

LPV model identification based on error-based ultra-local model for lateral control purposes

Tamás Hegedűs*, Dániel Fényes*, Balázs Németh*,**,
Zoltán Szabó*, Péter Gáspár*,**

* *HUN-REN Institute for Computer Science and Control, Kende u. 13-17, H-1111 Budapest, Hungary. E-mail: [tamas.hegedus;daniel.fenyesh;balazs.nemeth;zoltan.szabo;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: [balazs.nemeth;peter.gaspar]@kjk.bme.hu*

Abstract:

In this paper, a Linear Parameter-Varying (LPV) control algorithm that integrates an error-based ultra-local model is presented. At each iteration, an ultra-local model is computed, serving as the basis for identifying the scheduling variable of the polytopic system. Analysis of the identified models presents a high correlation among the ultra-local models and key system parameters, which are challenging to measure with high accuracy in practice. To overcome this limitation, the deviation from the nominal model is computed using the error-based ultra-local model. The proposed approach is validated through a lateral control problem for automated vehicles, which is implemented in the vehicle dynamics simulation software, CarMaker. The results highlight the potential of combining the LPV control framework with the error-based ultra-local modeling technique to increase the adaptability and performance of control algorithms in dynamic environments.

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

1. INTRODUCTION AND MOTIVATION

One of the major challenges in control theory is the accurate modeling of the considered system. The vast majority of the systems are not linear and time-invariant (LTI) but nonlinear and their parameters can change over time. However, in most cases, only an LTI representation of the system is available, which leads to a more complex and conservative control design process. In the beginning, the classical PID control structure was utilized to control non-LTI systems, by using large phase and amplitude margins, as presented in Diaz-Rodriguez et al. (2009). While this approach can be effective for relatively simple systems, it fails to reach satisfactory performance for highly nonlinear systems. This limitation motivated research on robust control techniques such as $\mathcal{H}_\infty$, see Zhou and Doyle (1998), which can handle unmodeled dynamics and disturbances by minimizing the induced norm between disturbances and

performance criteria. These techniques work well when the ratio between the known and unknown parts is small. However, when the unmodelled/unknown part becomes large, the robust control design provides a conservative solution, which can significantly reduce the reachable performance level.

To address this issue, the polytopic Linear Parameter Varying (LPV) framework was developed as discussed in Mohammadpour and Scherer (2012). This approach models parameter variations as scheduling parameters, allowing controllers to adapt to system changes using Lyapunov-based techniques. There are crucial requirements for LPV control design: the scheduling parameters must be measured or estimated during the operation of the control system. However, in some cases, this information is not available.

To deal with the difficulties of nonlinear and uncertain systems, a new control technique was developed. This method is called Model Free Control (MFC), see Fließ and Join (2013). The basis of the MFC is an ultra-local model, which is continuously updated during the operation of the control system. Basically, the ultra-local model is used to compute an additional input signal, which aims to

¹ The paper was funded by the National Research, Development and Innovation Office under OTKA Grant Agreement No. K 143599. The work of Daniel Fényes was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

² 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).

compensate the unmodelled/unknown part of the system. Although this method has already been applied to several control problems, in some cases, it can destabilize the system. Thus, a new formulation of the ultra-local model-based control structure was introduced, which is called an error-based ultra-local model. This algorithm uses two ultra-local models, one is computed from the measured signals, while the other one is from a nominal model, as detailed in Hegedűs et al. (2022). Although this new formulation can solve some issues, it is still used as an additional control signal, which makes the analysis of the closed-loop system difficult.

In this paper, an LPV control design is presented integrating an error-based ultra-local model to enhance system adaptability and performance. In the original ultra-local model control structure, the computed ultra-local model is directly used as an additional feedback term in the control signal. However, in this paper, the main goal is to exploit the capability of the error-based ultra-local model to catch the error between the actual operational point of the system and the nominal model. The main contribution of this paper is an algorithm, based on the error-based ultra-local model, which can determine the actual operational point of the considered system. The proposed control structure is validated through a vehicle control problem: trajectory tracking. The simulations are performed in the vehicle dynamics software, CarMaker.

The paper is organized as follows: The main idea of the error-based ultra-local model and the nominal model is detailed in Section 2. The control structure and the identification of the actual operating points are explained in Section 3. The simulation results are given in Section 4, while the whole paper is concluded in Section 5.

