10.24425/acs.2026.1544

Archives of Control Sciences
Volume 36(LXXII), 2026
No. 2, *pages* 447–467

Optimization and numerical approaches for autonomous vehicles path following problem

Michał BRZOZOWSKI

This study presents and evaluates three steering angle selection methods formulated for path following tasks. The methods determine the steering angle input through numerical integration of the vehicle equations of motion over a finite prediction horizon, divided into small time subintervals. The first approach introduces classical Nelder-Mead optimization algorithm when the steering angle is defined by minimizing a cost functional which is trajectory-tracking error. In the second and third cases the minimization of the cost functional is obtained by applying the following numerical methods: Newton iteration and bisection. The methods are applicable to vehicle dynamics models of varying complexity. Presented methods have been evaluated using planar (3 degrees of freedom – DoF) and spatial (10-DoF) vehicle models. Simulation results demonstrate that proposed approaches achieve trajectory-tracking accuracy comparable to those reported in literature. The bisection-based method provides balance between accuracy and computational efficiency, enabling reliable steering angle computation for trajectories with varying curvature and non-uniform speed profiles. A key advantage of the proposed numerical methods is their ability to incorporate nonlinear vehicle dynamics while significantly reducing the computational effort compared to conventional optimization-based approaches

Key words: autonomous vehicles, optimization, numerical methods, path following, vehicle dynamics

1. Introduction

Given a reference path and a prescribed speed profile, the control task typically involves determining an appropriate front-wheel steering angle sequence that allows the vehicle to follow the desired path with acceptable accuracy. Comparative studies about steering control can be found in [1] and [2] which provide useful performance benchmarks. In general, path-following control methods can be classified into three main categories: geometry-based algorithms, model-based (vehicle dynamics) controllers, and data-driven or learning-based approaches.

Copyright © 2026. The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (CC BY-NC-ND 4.0 https://creativecommons.org/licenses/by-nc-nd/4.0/), which permits use, distribution, and reproduction in any medium, provided that the article is properly cited, the use is non-commercial, and no modifications or adaptations are made

M. Brzozowski (e-mail: mbrzozowski@ubb.edu.pl) is with the University of Bielsko-Biala, Poland.
Received 18 January 2026. Revised 19 March 2026. Revised 26 May 2026. Accepted 19 May 2026.

Simple steering control algorithms may be based on Proportional–Integral–Derivative controllers (PID) [3] or basic geometric principles, such as wheel base or corner radius. Among geometric approaches, the Pure Pursuit (PP) algorithm is one of the most widely adopted methods. Originally introduced in the early 1990s [4], it continues to be used in modern autonomous vehicle systems [5,6]. The method does not take into account vehicle dynamics, which limits its performance at higher speeds or in low-adhesion conditions. Another well-known geometric method is the Stanley control (SC) method, described in detail in [7], where particular attention is paid to the influence of path curvature estimation on control performance. Geometric control algorithms exhibit limitations in demanding driving conditions, including higher speeds, aggressive maneuvers or varying road conditions.

To overcome these limitations, model-based approaches introduce: Linear Quadratic Regulator (LQR) and Nonlinear Quadratic Regulator (NLQR) methods have been widely studied [8–10]. These techniques explicitly incorporate vehicle dynamics models but often require model linearization. A widely investigated research area involves Model Predictive Control (MPC) and Nonlinear Model Predictive Control (NMPC) methods [11–13], which compute control inputs over a receding prediction horizon using numerical optimization. While MPC-based approaches offer improved handling of system constraints and dynamic effects, their computational cost can limit real-time applicability, especially when higher-fidelity vehicle models are employed [14, 15].

Learning-based control strategies constitute another important class of path-tracking methods. These approaches do not rely on explicit vehicle models and are particularly attractive when accurate modeling is difficult but large datasets are available. In many cases, model-based formulations are used as a foundation for training learning-based controllers. Examples include reinforcement learning–based steering control [16], which can achieve good performance in simulation as well as fuzzy logic–based methods [17]. More recently, model-free learning-based control has gained attention as a promising approach for handling nonlinear and under-actuated systems with complex and sometimes chaotic behavior. For instance, Tutsoy and Barkana [18] introduced an adaptive control framework. In [19], a reinforcement learning-based approach is proposed in which the tire force model and associated nonlinearities are replaced by a data-driven approximation. In [20], the authors indicate that reinforcement learning is not without limitations. It has been shown that even in relatively simple minimum-time stabilization problems, learning-based approaches may suffer from convergence issues and ill-conditioned value function approximation.

