'Design of high performance safe autonomous vehicle systems via integrated robust control and learning-based methods' (2021-1.2.4-TÉT-2022-00065). The work of Daniel Fényes was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. 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).

troller, which aims to guarantee the tracking performance. The MFC structure has been applied for several control problems, see Polack et al. (2019), Scherer et al. (2023). However, the same problem arises as in the case of the machine learning algorithm: robustness and stability. To resolve this issue a combined technique is proposed, which merges the ultra-local model into an LPV framework, see Fenyes et al. (2022). The proposed method is demonstrated through a vehicle-oriented control problem: longitudinal motion. The tire characteristics and the unknown braking system and engine provide a good example to show the effectiveness and the operation of the developed combined method.

The paper is organized as follows: The main idea of the error-based ultra-local model and the nominal model is shown in Section 2. The LPV design and the extended state space representation can be found in Section 3. The simulation results are presented in Section 4, while the whole paper is concluded in Section 5.

2. THE ERROR-BASED ULTRA-LOCAL MODEL

In this section, a brief introduction to the Model-Free Control (MFC) method is given (see Fliess and Join (2009)). In this paper, this control structure will be called the original structure. A nonlinear physical system can be represented by the following differential equation:

$$\dot{x} = f(x, u), \tag{1}$$

where x gives the states, while the control input of the system is provided by u. The core idea behind the Model-Free Control (MFC) framework is that the real system is described by an ultra-local model, which is recomputed at every sample time step. This so-called "phenomenological" model is expressed as Fliess and Join (2013) d'Andrea Novel et al. (2010):

$$y^{(\nu)} = F + \alpha u, \tag{2}$$

where $F \in \mathbb{R}$ describes the unknown part of the system and the input signal is denoted by $u \in \mathbb{R}$. The output of the system is provided by $y \in \mathbb{R}$, while the design parameter of the ultra-local model-based concept is given by $\alpha \in \mathbb{R}$. Finally, the derivative order is represented by ν. In practice, due to measurement noise, the derivative order is typically limited to $\nu = 2$. Based on (2) the ultra-local model can be formulated as:

$$F = y^{(\nu)} - \alpha u. \tag{3}$$

In most cases, the goal of the control algorithm is to achieve a predefined reference value y_{ref}. Using (2) and the ν^{th} derivative of the reference output and error can be defined as:

$$e^{(\nu)} = y^{(\nu)} - y_{ref}^{(\nu)} = F + \alpha u - y_{ref}^{(\nu)}. \tag{4}$$

The goal is to eliminate the error signals ($e^{(\nu)} = 0$). Consequently, the control signal, with which zero error can be achieved is computed as:

$$u = \frac{-F + y_{ref}^{(\nu)}}{\alpha}. \tag{5}$$

This control signal does not guarantee zero tracking error in the steady state. To overcome this limitation of the control structure, an additional classical controller is incorporated to eliminate the tracking error in a steady state. This ensures improved performance and accurate reference tracking:

$$u = \frac{-F + y_{ref}^{(\nu)} + \mathcal{K}(e)}{\alpha}, \tag{6}$$

where $\mathcal{K}(e)$ can be classic controller, such as PID Fliess and Join (2009), where $e = y - y_{ref}$ denotes the tracking error.

Remark In a real model-free setting, no prior knowledge of the system is available, which makes the selection of an appropriate feedback controller particularly challenging. In real-world applications, even after data-driven tuning, this control structure may not guarantee performance specifications. Nevertheless, the fundamental concept offers significant advantages in dealing with the uncertainties in the control loop. Therefore, the goal is to exploit the benefits of this approach while incorporating prior knowledge of the specific system. In the following, the original control structure is enhanced by integrating available system knowledge.