2. STRUCTURE OF THE ERROR-BASED ULTRA-LOCAL MODEL AND THE NOMINAL SYSTEM

In this section, the original ultra-local model-based control algorithm is presented, then, the extended version is briefly introduced. Moreover, the nominal system is presented through which the presented control structure is validated.

2.1 Error-based ultra-local model

The main idea behind the ultra-local model-based control algorithm is to use an additional control signal to compensate the unmodeled, nonlinear dynamics of the considered system. The ultra-local model is continuously updated at each time step. Originally, the ultra-local model-based control structure is formulated as: Fliess and Join (2009):

$$u = \frac{-F + y_{ref}^{(\nu)}}{\alpha} + \mathcal{K}(e, \hat{x}), \quad (1)$$

where u denotes the control input, $y_{ref}^{(\nu)}$ gives the ν^{th} derivative of the reference output of the system. The ultra-local model, denoted by F , is recomputed at every time step using the two measured signals and typically used as an additional control input. Moreover, $\mathcal{K}(e, \hat{x})$ is a classical feedback controller (e.g., PID or LQR) to eliminate the steady-state tracking errors. The computation of the ultra-local model comes from the following relation:

$$y^{(\nu)} = F + \alpha u. \quad (2)$$

where ν^{th} derivative of the output (y) and the previously applied control input multiplied by a parameter (α).

In some cases, the original ultra-local model can destabilize the system, thus a new formulation was introduced in Hegedűs et al. (2024). The extended structure is called the error-based ultra-local model, which mainly consists of two ultra-local models Hegedűs et al. (2024):

$$y^{(\nu)} = F + \alpha u, \quad (3a)$$

$$y_{ref}^{(\nu)} = F_{nom} + \alpha u_{nom}, \quad (3b)$$

$$\underbrace{y^{(\nu)} - y_{ref}^{(\nu)}}_{e^{(\nu)}} = \underbrace{F - F_{nom}}_{\Delta} + \underbrace{\alpha u - \alpha u_{nom}}_{\alpha \tilde{u}}, \quad (3c)$$

$$e^{(\nu)} = \Delta + \alpha \tilde{u}. \quad (3d)$$

One of the ultra-local models is computed from the measured signals, while the other one is from a nominal model. Using the two ultra-local models, the error-based ultra-local can be computed, which is denoted by Δ . Moreover, using the output of the system and the reference signal (y_{ref}), the tracking error can be calculated. In the context of control design, the objective is to achieve zero tracking error, ensuring that $e^{(\nu)} \rightarrow 0$. Similarly to the original structure, the zero steady-state error is reached through an additional, classical feedback controller:

$$u = -\frac{\Delta}{\alpha} - \mathcal{K}(e, \hat{x}). \quad (4)$$

2.2 Ultra-local model in LPV framework

In this paper, the ultra-local model-based term is not directly used as an additional control input, but it is used to determine the operational point of the system. Based on the identified operational points, a polytopic system is created, which serves as the basis of the control design. A polytopic, LPV system generally can be formulated as described in Toth (2010):

$$\dot{x}(t) = A(\rho)x(t) + B(\rho)u(t), \quad (5a)$$

$$y(t) = C(\rho)x(t) + D(\rho)u(t), \quad (5b)$$

where $A(\rho), B(\rho), C(\rho), D(\rho)$ are parameter dependent system matrices. ρ gives the scheduling variable, while the output of the system is y and the states are given by x . The goal of this paper is to use the error-based ultra-local model to characterize the deviation between the nominal model and the system at the given operational point. Thus, the state-space representation is augmented as follows:

$$\dot{x}(t) = (A_{nom} + \Delta A)x(t) + (B_{nom} + \Delta B)u(t), \quad (6a)$$

$$y(t) = (C_{nom} + \Delta C)x(t) + (D_{nom} + \Delta D)u(t), \quad (6b)$$

where the nominal system matrices $A_{nom}, B_{nom}, C_{nom}, D_{nom}$ are computed from the nominal model. On the other hand, the deviation of the nominal model is formulated through the error-based ultra-local model as:

$$\Delta A = \mathcal{F}_A(\Delta), \quad \Delta B = \mathcal{F}_B(\Delta), \quad (7)$$

where $\mathcal{F}_i$ describes the relationship between the error-based ultra-local model and the deviation of the system matrices.

