Tuning Parameter-free Model Predictive Control with Nonlinear Internal Model Control Structure for Vehicle Lateral Control

Adorjan Kovacs* **and Istvan Vajk**

Department of Automation and Applied Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Műegyetem rkp. 3, H-1111 Budapest, Hungary; vajk@aut.bme.hu
*Corresponding author: adorjan.kovacs@aut.bme.hu

Abstract: The paper presents a new methodology for the lateral control of autonomous vehicles. The proposed cascade structure realizes two main components: a model predictive control (MPC)-based outer loop and an internal model control (IMC) based inner loop. In the outer loop, a unique model predictive control is introduced that eliminates the tuning parameters of the system by introducing a hierarchical optimization system. Each cost function of the hierarchical optimization focuses on minimizing a physical phenomenon. The inner loop handles system dynamics and nonlinearities, providing a robust system against external disturbances and parameter changes. After presenting the proposed structure, proper comparisons were performed: firstly, to see the advantages of the tuning parameter-free method and secondly, to highlight the benefits of the IMC-based method. Finally, the whole system is compared to a reference controller, available in MatLab.

Keywords: model predictive control; internal model control; minimal tuning parameters; vehicle lateral control

1 Introduction

The lateral control problem of vehicles is one of the main emphasized questions of autonomous vehicles. One of the biggest goals of autonomous vehicles is to increase safety on roads. That means the vehicle control methods should be prepared to handle unexpected or unmeasured effects such as external disturbances, parameter changes, and suddenly changing environments. Therefore, the nonlinearities and the dynamics of the vehicle should be considered, giving a solution that is universal under all circumstances. It is supposed that the path is given as a reference, e.g., calculated by a receding horizon control algorithm [1] or a dynamic optimal control problem [2], so the path planning part of the problem is not included in this paper.

Several solutions handle the nonlinearities, parameter uncertainties, and dynamics of the vehicle, even using machine learning or fuzzy techniques [3, 4]. Adaptive control methods manage model imperfections using the multivariable fixed point iteration method [5]. An iterative feedback tuning controller can handle the strong nonlinearities of systems [6]. Integral backstepping control realizes a feedback control rule for the problem [7], ensuring stability based on the Lyapunov theory. Two parameters are weighting the lateral position and the orientation errors for the feedback loop that should be tuned for the controller. The feedback linearization method leads to a chained system that can be handled by a linear matrix inequalities problem using the peak-to-peak performance approach [8]. However, this method includes a trial-and-error-based parameter tuning method, which provides knowledge of the behavior of the system only in the tested cases. Another feedback solution, the potential-field-based method, was introduced in [9], but only a proportional-derivative controller is tuned for the feedback control. This method also lacks the usage of existing knowledge of the model. The flatness-based method deals with the dynamics of the vehicle, but only linear tire models are included [10, 11]. The proper knowledge of the nominal model parameters is crucial in this method, and a tuning process should also be performed on the gains. The main disadvantage of the different feedback-based methods [8] is that system noises can result in unnecessary control actions compared to methods that consider future references.

The feedforward-feedback method was introduced in [13], which is a virtual potential field-based solution. However, the steering control signal is determined from three independent signals (yaw damping, lane-keeping, and feedforward branch), which are hard to handle if the system reaches its rate limit or final value limit. A path planning and tracking algorithm realizes both feedforward and feedback parts of the control, but separately. The physical limitations are handled by the curve-based feedback loop [14].

The model predictive controller (MPC) is a method that integrates the feedforward and feedback loops into one system. This method can determine the control signal based on optimization, using the existing knowledge about the system: its model with accurate complexity (including nonlinearities and dynamics) and its parameters [15]. The methods presented in [2] and [16] use linearization around the prescribed nominal trajectory to gain a real-time solvable problem. However, this method operates with more than a dozen of parameters, and the control structure excludes direct feedback, so the reaction of the controller to sudden disturbances or changes could be improved. The parameter changes can be handled with an adaptive MPC [17], but the adaptation rule presupposes that specific parameters are measured. The MPC method can be formulated based on the input-output variables and internal states such as yaw-rate [18]. The advantage of the second approach is that the behavior of the vehicle can be controlled directly, concerning the states, despite the indirect methods.

The main contributions of this paper include a methodology that can eliminate the intuitively tuned parameters from MPC controllers, providing non-specific control rules. The optimization is performed based on physical phenomena instead of summing up different expressions with weighting coefficients in the cost function. The proposed algorithm includes a novel hierarchization method that ensures feasibility for the parameter-free approach. This method is placed in a cascade control structure [19, 20], formulating the outer loop. The inner loop handles the dynamics and the nonlinearities of the system, providing robustness against external disturbances and parameter changes. The proposed control approach is compared with the classical MPC methods to see its advantages. The parameter-free method is examined in the simulation, comparing the outer loop. Then, the inner loop is compared to see the performance of the IMC structure. The whole structure was compared to the lane-keeping assist (LKA) reference controller available in MATLAB.

In the following, in Section 2, the system modeling approaches are detailed. After, Section 3 introduces the solution of the proposed structure for dynamics handling together with the custom solution. The classical and the proposed model predictive approaches are detailed in Section 4. The controllers developed for the comparison-based evaluation are introduced in Section 5. The proposed and the reference controllers are compared and evaluated in simulation, and the results are written in Section 6. Finally, the conclusions are gathered in Section 7.

2 Modeling Considerations for Lateral Vehicle Control

In this section, the modeling considerations are presented. The proposed controller uses the kinematic model in the outer and the dynamic model in the inner loop. The simulation framework uses the nonlinear dynamic bicycle model.

2.1 Kinematic Model and the Frenet Frame

The kinematic unicycle model can determine the planar behavior of the vehicle. This model is described in the Frenet frame, as can be seen in Fig. 1. This frame defines the states of the vehicle in a path-based coordinate system with three parameters: the distance from the reference path (d), the orientation compared to the orientation of the reference path (ψ_p), and the distance taken along the reference path (s). The kinematic behavior of the vehicle can be described by a nonlinear state equation system [21]:

$$\dot{s} = \frac{\cos\psi_p}{1 - dC(s)} u_x, \qquad \dot{d} = u_x \sin\psi_p, \qquad \dot{\psi}_p = r - C(s)\frac{\cos\psi_p}{1 - dC(s)} u_x, \tag{1}$$

where u_x is the longitudinal speed, r is the yaw-rate (r=ψ, where ψ is the orientation of the vehicle), and C(s) is the curvature of the path.