The error-based ultra-local model

The key concept behind the improved structure is to determine a nominal ultra-local model to define an error system. This error system represents the deviation between a predefined nominal model and the real system, which is valid in the given operational point. The formulation of the modified error-based ultra-local model is given as follows:

$$y^{(\nu)} = F + \alpha u, \tag{7a}$$

$$y_{ref}^{(\nu)} = F_{nom} + \alpha u_{nom,ref}, \tag{7b}$$

$$\underbrace{y^{(\nu)} - y_{ref}^{(\nu)}}_{e^{(\nu)}} = \underbrace{F - F_{nom}}_{\Delta} + \underbrace{\alpha u - \alpha u_{nom,ref}}_{\alpha\tilde{u}}, \tag{7c}$$

$$e^{(\nu)} = \Delta + \alpha\tilde{u}, \tag{7d}$$

where the ultra-local model of the real system is given by F, while F_{nom} is the nominal ultra-local model, which is derived from the nominal model of the system. The error-based ultra-local model is computed from the error between the real system and the nominal model. Moreover, y_{ref} is the reference signal, $u_{nom,ref}$ represents the computed reference input signal, which is calculated from the nominal model. During the control of an arbitrary system, the objective is to ensure that the ν^{th} derivative of the error remains zero, which implies:

$$\tilde{u} \approx \frac{-\Delta}{\alpha}. \tag{8}$$

By incorporating information from the nominal model into the ultra-local model-based structure, the error-based ultra-local model can be created, which is denoted by Δ. Similarly to the original ultra-local model-based control structure, the extended version also involves a classical controller:

$$u = -\frac{\Delta}{\alpha} - \mathcal{K}(e, x), \tag{9}$$

where the error is defined as $e = y_{ref} - y$ and the states of the system are denoted by x.

2.1 Nominal model

In this subsection, the nominal model of the longitudinal vehicle dynamics is presented. The objective is to formulate a simplified model, with which the control design process can be performed. The longitudinal acceleration is computed as:

$$m\ddot{x} = F_t(t) + F_b(t) - F_d(t), \tag{10}$$

where m gives the mass of the vehicle, F_t represents the driving force, the braking force is $F_b(t)$. Moreover, the drag force is denoted by $F_d(t)$. In this paper, the control input is defined as the throttle and brake pedal positions of the vehicle, while the system output is the longitudinal velocity. During the computation of the driving and braking forces, a simplified actuator model is considered as:

$$F_t = \frac{A_t(t)}{Ts+1}u_t(t), \quad F_b = -\frac{A_b(t)}{Ts+1}u_b(t), \tag{11}$$

where $A_t(t)$ and $A_b(t)$ provide the amplitude parameters of the transfer functions between the pedal position and the applied force. T is a parameter of the braking system and the transmission of the vehicle. $u_t(t) \in [0, 100]$ gives the throttle position, while the brake pedal position is given by $u_b(t) \in [0, 100]$. In the simulation example, only the aerodynamic drag is considered in the nominal model. Thus the following linearized dynamic equation can be used to describe the longitudinal dynamics:

$$m\ddot{x}(t) = \begin{cases} \frac{A_t(t)}{Ts+1}u_t(t) - 0.5Ac_d\rho P_d(t)\dot{x}(t), \\ -\frac{A_b(t)}{Ts+1}u_b(t) - 0.5Ac_d\rho P_d(t)\dot{x}(t), \end{cases} \tag{12}$$

where A is the cross-sectional area, the drag coefficient is given by c_d, and $P_d(t)$ is a varying parameter related to the longitudinal velocity. Using the mathematical representation of the longitudinal dynamics of the vehicle, the following state space representation can be created:

$$\dot{x}_{nom} = A_n x_{nom} + B_n u, \tag{13}$$

$$y_{nom} = C_n^T x_{nom}, \tag{14}$$

where A_n, B_n, and C_n^T are the nominal system matrices. The input signal is u and the output is denoted by y. The actuator dynamics is represented by the following state space model:

$$\dot{F}_t(t) = \left[-\frac{1}{T}\right] F_t(t) + \left[\frac{A_t(t)}{T}\right] u_t(t)$$

and

$$\dot{F}_b(t) = \left[-\frac{1}{T}\right] F_b(t) + \left[\frac{A_b(t)}{T}\right] u_b(t)$$

$$A_n(\rho) = \begin{bmatrix} -0.5Ac_d\rho P_d(t) & 0 & \frac{1}{m} & \frac{1}{m} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{T} & 0 \\ 0 & 0 & 0 & -\frac{1}{T} \end{bmatrix}, \; B_n(\rho) = \begin{bmatrix} 0 \\ 0 \\ \frac{A_t(t)}{T} \\ \frac{A_b(t)}{T} \end{bmatrix}, \tag{15}$$