2.3 Nominal system description

In this subsection, the nominal model is presented, which is the two wheeled bicycle model. The vehicle motion can be described by two main equations Rajamani (2005):

$$I_z \ddot{\psi} = \left(\delta - \beta - \frac{\dot{\psi} l_1}{v} \right) C_1 l_1 - \left(\beta + \frac{\dot{\psi} l_2}{v} \right) C_2 l_2, \quad (8a)$$

$$mv(\dot{\psi} + \dot{\beta}) = \left(\delta - \beta - \frac{\dot{\psi} l_1}{v} \right) C_1 + \left(-\beta + \frac{\dot{\psi} l_2}{v} \right) C_2, \quad (8b)$$

where, $v > 0$ and v_y give to the vehicle longitudinal and lateral velocity, respectively. The parameters m and I_z represent the mass and yaw moment of inertia. The variable β defines the side-slip angle, while $\dot{\psi}$ denotes the yaw rate. The geometric parameters of the vehicle are characterized by l_1 and l_2 , whereas C_1 and C_2 describe the cornering stiffness of the front (1) and rear (2) axles. The system is controlled via the steering angle δ . In the nominal model changes in the cornering stiffnesses (C_1, C_2) are not explicitly considered during operation. The presented vehicle model can be transformed into a parameter-dependent state-space representation, where the states are $x_v = [\dot{\psi}, v_y, p_y]^T$. p_y is the lateral position of the vehicle, the control signal is $u_v = [\delta]^T$, and the scheduling parameter is v .

3. THE CONTROL STRUCTURE AND THE IDENTIFIED SYSTEMS

In (8) the main dynamical equations are presented for the lateral motion of the vehicle. The first equation describes the rotational motion of the vehicle, while the second one describes the lateral acceleration. Thus, two error-based ultra-local models are applied to estimate the error between the nominal model and the real system.

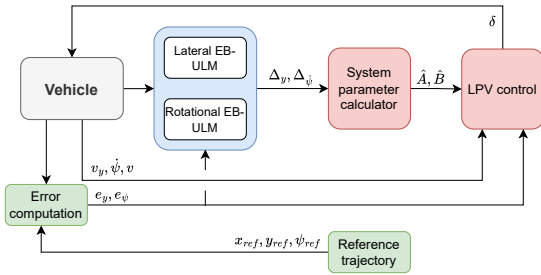


Fig. 1. Structure of the control system

Fig. 1 illustrates the control structure, which consists of three key components:

- **Error Computation:** The deviation between the measured vehicle position and the reference trajectory.
- **Error-Based Ultra-Local Model:** Two ultra-local models are computed using measurements and the nominal model.
- **Control Layer:** LPV-based controller and a system parameter identification layer, which utilizes the error-based ultra-local models (EB-ULM).

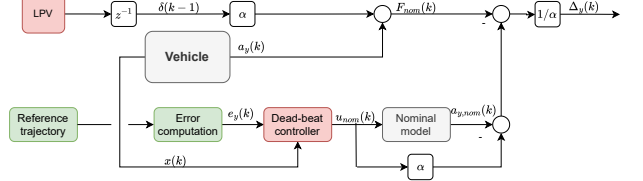


Fig. 2. Error-based ultra-local model

Furthermore, a more detailed structure is illustrated for the error-based ultra-local model computation in Fig. 2.

In Fig. 2 ultra-local model for the lateral acceleration of the vehicle is computed, where the derivative order of the output is set to 2, $\ddot{y}_1 = a_y$. The second ultra-local model is computed similarly, however, in that case, the output is $\ddot{y}_2 = \ddot{\psi}$. The ultra-local models use the control input, which is the applied steering angle $u = \delta$. The nominal ultra-local model is computed from the state of the vehicle, the nominal model, and the dead-beat controller, which is briefly described in the following subsection. In this paper, considering the two main dynamical equations of the system, two error-based ultra-local models are computed as described in 3:

$$\Delta_y = (a_y - a_{y,nom}) - (\alpha\delta - \alpha\delta_{nom}) \quad (9)$$

$$\Delta_{\dot{\psi}} = (\ddot{\psi} - \ddot{\psi}_{nom}) - (\alpha\delta - \alpha\delta_{nom}) \quad (10)$$

3.1 Dead-beat controller

In this subsection, the calculation of the nominal control input (u_{nom}) is presented. Using the nominal model of the system, a discrete state space representation can be created Cassandras and Lafortune (1999):

$$x_d(t+1) = \Phi x_d(t) + \Gamma u_d(t), \quad (11a)$$

$$y_d(t) = C^T x_d(t). \quad (11b)$$

where the Φ , Γ , C^T are the discrete state matrices, which are computed from the continuous model using the sample time $T_s = 0.02s$. The computation of the nominal control input is detailed in Hegedűs et al. (2024).