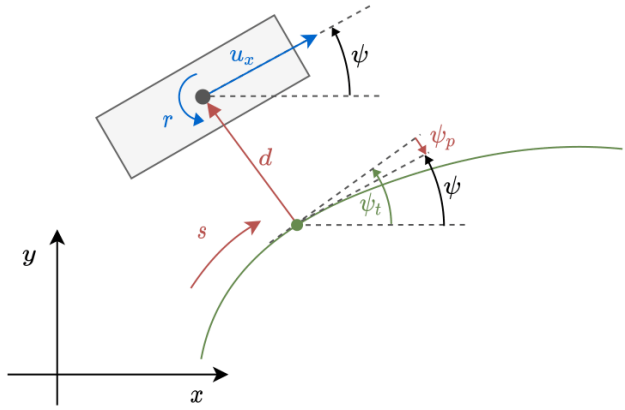

Figure 1
The unicycle in the Frenet frame

It is assumed that the vehicle goes with constant velocity to make the latter controller comparison methods clearer by omitting the longitudinal dynamics. Also, the small-angle assumptions and the first-order Taylor-series approximation can be used on this model [20]. The derivative of the yaw-rate $\dot{r} = \ddot{\psi} = \rho$ is chosen to be the control signal since it describes the control effort performed during the maneuver. By using ρ, the linear state equation can be derived for the Frenet frame model:

$$\begin{bmatrix} \dot{r} \\ \dot{d} \\ \dot{\psi}_p \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & u_x \\ 1 & -C(s)^2 u_x & 0 \end{bmatrix} \begin{bmatrix} r \\ d \\ \psi_p \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \rho + \begin{bmatrix} 0 \\ 0 \\ -C(s) u_x \end{bmatrix}. \tag{2}$$

This linear equation system can be used for state prediction. The future state values of the model can be determined by using a pre-known input vector and the initial state values [20].

2.2 Dynamic Bicycle Model

The dynamic nonlinear bicycle model is needed to have proper knowledge of the behavior of the vehicle. This model can be seen in Fig. 2. The state equations of this model can be derived by writing up the forces and moments balance on the center of the gravity (COG):

$$\dot{u}_y = \frac{F_{yF}\cos(\delta) + F_{yR}}{m} - r u_x, \qquad \dot{r} = \frac{a F_{yF}\cos(\delta) - b F_{yR}}{I_z}, \tag{3}$$

where u_y is the lateral speed, F_{yi}, i={F, R} is the lateral tire force for the front (F), and the rear (R) wheels, m is the mass of the vehicle, I_z is the inertia around axis z, a is the distance between the front axle and the COG, b is the distance between the rear axle and the COG, and δ is the road wheel angle.

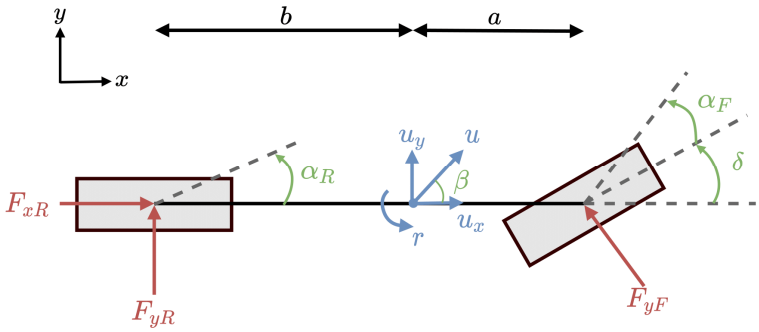

Figure 2
The bicycle model of the vehicle

The nonlinear tire model is created by a two-parameter approximation [22] of the Pacejka tire model [23] to determine the lateral tire forces:

$$F_{yt} = F_{zt}\mu \cdot \sin(c_{lat}\arctan(-b_{lat}\alpha_t)), \quad i \in [F, R], \tag{4}$$

where clat is the shape factor, and b_{lat} is the stiffness factor, α_i is the tire sideslip, and F_{zi} is the vertical force, calculated from the geometry of the model:

$$F_{zF} = mg\frac{b}{a+b}, \qquad F_{zR} = mg\frac{a}{a+b}. \tag{5}$$

The tire sideslips can be determined based on geometrical considerations:

$$\alpha_F = \arctan\frac{u_y + ar}{u_x}, \qquad \alpha_R = \arctan\frac{u_y - br}{u_x}. \tag{6}$$

It is common to handle the vehicle with the linearized model, which can be gained by the small angle assumptions and substituting the tire model to a linear one. This linear tire model can be described by only one parameter, the cornering stiffness $F_{yi} = C_iF_{zi}$, i={F, R} [17]. This way, the linearized dynamic model can be derived from Eq. 3:

$$\dot{u}_y = -\frac{C_F + C_R}{mu_x}u_y - \left(u_x + \frac{aC_F - bC_R}{mu_x}\right)r + \frac{C_F}{m}\delta$$
$$\dot{r} = -\frac{aC_F - bC_R}{I_z u_x}u_y - \frac{a^2C_F + b^2C_R}{I_z u_x}r + \frac{aC_F}{I_z}\delta. \tag{7}$$

The nominal values of the vehicle model used in the simulation can be found in Table 1, together with the parameter names and units.

3 Handling Model Dynamics

Two ways of handling the dynamics of the system are presented in this paper. The widespread method among model predictive controls is that the model with its dynamics is included in the prediction. It means that the linearized dynamic model should have the problem manageable with the convex optimization methods, or global nonlinear solvers should be included to handle the model without linearization. The other way is that the dynamics are not included in the prediction, only the kinematics, creating a cascade structure. In this structure, there is an outer loop for handling the predictive control with a kinematic approach (considering limitations derived from the system dynamics) and an inner loop driving the dynamics of the system.