The two independent control inputs are merged into one in the following way: if $u(t) > 0$, then $A_t(t) = A_t(t)$ and $A_b(t) = 0$, if $u(t) < 0$, then $A_t(t) = 0$ and $A_b(t) = A_b(t)$. The scheduling vector of the state-space representation are: $\rho = [A_t(t), A_b(t), P_d(t)]$ The state.vector contains: $x_{nom} = [v_x, p_x, \hat{x}_1, \hat{x}_2]$. p_x is the traveled distance, and $\hat{x}_1, \hat{x}_2$ are the inner variables of the braking and transmission systems.

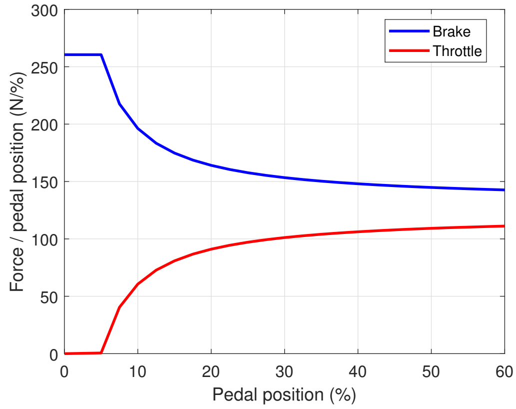

Fig. 1. The characteristics of the brake and throttle forces

2.2 Computation of the reference input and output signals

In this section, the computation process is shown, which is sufficient for the calculation of the nominal ultra-local model. It is mentioned, that the nominal part of the error-based ultra-local model is determined using the nominal model of the real system. In this paper, it is assumed that the derivatives of the outputs are directly measurable and can be computed using the simplified longitudinal dynamics described in (12). Accordingly, the nominal control inputs can be determined from the measured longitudinal acceleration by considering the characteristics of the throttle and brake actuators (see Figure 1). To improve the accuracy of the estimation, the effect of air drag is also taken into account. Furthermore, the nominal longitudinal acceleration ($\dot{y}_{ref}$) can be computed based on the applied pedal positions. In cases where these signals are not available, a filtering algorithm-such as ALIEN filters-can be used for derivative estimation, and the reference control input can be determined using a deadbeat-like controller (see Hegedűs et al. (2024)).

2.3 Selection of the parameter α

The final control input is composed of two main components: the output of the LPV controller and the computed error-based ultra-local model. The tuning parameter α plays a crucial role in the balancing between the LPV-based and the ultra-local model-based part:

- $\alpha \to \infty$: the control input is mainly computed from the LPV-based part, which means that the effect of the error-based ultra-local model is suppressed. If α is selected to a high value, the performance level of the control system cannot be improved.
- $\alpha \to 0$: the influence of the ultra-local model-based part increases during the determination of the control input. If the parameter α is set too low, oscillation may occur, which is not suitable.

The determination of the α is a non-trivial challenge since it cannot be computed analytically. Therefore, an iterative algorithm is used to adjust the optimal value of the tuning parameter.

During the optimization, the squared sum of the tracking error and the squared sum of the control input are considered. The squared sum of the control input aims to prevent

Algorithm 1 Optimization of Tuning Parameter α

Input: α_0, M_{max}, $\Delta\alpha$, γ
Output: Optimized parameter α
for $m = 1$ to M_{max} **do**
  Compute error vector:

$$e_{m,i} = \gamma \|y_{ref,i} - y_i\| + \|u_i\| \tag{16}$$

  if $\frac{1}{N}\sum_i^N e_{m,i} \geq \frac{1}{N}\sum_i^N e_{m-1,i}$ or $m > M_{max}$ **then**
    Terminate iteration.
  end if
  Update α:

$$\alpha_m = \alpha_{m-1} + \Delta\alpha \frac{d\left(\frac{1}{N}\sum_i^N e_{m,i}\right)}{d\alpha}. \tag{17}$$

end for

the choice of parameter α that results in oscillations. In Figure 2 an example is shown for the effect of the tuning parameter.