In this concept, both the translational and angular motion are considered. The ultra-local model for the translational motion is determined from the lateral error and the predicted lateral positions, while the angular motion is also considered through the nominal steering angle see: Hegedűs et al. (2024). This means, that using the nominal steering angle and the dynamical equations, the nominal ultra-local model can be computed for both cases.

Note that, the computation of derivatives of the signal with unspecified noise is challenging. To address this, an ALIEN filter is employed to estimate the derivatives, see Polack et al. (2019). Moreover, the design of parameter α is crucial since it scales the signals. The scaling parameter is chosen iteratively to a constant value. Detailed description can be found in Hegedűs et al. (2022).

3.2 Data collection and the identified models

The nominal vehicle model is constructed using parameters derived from the Tesla Model S, which is from the CarMaker simulation software. To explore the unmodeled

dynamics of the lateral dynamics and to find the connection with the error-based ultra-local models, several test scenarios have been conducted in the vehicle dynamics simulation software, CarMaker. During the test scenarios, a chirp signal is used with different amplitudes to cover the whole operational range:

$$y_{ref}(t) = A(t) \cos(\omega_0 (f_0 t + f_1 t^2) + \phi_0), \quad (12)$$

where $A(t)$ is adjusted to the physical limits of the vehicle and the longitudinal velocity. In this paper $\phi_0 = 0$, moreover, f_1 is selected to 0.01 and f_0 is 1. Finally, ω_0 is selected to 0.5. Moreover, during the data generation, the amplitude of the reference lateral position (y_{ref}) is decreased along with the frequency, which means $A(t) = A_0 - \gamma t$, where γ characterizes the rate of the change of the amplitude of the reference signal. Using the saved dataset, lateral models can be identified in operational ranges, which are identified by the error-based ultra-local models. In Fig. 3 an example is shown for the different models with blue lines, while the nominal model is presented by the red line, these differences come from mainly the change of the cornering stiffness.

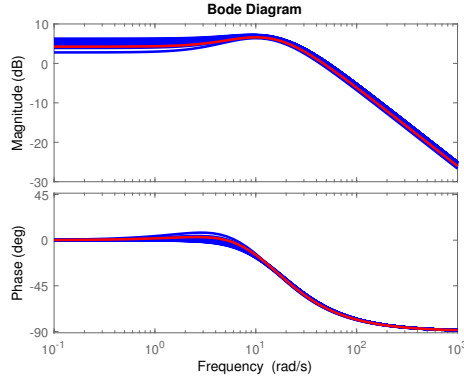


Fig. 3. Bode diagram for $v=10m/s$

The resulted different models (see Fig. 3) are depicted in terms of error-based ultra-local models. In this example, various amplitudes are presented at a fixed frequency of $1rad/s$. The figure shows that the amplitude response of the models is well-defined by the ultra-local models, which indicates that the model can be estimated through the computed ultra-local models at each operational point.

It is shown in Fig. 4(a), that the nominal model depicted in red can be identified in the middle, where both ultra-local models are zero. In the nominal case, the longitudinal velocity is selected to $10m/s$. Moreover, the other models are presented with blue, which comes from the different cornering stiffness. Thus, the actual model is formulated as a deviation from the nominal model, which is represented by blue dots in the figure.