Table 1
Parameters of the vehicle

Symbol	Name	Value
m	Vehicle mass	1523 kg
I_z	Inertia around z-axis	2330 kgm^2
a	Distance between COG and front axle	1.5 m
b	Distance between COG and rear axle	1.2 m
c_{lat}	Shape factor	1.472
b_{lat}	Stiffness factor	10.87
$\|\delta\|_{max}$	Maximum of road wheel angle	1.05 rad
$\|d\delta/dt\|_{max}$	Maximum steepness of road wheel angle	1.35 rad/sec

3.1 Linearized Dynamic Model

The first presented method handles the dynamics by linearization. The linearized dynamic model can be written up in the Frenet frame, so the steering wheel angle can be determined directly from this model using an MPC formulation. The state equations are gained from the linearized Frenet frame (Eq. 2) and linearized dynamic model of Eq. 7 using the state vector $x^d = [d, \psi_p, u_y, r]$:

$$\begin{bmatrix} \dot{d} \\ \dot{\psi}_p \\ \dot{u}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & u_x & 0 & 0 \\ c_k^2 u_x & 0 & 0 & 1 \\ 0 & 0 & -\frac{C_F + C_R}{m u_x} & -u_x - \frac{aC_F - bC_R}{m u_x} \\ 0 & 0 & -\frac{aC_F - bC_R}{I_z u_x} & -\frac{a^2 C_F + b^2 C_R}{I_z u_x} \end{bmatrix} \begin{bmatrix} d \\ \psi_p \\ u_y \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{C_F}{m} \\ \frac{aC_F}{I_z} \end{bmatrix} \delta + \begin{bmatrix} 0 \\ -c_k u_x \\ 0 \\ 0 \end{bmatrix}. \tag{8}$$

The control structure of the MPC using the linearized dynamic model can be seen in Fig. 3. The localization block is responsible for determining the Frenet frame state variables and the curvature of the reference path for the prediction and control horizon. The dynamic MPC uses the model described in Eq. 8 for determining the requested control signal (δ_c). This MPC can be both the

parameter-free and the classical method in this approach (as these two types are detailed later in Section 4. It should be noted that in this structure, the errors are not fed back directly. Only the prediction-based part compensates for them, which naturally introduces a delay in the reaction.

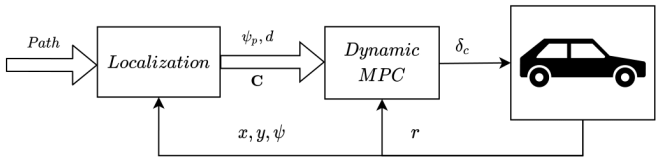

Figure 3
Control structure using the linearized dynamic model

3.2 Nonlinear Dynamic Model-based Feedback Structure

In this section, a different dynamics model handling method is proposed. This method handles the problem in a cascade structure, as can be seen in Fig. 4. The outer loop is a model predictive method using the linearized kinematic model (Eq. 2), and the inner loop realizes an internal model control (IMC) structure. The kinematic-based MPC does not determine the required road wheel angle but determines the required ρ, which is the most important state variable of the system concerning the lateral behavior. This MPC can also use the parameter-free approach or the classical method detailed later. It should be noted that proper limitations should be used in the outer loop to ensure feasibility in this cascade structure.

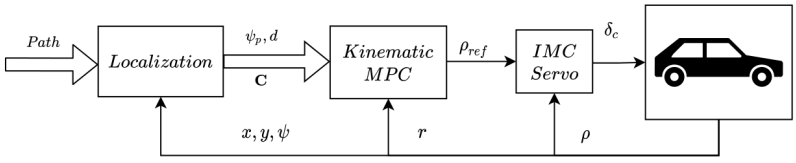

Figure 4
Control structure using kinematic and dynamic models in a cascade structure

The inner IMC structure can be seen in Fig. 5. A realizable inverse is placed on the feedforward branch that can determine the required road wheel angle (RWA). Here, the physical limitations (rate limit and final value limit) of the given system can be enforced. The inverse of the model is calculated based on the nonlinear model of the vehicle (Eq. 3) by solving the nonlinear equation for ρ_{ref}, using numerical approximation. The calculated RWA is then actuated in the vehicle and inputted to a model connected parallel with the plant. Then, the difference between the model and the plant is fed back through an autoregressive-like filter. This filter is responsible for noise suppression, using the following equation:

$$\rho_{fb}(k) = a_f\big(\rho(k) - \rho_m(k)\big) + \big(1 - a_f\big)\rho_{fb}(k-1), \tag{9}$$

where a_f is the filter parameter, k is the discrete-time step-index, ρ is the value measured on the vehicle, and ρ_m is the value calculated by the model. The whole feedback loop is responsible for compensating the external noises and the effect of parameter mismatch. The presented structure exceeds the classical MPC method with a feedback loop. This system can react faster to the disturbances since the most critical internal state parameter is controlled in the inner loop. Additionally, this method can be extended to handle multi-actuator systems [19].

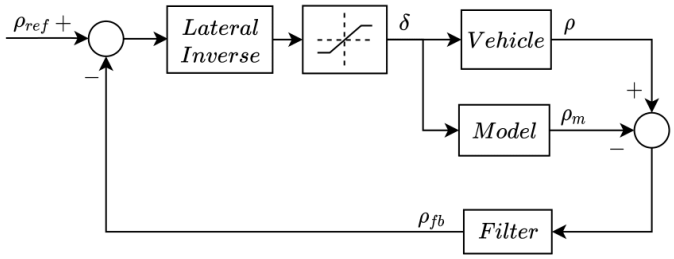

Figure 5
The inner loop realizing IMC

4 Model Predictive Control

The model predictive control method is an advanced technique widely used in various fields of optimal control problems [24]. The Model Predictive Control approach mentioned above will be detailed in this section by presenting the classical MPC method. The paper focuses on the main disadvantage of the classical MPC methods: they have cost functions with mixing values by tuning weights. The proposed parameter-free, hierarchical method that answers this problem will be detailed in this section.

4.1 Classical Model Predictive Control

Model predictive control is the most common approach among predictive controllers. The most obvious use case of predictive controllers is the discrete-time version with finite prediction and control horizon. For the sake of simplicity, the prediction and the control horizons are defined to be equal.