A key drawback of current methods lies in the absence of a framework that combines system nonlinearity with low computational cost and scalar control

parameterization. This study proposes a simplified predictive control framework inspired by model predictive control (MPC), in which the steering input over the prediction horizon is parameterized by a single scalar variable. Importantly, the proposed formulation enables the use of a full nonlinear vehicle dynamics model in the prediction stage, without requiring linearization or model simplification.

The path approximation is based on B5-type Finite Element Method (FEM) functions. Steering angles are computed using three approaches: optimization, Newton based iterations and bisection-based method. The performance of the methods is evaluated using vehicle models of different complexity, with particular emphasis on computational efficiency.

The main contributions of this paper are:

1. Development of a simplified predictive steering control framework inspired by MPC and based on direct numerical integration of nonlinear vehicle dynamics.
2. Reformulation of the control problem into a scalar optimization problem.
3. Comparative analysis of three numerical methods (optimization, Newton, bisection).
4. Validation of the proposed approaches using vehicle models of varying complexity (3-DoF and 10-DoF).

2. Vehicle dynamic models

The 3 DoF (bicycle) model is the most frequently employed for motion planning and autonomous vehicle control tasks. The model remains a widely used compromise between simplicity and accuracy. It has been applied in combination with MPC for path tracking [8, 21], as well as in other control tasks such as parking maneuvers [22] and highway driving [23]. However, despite its versatility, the model neglects several important dynamic effects, including load transfer and detailed tire behavior, which limit its accuracy in aggressive driving scenarios.

More complex models are less commonly used due to their increased computational cost. A 5-DoF planar model extending the bicycle formulation is presented in [24], while 7-DoF models incorporating individual wheel dynamics are discussed in [25]. The application of a 9-degree-of-freedom model in an obstacle avoidance maneuver was presented in [26]. Another example of an extended vehicle model (10 degrees of freedom), additionally including a trailer (8 degrees of freedom), can be found in [27]. While these models provide improved fidelity, they significantly increase the computational burden, particularly when used within optimization-based control frameworks.

In this context, the study considers vehicle models of different complexity in order to evaluate the proposed control methods under varying levels of model fidelity and computational demand. Two models are used: a 3-DoF planar model and a 10-DoF spatial model. The latter represents the vehicle as a rigid body with six degrees of freedom describing the chassis, complemented by four rotating wheels (Fig. 1).

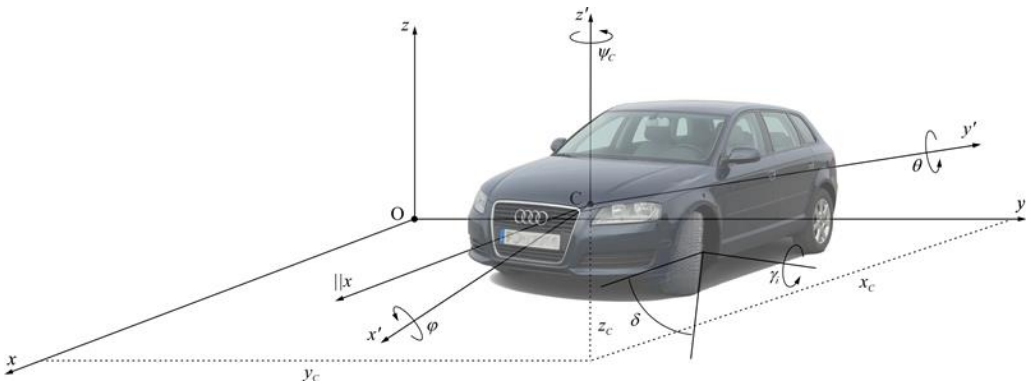

Figure 1: The vehicle model with 10 DoF

To derive the equations of motion for the chassis along with the attached concentrated masses (including wheels, suspensions, and other masses treated as lumped), a formalism based on the Newton-Euler equations was applied [28]. The vector of generalized coordinates takes the form:

$$\mathbf{q} = \begin{bmatrix} x_c & y_c & z_c & \psi_c & \theta & \varphi & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \end{bmatrix}^{\mathrm{T}}, \tag{1}$$

where:
$\mathbf{q}$ – vector of generalized coordinates,
x_c, y_c, z_c – coordinates of the vehicle's center of mass,
ψ_c, θ, φ – Euler angles defining the orientation of the local body frame axes,
$\gamma_{1,2,3,4}$ – wheel rotation angles.