In the next step, the nominal model is augmented with the identified error model by the ultra-local models. To simplify this task, only the cornering stiffness of the model is considered since it is the main source of the nonlinear behavior. Note that the longitudinal velocity (v) can also change, however it is a measurable signal. Therefore, the characteristics are analyzed in terms of the longitudinal velocity. Note that this does not imply that cornering stiffness is inherently velocity-dependent. However, the

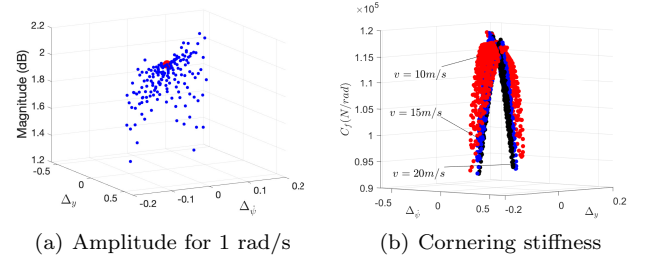


Fig. 4. Cornering stiffness and the magnitude

ultra-local models used to estimate the cornering stiffness are velocity-dependent. Fig. 4(b) illustrates an example of cornering stiffness values at different longitudinal velocities. Moreover, Fig. 4(b) shows that the error-based ultra-local model estimation of stiffness is highly velocity-dependent. The red dots represent measurement points at $v = 10m/s$, the blue dots give the values at velocity $v = 15m/s$, while the black dots belong to measurements with $v = 20m/s$.

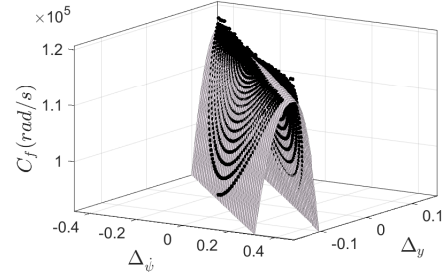


Fig. 5. Example for the fitted characteristics

In Fig. 5 an example is presented for the estimation of the model parameters in terms of the error-based ultra-local models. Using the measurements, it can be seen that the characteristics can be approximated by a second-order polynomial.

$$\mathcal{F}_C(\Delta_{\dot{\psi}}, \Delta_y) = a_1 \Delta_{\dot{\psi}} + a_2 \Delta_y + a_3 \Delta_{\dot{\psi}}^2 + a_4 \Delta_y^2 + a_5 \Delta_{\dot{\psi}} \Delta_y, \quad (13)$$

where the parameters of the fitted polynomial are given by: a_1, a_2, a_3, a_4, a_5 , and $\mathcal{F}_C$ describes the characteristics. Using the LPV formulation of the system:

$$\dot{x}(t) = (A_{nom} + \Delta A(\mathcal{F}_C(\Delta_{\dot{\psi}}, \Delta_y)))x(t) + \quad (14a)$$

$$(B_{nom} + \Delta B(\mathcal{F}_C(\Delta_{\dot{\psi}}, \Delta_y)))u(t), \quad (14b)$$

where $\Delta A, \Delta B$ comes from the nominal model of the system, and $\mathcal{F}_v$ describes the velocity dependent characteristics.

3.3 LPV control design

The LPV control design aims to guarantee the accurate trajectory tracking of the vehicle. The polytopic LPV model has 7 scheduling parameter $\rho = [\Delta a_{11}, \Delta a_{12}, \Delta a_{21}, \Delta a_{22}, \Delta b_1, \Delta b_2, v_x]$. $\Delta a_{11} \dots \Delta a_{22}$ are the uncertain part of the matrix A_v derived from the error-based ultra-local model. Similarly, $\Delta b_1, \Delta b_2$ are the uncertain part of the matrix B_v , see (14). The measurement

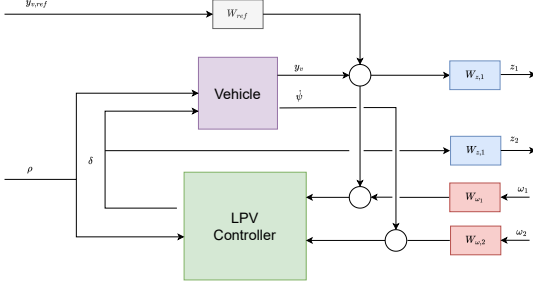


Fig. 6. Structure of LPV controller

vector contains the states: $y = [\dot{\psi}, p_y]$. The required performances of the controller are defined as:

- *Minimization of the lateral error* As a primary goal of the LPV control design, the error between the measured lateral position p_y and the reference signal $p_{y,ref}$ must be minimized:

$$z_1 = p_{y,ref} - p_y, \quad |z_1| \rightarrow \min, \quad (15)$$

- *Minimization of the steering intervention* As secondary goal, the intervention - in this specific case, the steering angle - must be minimized in order to meet the system's limitations:

$$z_2 = \delta, \quad |z_2| \rightarrow \min. \quad (16)$$

Detailed performances can be reached by applying appropriate weighting functions, W_{ref} to scale the reference signal for the controller. $W_{z,1}$ and $W_{z,2}$ aims to reach to performance defined above. The weighting functions $W_{\omega,1}, W_{\omega,2}$ are to attenuate the noises on the measured signals. The augmented system is shown in Fig. 6.