In general, the cost functions of predictive controllers include two parts. The control signals and the reference tracking error are included in these cost functions [24]. Defining N as the length of the horizon, N_x as the number of the

states, N_u as the number of inputs, N_y as the number of outputs, $\mathbf{x}=[x_1,x_2,\ldots,x_{Nx}]$ as the state vector, $\mathbf{u}=[u_1,u_2,\ldots,u_{Nu}]$ as the input vector, and $\mathbf{y}=[y_1,y_2,\ldots,y_{Ny}]$ as the output vector, the general MPC problem formulation can be given. Minimize:

$$J\big(x(0),u(1\ldots N)\big)=\sum_{j=1}^{N_y}\sum_{k=1}^{N}W_{ej}(k)e_j^2(k)+\sum_{j=1}^{N_u}\sum_{k=1}^{N}W_{uj}(k)\Delta u_j^2(k), \tag{10}$$

where $e_j(k)$ is the difference between the j^{th} reference and the predicted state value at the k^{th} time step ($e_j(k)=y_j(k)-y_j^{ref}(k)$), and W_{ej} is the corresponding weight. For the control signal, $\Delta u_j(k)^2$ corresponds to the control effort performed at the j^{th} input at the k^{th} time step, and $W_{uj}(k)$ is the corresponding weight.

The minimization should be performed subject to constraints coming from the state equations of the controlled system:

$$\begin{aligned}\mathbf{x}(k+1)&=f\big(\mathbf{x}(k),\mathbf{u}(k)\big),\quad k=1,\ldots,N\\ \mathbf{y}(k)&=g\big(\mathbf{x}(k),\mathbf{u}(k)\big),\quad k=1,\ldots,N,\end{aligned} \tag{11}$$

and to the constraints derived from the limitations of the states, the inputs, and the outputs.

It can be seen that this cost function formulation has $N\cdot(N_y+N_u)$ weighting parameters. These weights provide the possibility for the designers of the controller to determine different weighting strategies in the cost function. The simplest solution is when the weights are constant for the whole horizon for each input or output. In some cases, the cost concerning the final state is highlighted compared to the running cost.

The existence of this amount of tuning parameters is twofold. On the one hand, the system performance can be maximized in predefined specific scenarios by finding the proper parameter tuning. On the other hand, the cost function including these parameters is a mixture of different values (considering physical meaning) on a different scale. Even if these values are normalized in some approaches, it is hard to interpret the real meaning of the cost function in the control environment, and it is not defined by physical law. However, there is no proper method given to find this parameter tuning. Additionally, there is no insurance that using the found parameter set, the performance of the system will remain if the test scenario or the system parameters change.

Linear state equations are created in Eq. 11. In our case, the linearized kinematic or the dynamic model (Eqs. 2 and 8) can be used for the state prediction. The future states and outputs can be determined in a closed form, using the linear equations if the future inputs are known. The linearized problem results in convex quadratic programming (QP) optimization problem. The problem complexity is crucial concerning the real-time applicability of the control method [25]. Due to the improvement of the available computing capacities, this problem can be solved in real-time, so the MPC method has become a widespread solution [24].

4.2 Parameter-free Model Predictive Control

The proposed method eliminates the weights and creates a cost function that has a physical interpretation. This is reached by decomposing the mixed cost function presented in Eq. 10. For the actual lateral control problem, the cost function should be used with the following variables:

$$\mathbf{y} = [\psi_p, d], \qquad \mathbf{u} = r, \qquad \Delta\mathbf{u} = \rho, \tag{12}$$

According to the approach of a chauffeur, the path following is performed by minimizing the lateral and orientation error at a certain looking ahead distance. In the ideal case, these errors can be driven to zero during the control horizon so that these parts can be transformed to equality constraints. Finally, it results in that the remaining part includes only the control effort that was requested at the control input.

If the case is not ideal, this transformation of the cost function may cause infeasibility. A hierarchic solver method is introduced in order to solve this feasibility problem. This method drives the system step-by-step towards feasibility. Each equality constraint is first introduced as a cost function to minimize the distance from reaching equality. If equality is reached, it is introduced as a constraint while the following constraint is transformed into a cost function. After introducing all constraints that continuously maintain feasibility, the original cost function minimizing the control effort can be used in the optimization.

The outputs of the lateral control problem are formulating an integrator chain since the lateral error is connected to orientation via integration, as can be seen in Eq. 2. This chain determines the order of the introduction of the constraints: firstly, the orientation constraint is satisfied, then the position, to prevent overshoot.

In the following, the optimization problems of the sequential algorithm are given for the kinematic model defined in Eq. 2. Using the notations $\boldsymbol{\rho}=[\rho_1,\rho_2,\ldots,\rho_N]$, and the state vector $\mathbf{x}=[r,d,\psi_p]$, the first optimization can be formulated:

$$\begin{aligned} &\min_{\boldsymbol{\rho}} |\psi_{pN}(\boldsymbol{\rho})| \\ s.t.:\ &|\rho_i| \leq \rho_{max}, \qquad i = 1,\ldots,N \\ &|r_i(\boldsymbol{\rho})| \leq r_{max}, \quad i = 1,\ldots,N, \end{aligned} \tag{13}$$

where the limitations are considered for the yaw rate and its derivative, coming from the physics of the vehicle, and the state values are calculated in a closed form using the measured states and the linear models presented in section 2. After ensuring the orientational constraint, the lateral error is minimized by replacing the cost function with $|d_N(\boldsymbol{\rho})|$ and adding an equality constraint $\psi_{pN}(\boldsymbol{\rho})=0$ to equation (13). Finally, in the third optimization, the control effort is minimized, transforming the last goal into a constraint:

$$\min_{\boldsymbol{\rho}} \sum_{i=1}^{N} \rho_i^2$$

$$s.t.: \quad |\rho_i| \leq \rho_{max}, \qquad i = 1, \dots, N$$

$$|r_i(\boldsymbol{\rho})| \leq r_{max}, \quad i = 1, \dots, N$$

$$d_N(\boldsymbol{\rho}) = 0$$

$$\psi_{pN}(\boldsymbol{\rho}) = 0. \tag{14}$$

In this hierarchical method, the algorithm performs the following optimization only if the minimization reaches zero, showing that the constraint is feasible. After the optimization, the first element of the optimal control vector is actuated, realizing the receding horizon approach.