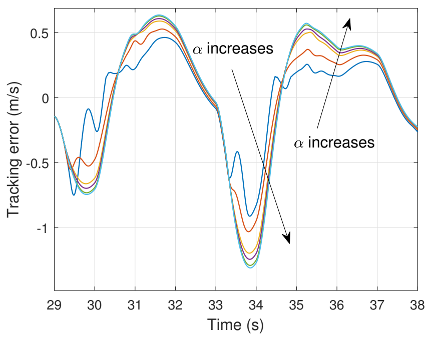

Fig. 2. Effect of the tuning parameter α on the tracking accuracy

Note that the accuracy of the nominal model depends on the actual operational point of the real system, Therefore, in several applications, the accuracy of the control algorithm can be increased by a dynamically varying α parameter see: Hegedűs et al. (2022).

Overall structure of the control algorithm

In this section, the extended state space representation of the system is presented, in which the error-based ultra-local model is incorporated. In this representation, the first derivative of the output is considered ($\nu = 1$). Moreover, it is shown that the extended state space can be simplified in the case when the derivatives are directly measurable.

3. EXTENDED STATE SPACE REPRESENTATION

The external components of the error-based ultra-local model are: $y^{\nu}_{ref} = \dot{y}_{ref}$, and $u_{nom,ref}$. It is important to note that, in general, the nominal input signal is computed using a deadbeat-like controller, which cannot be directly calculated within the classical control framework. To integrate the error-based ultra-local model into the LPV design framework, these signals are considered to be external, measurable disturbances. The inclusion of the error signals $\dot{y}_e$ and the previous control input (u) are more challenging because the first derivative of the output signal is taken into account. To address this issue, a filtering algorithm is incorporated into the state space representation of the system, with which the derivatives of the signals can be approximated. For this purpose, several algorithms can be employed, such as Pade Approximation Brezinski (2002). In this paper, to satisfy implementation-related requirements, the filtering algorithm is defined as a second-order term:

$$G_{f,i}(s) = \frac{s}{T_i^2 s^2 + 2T_i s + 1}, \tag{18}$$

where T_i is the time constant. Using 18 the following state-space representation can be formulated:

$$A_{f,i} = \begin{bmatrix} -2/T_i & 1 \\ -1/T_i^2 & 0 \end{bmatrix}, \quad B_{f,i} = \begin{bmatrix} 1/T_i^2 \\ 0 \end{bmatrix}, C_{f,i}^T = [1 \;\; 0]. \tag{19}$$

The calculation process of the $\dot{y}$ and u can be carried out using the filtering algorithm (18). Note that, the LPV-based controller determines the control, with which the stability requirements can be carried out considering the effect of the error-based ultra-local model, which is composed from: $[\dot{y}, \dot{y}_{ref}, u, u_{nom,ref}]$. In the following, the general form of the extended state-space representation is presented, considering a general state-space representation of the controlled system ($\dot{x} = Ax + bu$):

$$\dot{x}_e = A_e(\hat{\rho})x_e + B_e(\hat{\rho})u_e + B_{e,w}(\hat{\rho})w_e, \tag{20a}$$

$$A_e(\hat{\rho}) = \left[\begin{array}{c|c|c} A & BC_{f,1}^T & -BC_{f,2}^T/\alpha \\ \hline 0_{2\times3} & A_{f,1} & -B_{f,1}C_{f,2}^T/\alpha \\ \hline B_{f,2}A^{1\times3} & 0_{2\times2} & A_{f,2} \end{array}\right], \tag{20b}$$

$$B_e(\hat{\rho}) = \left[\begin{array}{c} B_v \\ \hline 0_{2\times1} \\ \hline 0_{2\times1} \end{array}\right], \quad B_{e,w}(\hat{\rho}) = \left[\begin{array}{c|c} B/\alpha & -B \\ \hline B_{f,1}/\alpha & -B_{f,1} \\ \hline 0_{2\times1} & 0_{2\times1} \end{array}\right], \tag{20c}$$

where $u_e = [u]$, $x_e^T = [x_{nom}, \; x_{f,1}, \; x_{f,2}]$, $w_e^T = [y_{ref}, u_{nom,ref}]$ and $A^{1\times3} = e^T A$, $e^T = [0,1,0]$, $\hat{\rho} = [\rho, \alpha]$. $u = [0, \; 1]x_{f,1}$. The LPV-based control design relies on the extended state space representation. Note that, it is mentioned that the α can be selected dynamically to increase the performances, however, in this paper, it is selected to a constant value. This means, that the parameter α is not considered as a scheduling variable. Further details can be found in the paper Fenyes et al. (2022).