The design process leads to a quadratic optimization problem. The solution is a controller $K(\rho)$ that guarantees the required performances and closed-loop stability of the system. As an additional requirement, the induced norm from the disturbances to the performances must be less than a given value γ .

$$\inf_{K(\rho)} \sup_{\rho \in F_\rho} \sup_{\|w\|_2 \neq 0, w \in \mathcal{L}_2} \frac{\|z\|_2}{\|w\|_2}, \quad (17)$$

where F_ρ bounds the scheduling variables. The computed controller $K(v_x, \rho)$ is formed as:

$$\dot{x}_K = A_K(\rho)x_K + B_K(\rho)y_K, \quad (18a)$$

$$u = C_K(\rho)x_K + D_K(\rho)y_K, \quad (18b)$$

where $A_K(\rho), B_K(\rho)$ and $C_K(\rho), D_K(\rho)$ are scheduling variable dependent matrices.

4. SIMULATION EXAMPLE

This section presents the simulation results for trajectory tracking. The entire validation process is carried out in the CarMaker simulator. The Tesla Model S, which is a widely used passenger vehicle, is selected for simulation purposes. In the first simulation example, the vehicle is driven along an F1 race track Yas Marina and it is shown that using the proposed solution, the control problem can be solved with high performance. Furthermore, in the second simulation, the impact of varying external parameters on the proposed method is analyzed.

4.1 Trajectory tracking of the vehicle

Firstly, the reference trajectory and the computed lateral error can be examined in Fig. 7.

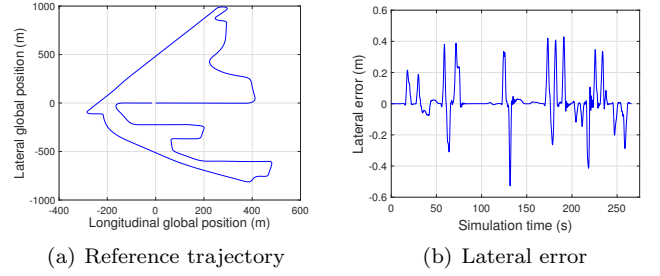


Fig. 7. Reference trajectory and the computed lateral error

The maximum lateral tracking error of the vehicle reaches 0.4 m, which is acceptable for this application, as the vehicle operates near its physical limits to demonstrate the performance of the error-based ultra-local model-selector algorithm. Moreover, in Fig. 8 the velocity profile of the vehicle and the steering angle can be examined. The velocity profile is generated using the built-in driver model provided by the CarMaker simulation software. The velocity varies significantly during the simulation, ensuring comprehensive validation of the control algorithm within the entire range.

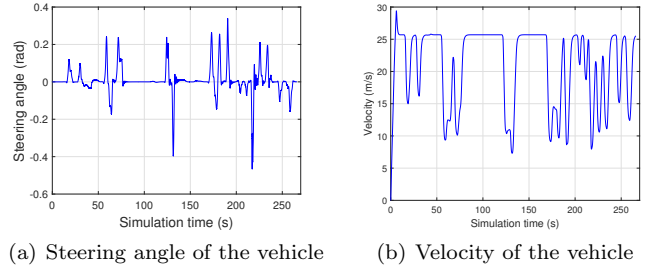


Fig. 8. Velocity and the steering angle of the vehicle

Finally, the translational and angular error-based ultra-local models, computed during the simulation, are illustrated in Fig. 9. The results show that the vehicle reaches various operational points, and the controller can effectively control the vehicle along the predefined path.

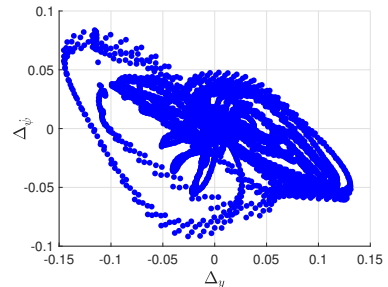


Fig. 9. The computed error-based ultra-local models

4.2 Changing external parameters

The effect of the varying parameters is examined. Since the lateral dynamics of the vehicle are highly influenced

by the friction coefficients through the tire characteristics Pacejka (2004) simulation examples are compared to each other when $\mu = 1, \mu = 0.6, \mu = 0.4$. Note that, in the nominal case $\mu = 1$.