The lateral and longitudinal tire forces are calculated using the modified brush model [29], as detailed by Rajamani [30].

The second model used is the aforementioned three degrees of freedom bicycle planar model. In this case, the generalized coordinates vector takes the form:

$$\mathbf{q} = \begin{bmatrix} x_c & y_c & \psi_c \end{bmatrix}^{\mathrm{T}}. \tag{2}$$

In both cases equations of the motion of a vehicle are written in the following form

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{f}\big(\mathbf{q},\ \dot{\mathbf{q}},\ \mathbf{u}(t)\big), \tag{3}$$

where:

$$\mathbf{u}(t) = \begin{bmatrix} \delta(t) \\ M_n(t) \end{bmatrix},$$

$\mathbf{M}(\mathbf{q})$ – inertia matrix, $\mathbf{f}$ – vector of generalized forces, $\mathbf{q}$, $\dot{\mathbf{q}}$ – vectors of general coordinates and velocities, $\delta(t)$ denotes the steering angle of the front wheels, $M_n(t)$ represents the driving torque applied to the front axis of vehicle.

The validation and verification of both models and their computer implementations are presented in the work [31]. It includes a comparison with the CarSim software as well as with experiments conducted by [32]. Acceptable agreement was obtained.

3. Path approximation

In this paper, the path is given as a sequence of points $P_0 \div P_m$ (Fig. 2).

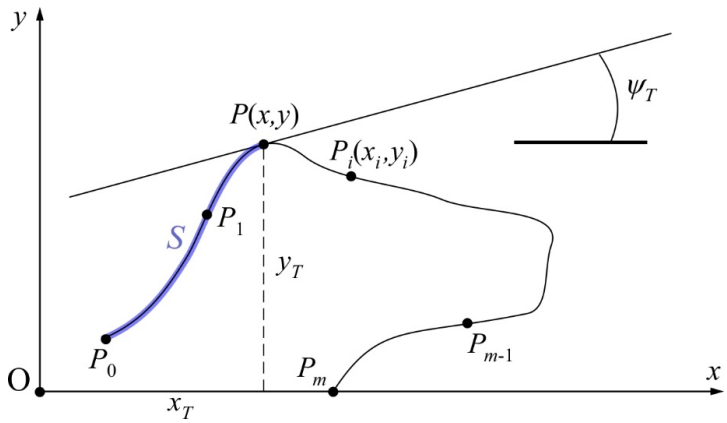

Figure 2: The vehicle model with 10 DoF

The B_5 finite element method functions were used to approximate a discretely defined path, following the approach presented in [2]. In order to approximate the trajectory points with a continuous path, it is first necessary to compute the distance traveled by the vehicle:

$$s_i = \sum_{j=1}^{i} \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2}\,. \tag{4}$$

In this way, the coordinates x and y can be treated as functions of the variable s. This allows for the formulation of the least squares method and define the

functions:

$$x(s) = \sum_{i=-1}^{n+1} a_i^x \varphi_i(s), \tag{5}$$

$$y(s) = \sum_{i=-1}^{n+1} a_i^y \varphi_i(s). \tag{6}$$

4. Proposed steering methods

The problem considered in this study is to determine the steering angle that minimizes trajectory tracking error over a finite prediction horizon, subject to vehicle dynamics and constraints.

The presented approaches are based on the underlying assumptions:

1. The vehicle dynamic models with 3 and 10 DoF are assumed to adequately represent the vehicle behavior.
2. The trajectory which the vehicle is to follow can be approximated by B_5 functions.
3. The steering angle can be determined numerically through simulations of the vehicle's behavior in some subintervals of time horizon.

In contrast to many MPC-based approaches, which rely on model linearization, the proposed methods operate directly on the nonlinear vehicle dynamics model. The continuous-time vehicle model is used directly and integrated numerically.