3.1 LPV control design

In the case of the longitudinal dynamics, the derivative of the reference output ($\dot{y}_{ref} = a_{x,ref}$) is computable, thus the filter $G_{f,1}$ is omitted.

The LPV control design has two goals: The first is to guarantee the tracking of the given reference signal. The second is to ensure the stability of the closed-loop system influenced by the external reference signals from the ultra-local model. Thus, the performances of the closed-loop system are:

- Guaranteeing the tracking of the reference speed: $z_1 : (v_{x,ref} - v_x) \rightarrow$ min!
- minimization of the intervention: $z_2 : u \rightarrow$ min!

The presented performances can be fulfilled by carefully selected weighting functions. W_{ref} scales the reference

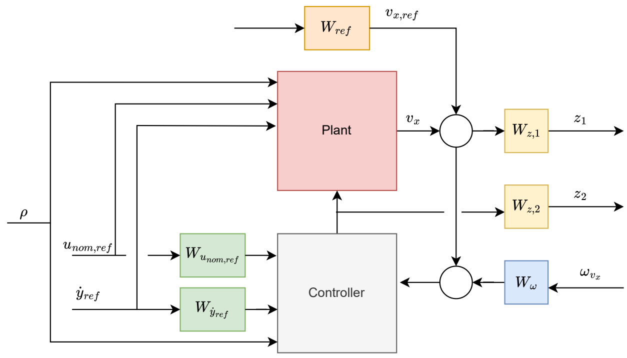

Fig. 3. Augmented LPV system

signal, $W_{z,1}$ guarantess the tracking performance, $W_{z,2}$ is to minimize the intervention, W_ω attenuetes the noise on the measured signal. The last two weighting functions: $W_{u_{nom,ref}}$ and $W_{\dot{y}_{ref}}$ are to scale the external disturbances from the ultra-local model. The interconnected structure is shown in Figure 3.

The augmented plant can be expressed in state-space form as:

$$\dot{x}_{ec} = A_{ec}(\rho)x_{ec} + B_{ec}(\rho)u_{ec} + B_{ec,w}(\rho)w_{ec}, \tag{21a}$$
$$z_{ec} = C_{ec,1}(\rho)x_{ec} + D_{ec,1}(\rho)u_{ec}, \tag{21b}$$
$$y_{ec} = C_{ec,2}(\rho)x_{ec} + D_{ec,2}(\rho)w_{ec,2}, \tag{21c}$$

The control design leads to a quadratic minimization problem, which results in a controller $\mathcal{K}(\rho)$. This controller can guarantee the stability of the closed-loop system by minimizing the induced norm $\mathcal{L}_2$ from the disturbances to the presented performances. The induced norm $\mathcal{L}_2$ should be less than a given value γ.

$$\inf_{K(\rho)} \sup_{\rho \in F_\rho} \sup_{\|w\|_2 \neq 0, w \in \mathcal{L}_2} \frac{\|z\|_2}{\|w\|_2}, \tag{22}$$

In figure 4 the error-based ultra-local model computation can be seen. It can be also observed, that a time delay is applied for the control input, which comes from the LPV-based part, in order to guarantee the signal matching between the control signal and the measurements.