During the design process of the LPV controller, 40 grid points are defined in terms of the two error-based ultra-local models such as the deviation of the nominal model in terms of the tire stiffness varies between $1.2 \cdot C \dots 0.25 \cdot C$, where C gives the nominal value. The minimum velocity value is selected to $v = 5m/s$, while the maximum velocity is $v = 30m/s$, and it is divided into 30 grid points. The goal is to demonstrate the current operational points of the system during the control along the race track. In Fig. 10, the operational points can be seen, which are provided by the error-based ultra-local models. The velocity is limited during the simulation example to $v_{max} = 26m/s$. Moreover, the velocity profile is decreased in the case, when the friction coefficient is decreased.

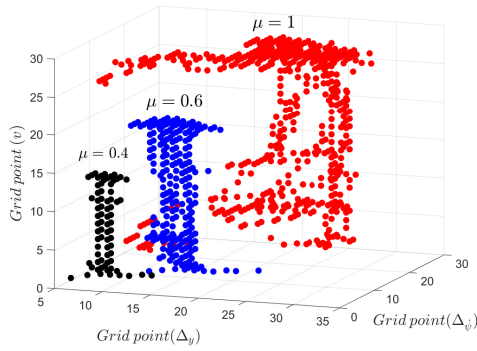


Fig. 10. Grid points during the simulation

Figure 10 demonstrates that the operating points are well-separated across different friction coefficients. This indicates that the proposed algorithm can effectively detect external factors influencing tire characteristics. Additionally, Fig. 5 shows that as the friction coefficient decreases, in the identified operational points tire stiffness is reduced. This means, that the operational points are correctly adjusted using the results from the error-based ultra-local models. Moreover, in Fig. 11 the accelerations can be seen in both x and y directions.

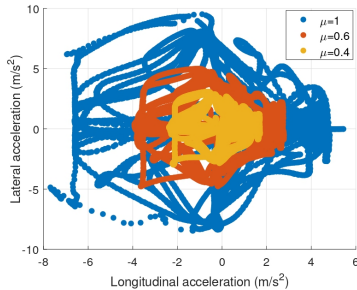


Fig. 11. Lateral and longitudinal acceleration

Fig. 11 illustrates the acceleration values in both longitudinal and lateral directions, showing that the vehicle operates close to its physical limits. It can be seen, that the lateral acceleration of the vehicle nearly reaches $10m/s^2$, which is close to the maximum value achievable by a vehicle.

5. CONCLUSION

This paper presented an LPV control algorithm in which the actual operational point is determined using error-based ultra-local models of the system. An ultra-local model is computed at each iteration step, serving as the basis for determining the scheduling variable of the polytopic system. Analysis of the identified models showed that a high correlation can be observed among the ultra-local models and two key system parameters, which cannot be directly measured with high accuracy. To address this limitation, the model deviation is formulated based on the error-based ultra-local model. The proposed algorithm was validated through a lateral control problem for automated vehicles, implemented in CarMaker. The results demonstrate that the ultra-local model-based approach effectively handles external disturbances and ensures robust system operation. These findings highlight the potential of the combined LPV and error-based ultra-local method in enhancing the adaptability and performance of the control algorithm.

REFERENCES

- Cassandras, C.G. and Lafontaine, S. (1999). Introduction to discrete event systems. In *The Kluwer International Series on Discrete Event Dynamic Systems*.
- Diaz-Rodriguez, I.D., Han, S., and Bhattacharyya, S.P. (2009). *Analytical Design of PID Controllers*. Springer Nature, 1 edition.
- 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.
- 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.
- Mohammadpour, J. and Scherer, C.W. (2012). *Control of Linear Parameter Varying Systems with Applications*. Springer-Verlag New York.
- Pacejka, H.B. (2004). *Tyre and vehicle dynamics*. Elsevier Butterworth-Heinemann, Oxford.
- 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.
- Rajamani, R. (2005). *Vehicle dynamics and control*. Springer.
- 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.
- Zhou, K. and Doyle, J.C. (1998). *Essentials of Robust Control*. Prentice-Hall.