This way, the weights are eliminated, but on the other hand, instead of one, three optimizations should be performed to ensure feasibility. Since the model is stable and the problem is feasible, the controller is stable [24]. However, using the linearized model, the optimization problems are created to be Convex problems, so they have unique solutions and can be solved in real-time [26].

5 MPC Controllers for Comparison

Firstly, the outer and the inner loop of the proposed method will be examined separately, and then the whole structure will be compared to a nominal solution. In this section, the three MPC models used as a base for the proper comparison are presented to support the presented cost function and dynamics handling approach.

5.1 Kinematic MPC with Mixed Cost

In this case, the classical MPC with mixed cost function uses the kinematic bicycle model presented in Eq. 2, within the structure presented in Fig. 4. This controller is the reference for the outer loop comparison, denoted as MIXIMC. The optimization problem is formulated as follows:

$$\min_{\boldsymbol{\rho}} \sum_{i=1}^{N} \left(K_d d_i^2(\boldsymbol{\rho}) + K_{\psi_p} \psi_{p_i}^2(\boldsymbol{\rho}) + K_\rho \rho_i^2 \right)$$

$$s.t.: \quad |\rho_i| \leq \rho_{max}, \qquad i = 1, \dots, N$$

$$|r_i(\boldsymbol{\rho})| \leq r_{max}, \quad i = 1, \dots, N, \tag{15}$$

where K_d $K_{\psi p}$, and K_ρ are the weights for the lateral error, orientation error, and yaw acceleration, respectively.

In this case, it should be noted that these weights are considered constant over the control horizon. However, in some cases, better performance can be reached by having weights that are changing over the horizon. It results that the cost function of the optimization problem in Eq. 14 may contain $3N$ parameters.

5.2 Dynamic MPC with Parameter-Free Cost Function

The second comparison focuses on the presented inner loop approach, the IMC structure. The reference controller for this comparison is created based on the linear dynamics model described in Section 3.1, using the structure presented in Fig. 3. This controller is denoted as PFD. Since the model includes the road wheel angle, it serves as an input for the system. The optimization problem is formulated similarly to the sequential, hierarchical approach presented in Section 4.2.

Using the dynamic model, the MPC handles the system dynamics and determines the control signal, using the parameter-free approach, but without having feedback for the internal states of the system. The cost function is calculated similarly to the kinematic MPC, based on ρ for a proper comparison. The only difference is that the linear dynamic model is used instead of the kinematic model. However, this approach does not consider the nonlinearities in the system and does not have direct feedback for the dynamics behavior.

5.3 LKA Subsystem

The publicly available most complex and advanced controller is chosen for the complete comparison of the system. The lane-keeping assist (LKA) subsystem [27] includes the linearized dynamic model (Eq. 8) expanded with state estimation for handling the input-output disturbances.

This is an adaptive model predictive control structure, implemented using the Frenet-frame. The MPC formulation of this system is quite similar to the one presented in the previous Section. The disturbance rejection is realized by estimating the plant model and the controller states based on a disturbance model and the measurement noise model, using a linear-time-varying Kalman filter (LTVKF). The state estimation introduces further tuning parameters since two gain matrices are needed for its algorithm. Additionally, this system uses scale factors that the controller designer should also specify.

In this paper, the default values of this subsystem were used (estimator gains, scale factors, etc.). Only the cost function weights were tuned during the comparison method.

6 Simulation Results

In this section, the simulation results will be presented. Firstly, the simulation environment and the evaluation methods are detailed, followed by the three comparisons for the inner loop, the outer loop, and the whole proposed structure.

6.1 Simulation Environment

The simulation environment was implemented in MatLab & Simulink. The plant was modeled using the nonlinear dynamic bicycle model using the nonlinear tire model, presented in Section 2.2, using the nominal parameters (Table 1). The simulation run with fixed step size (dt = 0.002 s), using the ode4 solver. The optimization problems were implemented as MatLab function blocks, using the Optimization Toolbox of MatLab. The controller runs on a lower frequency with f_c =50 [Hz]. The horizon was set to be N = 15 (similar to the LKA reference controller), and the discrete step of the linear system prediction was set to be dt_p = 0.05 [s]. The filter parameter (α_f) was 0.3, which is sufficient against the numerical errors of the simulation.

Due to the structure of the simulation software, it is easy to modify the parameters of the plant, emulating the mismatch between the controller model and the plant. Also, external disturbances can be added to the model for testing the disturbance rejection performance of the controller.

6.2 Evaluation Methods

In this paper, two types of evaluation methods were used to compare. Scalar-based evaluation gives a scalar number as a result of a successful measurement. The scenario-based comparison is performed based on a higher level overlook on the system, where a single scalar value is not enough to characterize the system. The following integral and maximal values were considered for the scalar evaluation of the presented controllers:

- Max error: $e_{max} = \max|e(k)|$, $e=\{d,\Psi_p\}$, calculating the maximum absolute value of the error. The errors are the lateral and orientation deviation at the k^{th} time step.
- Error integral: $e_{int} = dt\ \Sigma e^2(k)$, $e=\{d,\Psi_p\}$, the discrete-time integral of the squared value of the error.
- Max control value: $u_{max} = \max|u(k)|$, the maximal value of the actual control input of the system (that is usually included in the mixed cost function or is at the end of the hierarchic solution).
- Control integral: $u_{int} = dt\ \Sigma u^2(k)$, the discrete-time integral of the squared value of the control input.

The scenario-based comparison is based on the examination of the behavior of the vehicle during different predefined test scenarios [28]. Concerning the lateral control, the overshooting and the setting ability of the reference tracking were investigated. Three test cases were defined for examining the controllers: straight-line following with initial lateral error (dy), straight-line following with initial orientation error (dψ), and the lane change maneuver (LC).

These tests can give a picture of the controller upon its state error rejection and path-following performance. Also, these tests were expanded with further examinations with disturbance rejection and handling the plant parameter changes.