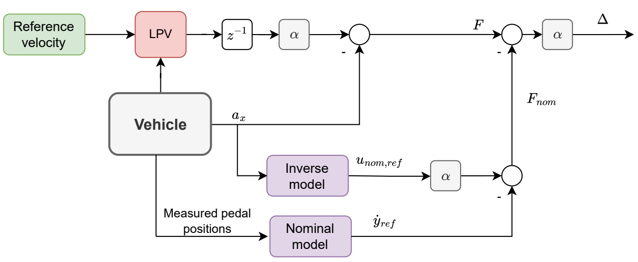

Fig. 4. Structure of the error-based ultra-local model

4. SIMULATION EXAMPLE

In this section, a simulation example is presented to demonstrate the combined LPV and error-based ultra-local control structure. During the simulation example, the tuning parameter is set to $\alpha = 5$. The entire simulation is carried out in CarMaker, a high-fidelity vehicle dynamics simulation software. During the simulation, the reference velocity is chosen randomly, and the goal is to demonstrate the enhanced performance level of the proposed control structure. Firstly, the control inputs for the throttle position are presented in Figure 5. The result of the LPV-based controller is presented in blue, while the red line shows the computed error-based ultra-local model.

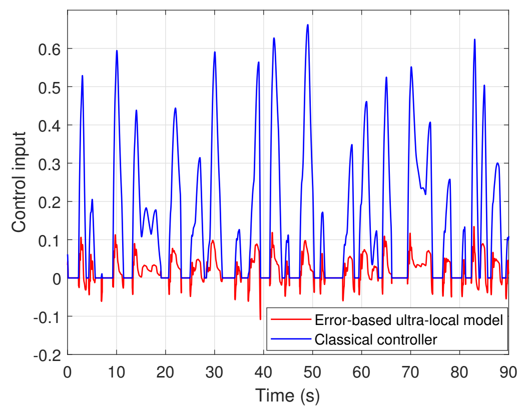

Fig. 5. Throttle position during the simulation

In the next step, the brake position is depicted in Figure 6. Similarly to the previous case, both the classical and the error-based ultra-local model-based parts are presented. It can be concluded that the ultra-local model-based part has higher values at the beginning and at the end of the given reference velocity segment. This means that the response for the reference value change is faster in the case when the ultra-local model is also used.

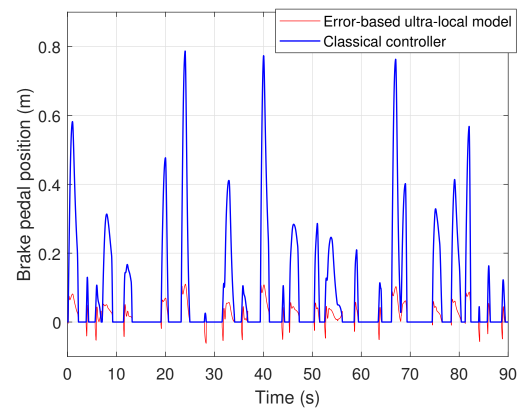

Fig. 6. Brake position during the simulation

In Figure 7 the final control inputs can be seen, where the brake and throttle positions are normalized to 1.

Finally, the performance is compared to a classical-only control structure, specifically the same LPV controller without the error-based ultra-local model. It can be observed that the extended controller reaches better tracking performances. Based on the mean of the sum of errors, the proposed control structure achieves a 38 % reduction in tracking error.