When desired velocity $v(t)$ of the car is given (known), the distance traveled by the vehicle can be calculated using the formula:

$$s(t) = \int_{t_0}^{t} v(\tau)\,\mathrm{d}\tau, \tag{7}$$

to ensure the realization of the velocity $v = \dfrac{ds}{dt}$. After conducting a series of test calculations, a procedure was adopted in which the driving torque from Eq. (3) was approximated as follows:

$$M_n(t) = k \times (v_0 - v), \tag{8}$$

where: v – the actual vehicle velocity, v_0 – the desired vehicle velocity, k – constant.

With this vehicle speed regulator, the control vector $\mathbf{u}$ reduces to a single component; thus, $\mathbf{u}(t) = \delta(t)$.

In this study, the objective is to determine the front-wheel steering angle $\delta(t)$ that ensures tracking of the desired trajectory. The time interval considered for the analysis is $\langle 0,\ \mathrm{T}\rangle$. It is assumed that the vehicle velocity profile $v_0(t)$ is known. As a result of approximating the trajectory using B_5 functions $x_T = x_T(s)$, $y_T = y_T(s)$ and $\psi_T = \psi_T(s)$, and given $v_0(t)$, the following functions can be determined: $x_T(t)$, $y_T(t)$ and $\psi_T(t)$. In this work, the steering angle profile is not determined over the entire interval $\langle 0,\ \mathrm{T}\rangle$ at once. Instead, the interval is divided into subintervals of length $\Delta\mathrm{T}$.

$$\Delta\mathrm{T} = mh_c\,, \tag{9}$$

where: h_c – the integration step of the equations of motion, m – the multiple of the integration step. Figure 3 illustrates the division into subintervals.

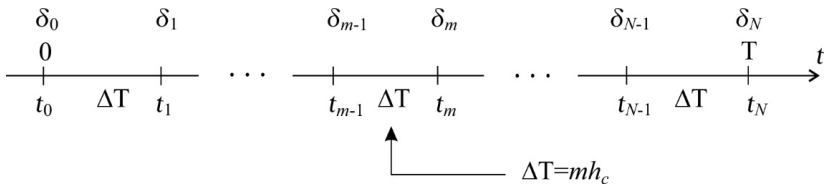

Figure 3: Division into subintervals

It is assumed that at $t = t_{m-1}$ the steering angle:

$$\delta_{m-1} = \delta(t_{m-1}), \tag{10}$$

is known as well as the initial conditions:

$$\mathbf{q}_{m-1} = \mathbf{q}(t_{m-1}), \qquad \dot{\mathbf{q}}_{m-1} = \dot{\mathbf{q}}(t_{m-1}). \tag{11}$$

The value of the steering angle in $t_m = t_{m-1} + \Delta T$ is denoted as:

$$\delta_m = \delta(t_m), \tag{12}$$

which can change linearly over the interval $\langle t_{m-1}, t_m\rangle$ according to the relation:

$$\delta(t) = \delta_{m-1} + \frac{\delta_m - \delta_{m-1}}{\Delta T}\big(t - t_{m-1}\big), \tag{13}$$

or remains constant over this interval, i.e.,

$$\delta(t) = \delta_m\,. \tag{14}$$

In this way the determination of the steering angle $\delta = \delta(t)$ over the interval $\langle t_{m-1}, t_m\rangle$ reduces to identifying a single representative value of δ_m which must satisfy the imposed constraints:

$$\delta_{\min} \leqslant \delta_m \leqslant \delta_{\max}\,, \tag{15}$$

where: $\delta_{\min} = -\min\left\{\delta_{\max},\ \delta_0+\Delta t\dot{\delta}_{\max}\right\}$, $\delta_{\max} = \min\left\{\delta_{\max},\ \delta_0+\Delta t\dot{\delta}_{\max}\right\}$, $\delta_{\max}$ – the maximum allowable steering angle of the vehicle wheels, $\dot{\delta}_{\max}$ – the maximum steering rate of the vehicle wheels.

The constraints imposed on the steering angle are enforced by limiting the admissible range of the decision variable δ_m during the optimization process.