6.3 Comparison of the Cost Functions

Firstly, the outer loops were compared, implementing the proposed IMC-based inner loop for the dynamics handling as it is described in Section 3.1. The tuning parameter-free (PF) method was compared with the MPC with the mixed cost function-based (MIX) method to examine the outer loop. The three parameters of the MIX controller were tuned so that during the lane-change maneuver, it reaches the same control effort as the PF has, resulting in the parameter tuning: $K_d = 6$, $K_p = 0.5$, $K_\psi = 10$. The performed maneuver can be seen in Fig. 6.

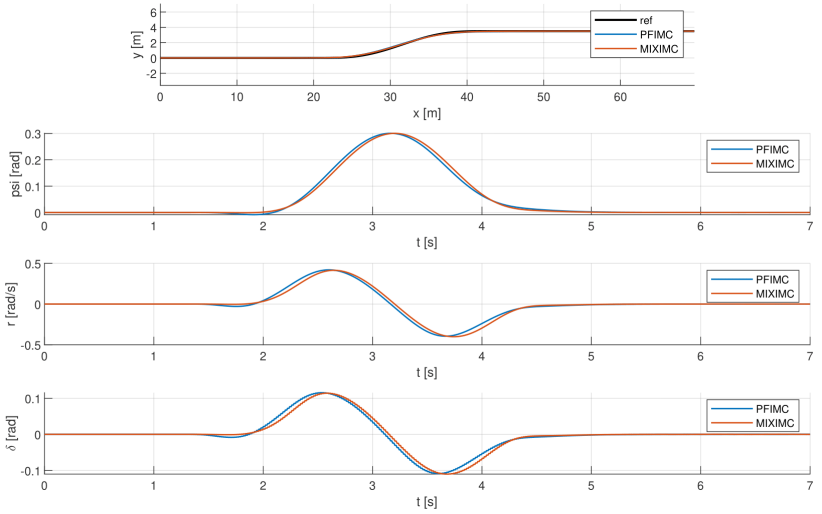

Figure 6
The trajectories of the Parameter-free and the Mixed MPC methods performing the lane change maneuver while using the same lateral control effort

Another scenario was performed, using the same parameter set tuned for the lane change to see the sensitivity of the parameter tuning. The results of the lateral error test (dy) can be seen in Fig. 7.

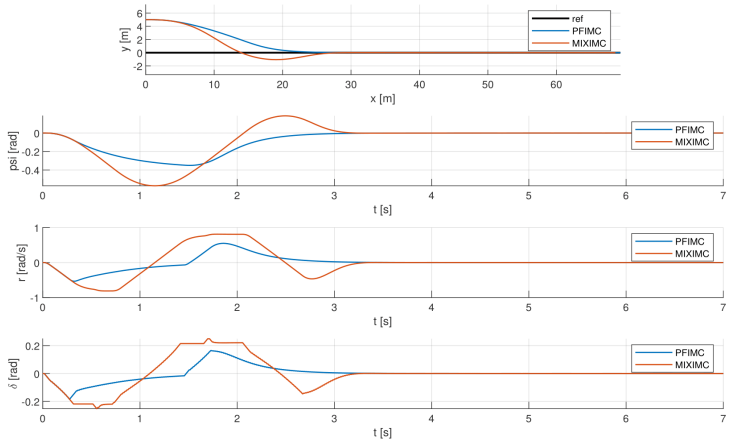

Figure 7
Comparison of lateral error elimination with the parameter set tuned for the lane change maneuver

Table 2
Maneuver evaluation of controllers

Man.	Cont.	u_{int}	u_{max}	d_{int}	d_{max}	Ψ_{pint}	Ψ_{pmax}
LC	PF	1.263	1.271	$2.10 \cdot 10^{-2}$	$1.43 \cdot 10^{-1}$	$7.76 \cdot 10^{-4}$	$2.84 \cdot 10^{-2}$
	MIX	1.251	1.262	$1.26 \cdot 10^{-2}$	$1.07 \cdot 10^{-1}$	$8.55 \cdot 10^{-4}$	$2.94 \cdot 10^{-2}$
dy	PF	2.740	2.108	24.15	5	0.132	0.350
	MIX	7.437	2.220	20.51	5	0.288	0.571

Both maneuvers were evaluated, and the results can be seen in Table 2. It can be seen that in the LC maneuver, as it was the goal, both controllers perform with almost identical control effort (u_{int}). Also, in the first test case, the MIX controller performs better in all the evaluations corresponding to the control input and the lateral error. The PF controller beats only the orientation error. There is a big difference between the controllers in the second test case. The MIX controller has an aperiodic setting in the position and the orientation; therefore, the results of the evaluations are significantly worse. The parameter-free method is now shown to be independent of the test scenario, providing an aperiodic setting in all the test cases. This stability and reliability are a great advantage among predictive controllers, even when the MIX-based approach performs better concerning the errors or the control signal.

6.4 Comparison of Dynamics Handling

In this section, the dynamics handling solutions, presented in Section 3: the proposed tuning parameter-free (PF) method with IMC in the inner loop is compared with the method that uses the linearized dynamic model with the PF

method in the optimization problem formulation. Therefore, only the dynamics handling method differs between the two approaches.

The first test case was the orientation error rejection (dψ) under low-mu conditions. The parameters of the test case were: μ=0.5, $\psi_0 = \pi/6$ [rad], and u_x=10 [m/s] The results of this test can be seen in Fig. 8.