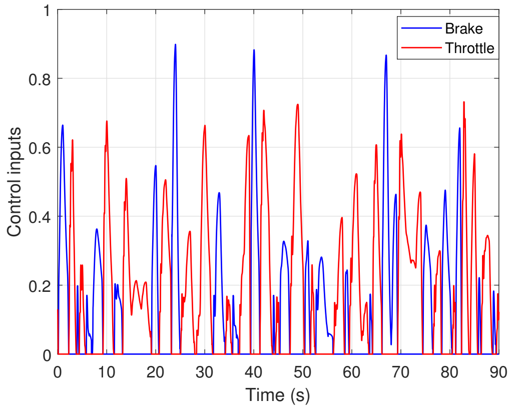

Fig. 7. Control input during the simulation

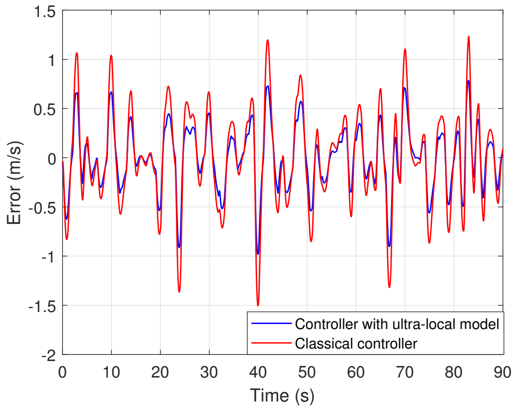

Fig. 8. Error between the measured and the reference value

5. CONCLUSION

In this paper, a combined control approach is proposed, integrating an error-based ultra-local model with an LPV-based control algorithm. An optimization-based iterative algorithm is introduced for tuning the parameter of the error-based ultra-local model. Then, the entire control structure is incorporated into an LPV framework to ensure robust system performance. The effectiveness of the proposed method was validated through a longitudinal control problem for automated vehicles, implemented in the CarMaker high-fidelity simulation environment. The simulation results demonstrate that the ultra-local model-based approach effectively compensates for external disturbances and enhances the accuracy of the conventional control algorithm. These findings highlight the potential of the combined LPV and error-based ultra-local control strategy to improve the adaptability and overall performance of advanced vehicle control systems.

REFERENCES

Brezinski, C. (2002). Computational aspects of linear control. *Springer US*.

d'Andrea Novel, B., Fliess, M., Join, C., Mounier, H., and Steux, B. (2010). A mathematical explanation via 'intelligent' PID controllers of the strange ubiquity of pids. *Proc. 18th Mediterranean Conference on Control and Automation*, 395–400.

Fenyes, D., Hegedus, T., Nemeth, B., Szabo, Z., and Gaspar, P. (2022). Combined lpv and ultra-local model-based control design approach for autonomous vehicles. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, 3303–3308. doi:10.1109/CDC51059.2022.9992983.

Fliess, M. and Join, C. (2009). Model-free control and intelligent PID controllers: Towards a possible trivialization of nonlinear control? *Proc. 15th IFAC Symposium on System Identification, Saint-Malo, France*, 42, 1531–1550.

Fliess, M. and Join, C. (2013). Model-free control. *International Journal of Control*, 86(12), 2228–2252.

Fliess, M. and JOIN, C. (2014). Stability margins and model-free control: A first look. *Proc. 13th European Control Conference*, 454–459.

Forsberg, H., Linden, J., Hjorth, J., Manefjord, T., and Daneshtalab, M. (2020). Challenges in using neural networks in safety-critical applications. In *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, 1–7. doi:10.1109/DASC50938.2020.9256519.

Fényes, D., Németh, B., and and, P.G. (2022). Design of lpv control for autonomous vehicles using the contributions of big data analysis. *International Journal of Control*, 95(7), 1802–1813. doi:10.1080/00207179.2021.1876922. URL `https://doi.org/10.1080/00207179.2021.1876922`.

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. In *2022 American Control Conference (ACC)*, 4101–4106.

Hegedűs, T., Fényes, D., Szabó, Z., Németh, B., Lukács, L., Csikja, R., and Gáspár, P. (2024). Implementation and design of ultra-local model-based control strategy for autonomous vehicles. *Vehicle System Dynamics*, 62(6), 1541–1564. doi:10.1080/00423114.2023.2242530.

Lelko, A., Balazs, N., and Peter, G. (2021). Stability and tracking performance analysis for control systems with feed-forward neural networks. In *2021 European Control Conference (ECC)*, 1485–1490. doi:10.23919/ECC54610.2021.9654983.

Lelko, A. and Nemeth, B. (2024). Optimal motion design for autonomous vehicles with learning aided robust control. *IEEE Transactions on Vehicular Technology*, 73(9), 12638–12651. doi:10.1109/TVT.2024.3394699.

Narendra, K. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1), 4–27. doi:10.1109/72.80202.

Polack, P., Delprat, S., and dAndrea Novel, B. (2019). Brake and velocity model-free control on an actual vehicle. *Control Engineering Practice*, 92, 1–8.

Scherer, P., Othmane, A., and Rudolph, J. (2023). Combining model-based and model-free approaches for the control of an electro-hydraulic system. *Control Engineering Practice*, 133, 105453. doi:https://doi.org/10.1016/j.conengprac.2023.105453.

Toth, R. (2010). *Modeling and identification of linear parameter-varying systems*. Lecture notes in control and information sciences. Springer, Germany. doi:10.1007/978-3-642-13812-6.