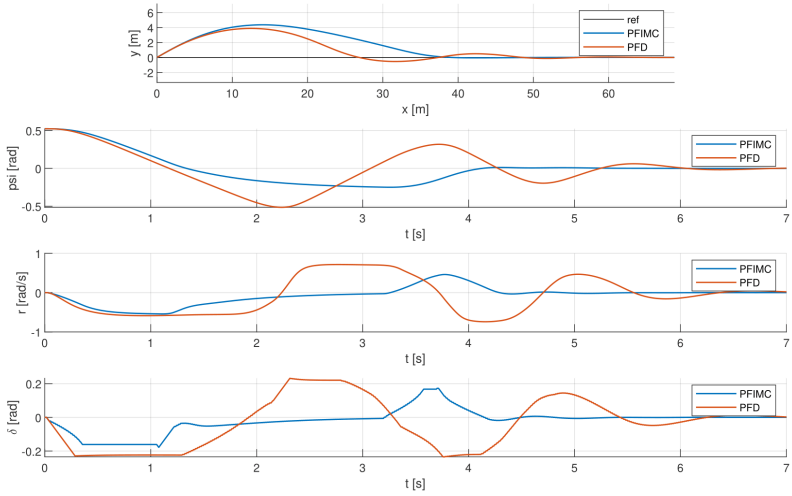

Figure 8
Comparison of the Dynamic model-based MPC and the kinematic-IMC-based controllers in low-mu situation

The IMC-based controller can handle the nonlinearities of the bicycle model successfully, even if the vehicle to road coefficient changes. This induces that at steering movements with high amplitude, the vehicle gets closer to its limits. The linearized dynamic model-based method controller results in a periodic setting in the errors. However, the IMC-based method can handle the nonlinearities with an aperiodic setting.

The second test was performed during straight-line following, examining the external disturbance handling ability of the controller. In this test case, the vehicle ran straight, with constant speed (u_x=10 [m/s]), and then at time t = 0.5 [s], a constant torque disturbance (M_d = 9000 [Nm]) around axis Z was added, inducing yaw moment into the system. The results of this test can be seen in Figure 9.

Both controllers compensate for the disturbance by turning the steering wheel in the proper direction. Due to the IMC loop, the proposed algorithm can react much faster to the disturbance, resulting in total disturbance rejection, eliminating the position error of the vehicle. However, the dynamic model-based controller has a significant constant lateral error, resulting from its structure since only the output states are fed back within the MPC method. To sum up, the proposed IMC method

can handle the nonlinearities and parameter changes, together with the appearing unmeasured external disturbances of the vehicle.

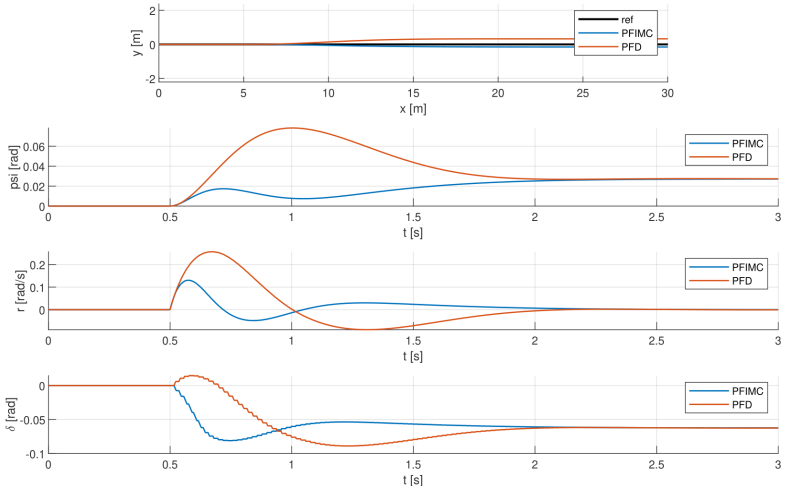

Figure 9
Comparison of the Dynamic model-based MPC and the kinematic-IMC-based controllers under Torque disturbance

6.5 Comparison with the LKA Subsystem

Finally, the proposed controller structure is compared with the LKA subsystem. Since the LKA subsystem should have the same control frequency and prediction frequency, both were set to be $f_c = 50$ [Hz], $dt_p = 0.05$ [s]. The parameters of the LKA subsystem were tuned similarly to the method presented in Section 6.3. The weight of the manipulated variables rate was set to be 4, and the output variables weight (concerning the default scaling factor given by the subsystem) was equally 1. The test was the dynamic lane change test performed on low mu ($\mu = 0.6$). The results of this test can be seen in Figure 10. and the scalar evaluations in Table 3.

Table 3
Numerical results comparison of LKA and PF-IMC under low mu

Controller	u_{int}	u_{max}	d_{int}	d_{max}	Ψ_{pint}	Ψ_{pmax}
PF-IMC	2.112	1.538	$1.524 \cdot 10^{-3}$	$3.591 \cdot 10^{-2}$	$2.634 \cdot 10^{-2}$	$1.516 \cdot 10^{-2}$
LKA	2.488	1.930	$4.510 \cdot 10^{-3}$	$6.536 \cdot 10^{-2}$	$4.168 \cdot 10^{-4}$	$2.049 \cdot 10^{-2}$

The results show that both systems can sufficiently perform the maneuver. However, the LKA system has an overshoot at the end of the maneuver, resulting in a small oscillation in the control signal. Also, it is significant that the proposed controller performs better in all the points of comparison.

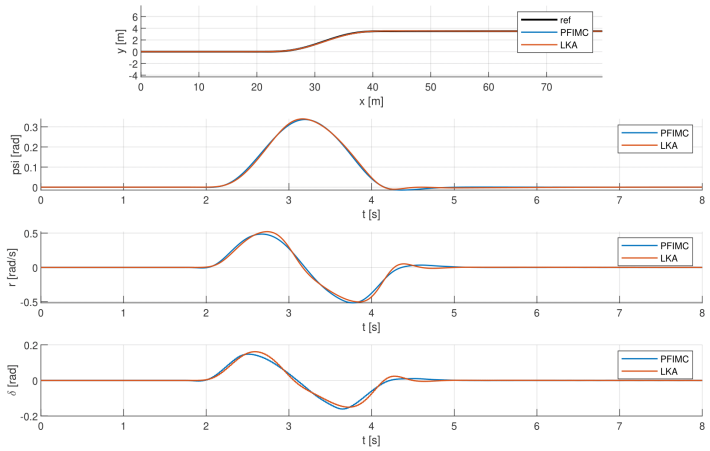

Figure 10
Path comparison of LKA and PF-IMC under low mu

Conclusions

In this paper, a novel approach is presented for lateral vehicle control. This parameter-free model predictive control transforms the classical model predictive control problem into a series of optimization problems, where the cost functions are hierarchized. Therefore, each cost function consists of a physical phenomenon. Due to this formulation, there is no need for tuning parameters in the system. This method is combined with the IMC structure for handling the dynamics and the nonlinearities of the system and implementing robustness against parameter changes and external disturbances.

The proposed method is compared with the well-known and widespread MPC methods. Three different comparisons were performed to see the advantages of the proposed method in detail, each focusing on a specified field of the proposed system. In Table 4, the main differences between the presented structure and the classical structure are gathered.

Table 4
Comparison table

LKA system	**PF-IMC**
Weights tuned intuitively	Hierarchical solver without weights
Different tuning and evaluation goals	Goal based on a physical phenomenon
Scenario-specific parameters	Consistent response overall scenarios
Single optimization task	Three optimization tasks
Linearized dynamics	Linear kinematics with nonlinear dynamics

From the simulation results and the aspects described in Table 4, it can be seen that the proposed algorithm has an outstanding contribution considering the

predictive controllers. This method can be generalized so that the reduction of tuning parameters can be reached in other control problems where MPC is used.

References

[1] Nilsson J, Falcone P, Ali M, et al. Receding horizon maneuver generation for automated highway driving. Control Engineering Practice. 2015 08;41

[2] Max G, Lantos B. Time optimal control of four-in-wheel-motors driven electric cars. Periodica Polytechnica Electrical Engineering and Computer Science, 2014;58(4):149-159

[3] Chen T, Babanin A, Muhammad A, et al. Modified evolved bat algorithm of fuzzy optimal control for complex nonlinear systems. Rom J Inf Sci Technol. 2020;23:T28-T40

[4] Precup RE, Preitl S, Petriu EM, et al. Generic two-degree-of-freedom linear and fuzzy controllers for integral processes. Journal of the Franklin Institute. 2009; 346(10):980-1003

[5] Redjimi H, Tar JK. Multiple components fixed point iteration in the adaptive control of single variable 2nd order systems. Acta Polytechnica Hungarica. 2021;18(9):69-84

[6] Roman RC, Precup RE, Hedrea EL, et al. Iterative feedback tuning algorithm for tower crane systems. Procedia Computer Science. 2022;199:157-165

[7] Tan Y, Chang J, Tan H, et al. Integral backstepping control and experimental implementation for motion system. Proceedings of the 2000. IEEE International Conference on Control Applications;02;2000, pp. 367-372

[8] Arogeti SA, Berman N. Path following of autonomous vehicles in the presence of sliding effects. IEEE Transactions on Vehicular Technology. 2012;61(4):1481-1492

[9] Galceran E, Eustice RM, Olson E. Toward integrated motion planning and control using potential fields and torque-based steering actuation for autonomous driving. In: 2015 IEEE Intelligent Vehicles Symposium (IV); 2015, pp. 304-309

[10] Menhour L, D'Andr´ea-Novel B, Fliess M, et al. Coupled nonlinear vehicle control: Flatness-based setting with algebraic estimation techniques. Control Engineering Practice. 2014;22:135-146

[11] Bodo Z, Lantos B. High level kinematic and low level nonlinear dynamic control of unmanned ground vehicles. Acta Polytechnica Hungarica. 2019 may;16(1)

[12] De Luca A, Oriolo G, Samson C. Feedback control of a nonholonomic car-like robot. Berlin, Heidelberg: Springer Berlin Heidelberg; 1998, Chapter 4; pp. 171-253

[13] Talvala K, Kritayakirana K, Gerdes J. Pushing the limits: From lanekeeping to autonomous racing. Annual Reviews in Control. 2011 04;35:137-148

[14] Li X, Sun Z, Liu D, et al. Combining local trajectory planning and tracking control for autonomous ground vehicles navigating along a reference path. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC); 2014, pp. 725-731

[15] Babqi AJ, Alamri B. A comprehensive comparison between finite control set model predictive control and classical proportional-integral control for grid-tied power electronics devices. Acta Polytechnica Hungarica. 2021; 18(7):67-87

[16] Jalali M, Khajepour A, Chen SK, et al. Integrated stability and traction control for electric vehicles using model predictive control. Control Engineering Practice. 2016 09; 54:256-266

[17] Lin F, Chen Y, Zhao Y, et al. Path tracking of autonomous vehicle based on adaptive model predictive control. International Journal of Advanced Robotic Systems. 2019 09; 16:1729881419880089

[18] Huang C, Li B, Kishida M. Model predictive approach to integrated path planning and tracking for autonomous vehicles. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC); 2019, pp. 1448-1453

[19] Kovacs A, Vajk I. Integrated path planning and lateral-longitudinal control for autonomous electric vehicles. In: 2021 AEIT International Conference on Electrical and Electronic Technologies for Automotive; 2021, pp. 1-6

[20] Kovacs A, Vajk I. Integrated lateral and longitudinal control with optimization-based allocation strategy for autonomous electric vehicles. Journal of Advanced Transportation. 2021 11;2021:1-18

[21] Kanatnikov A, Liu W, Tkachev S. Path coordinates in a 3d path following problem. Mathematical Models and Computer Simulations. 2018 05;10:265-275

[22] Kabzan J, Hewing L, Liniger A, et al. Learning-based model predictive control for autonomous racing. IEEE Robotics and Automation Letters. 2019; 4(4):3363-3370

[23] Pacejka H. Tire and vehicle dynamics. Elsevier; 2005

[24] Rakovic SV, Levine W. Handbook of model predictive control. Springer; 2018

[25] Preitl Z, Precup RE, Tar JK, et al. Use of multi-parametric quadratic programming in fuzzy control systems. Acta Polytechnica Hungarica. 2006; 3(3):29-43

[26] Alizadeh F, Goldfarb D. Second-order cone programming. Mathematical Programming. 2003 Jan;95:3-51

[27] MATLAB. Lane keeping assist system [https://www.mathworks.com/help/mpc/ref/lanekeepingassistsystem.html]; 2022, Accessed: 2022-02-10

[28] Ni J, Hu J, Xiang C. Envelope control for four-wheel independently actuated autonomous ground vehicle through afs/dyc integrated control. IEEE Transactions on Vehicular Technology. 2017;66(11):9712-9726