The control problem for the m-th subinterval $\langle t_{m-1}, t_m\rangle$ of length ΔT, is formulated as a continuous-time optimal control problem [33] as follows. Find:

$$\min_{\substack{\delta_{\min}\leqslant\delta_m\leqslant\delta_{\max}\\ t_{m-1}\leqslant t\leqslant t_m}} \Omega\big(\mathbf{X}(t),\delta_m\big), \tag{16}$$

Ω represents a functional, the form of which depends on the applied control scheme, $\mathbf{X} = \begin{bmatrix}\boldsymbol{q}\\ \dot{\boldsymbol{q}}\end{bmatrix}$ denotes a vector of state variables, being the solution to the initial value problem:

$$\dot{\mathbf{X}} = \boldsymbol{f}_c\big(t,\mathbf{X}(t),\delta(t)\big), \qquad \mathbf{X}(t_{m-1}) = \mathbf{X}_{m-1}\,, \tag{17}$$

where: $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ – generalized coordinates and velocities,

$$\boldsymbol{f}_c = \begin{bmatrix}\dot{\boldsymbol{q}}\\ \boldsymbol{f}(t,\boldsymbol{q},\dot{\boldsymbol{q}},\delta(t))\end{bmatrix}, \qquad \mathbf{X}_{m-1} = \begin{bmatrix}\boldsymbol{q}(t_{m-1})\\ \dot{\boldsymbol{q}}(t_{m-1})\end{bmatrix}.$$

The equations of motion were integrated using the fourth-order Runge-Kutta method with a time step of 0.002 s. The constraints are expressed by Eq. (15).

The discrete formulation of this control problem can be expressed as follows:

$$\min_{\mathbf{X},\mathbf{u}} \sum_{m=1}^{N} \Omega\big(\mathbf{X}(t),\delta_m\big). \tag{18}$$

The control problem over the intervals $\langle t_{m-1}, t_m\rangle$ is formulated below for all three approaches considered in this study. In each case, for every value of the steering angle δ_m, the equations of motion (17) are integrated and constraints (15) are imposed. The approaches differ in the formulation of Ω and in the methods employed to determine its minimum.

4.1. Optimization method

Figure 4 illustrates the quantity ε_y defined as the lateral deviation of the vehicle's center of mass from the desired trajectory, which can be calculated as follows:

$$\varepsilon_y = -(x_T - x_c)\sin\psi_c + (y_T - y_c)\cos\psi_c\,. \tag{19}$$

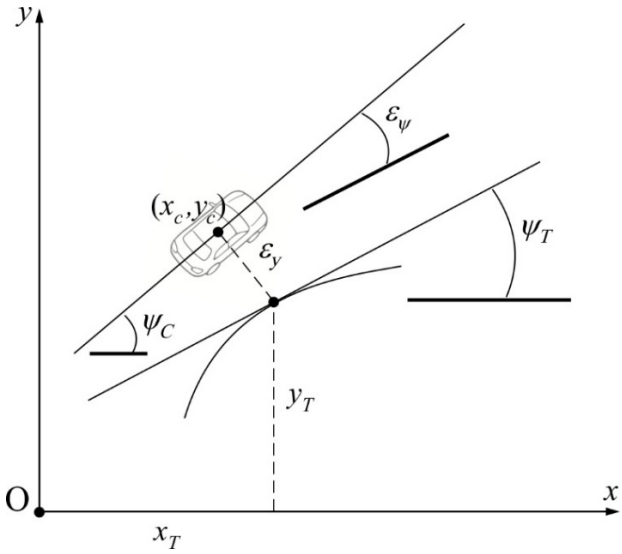

Figure 4: Path deviation

In this case, the functional Ω is formulated as follows:

$$\Omega = \left\{\varepsilon_y(t_m)\right\}^2. \tag{20}$$

In order to evaluate the functional (20) for each value of the steering angle δ_m indicated by the optimization procedure, the equations (17) must be integrated subject to the appropriate initial conditions (15).

4.2. Newton Iterative Method

In this approach, the objective is to find the minimum of the functional:

$$\Omega = \dot{\Delta}^2(\delta_m) = C_1^{x,y}\left[\left(\dot{x}_{c,m} - \dot{x}_{T,m}\right)^2 + \left(\dot{y}_{c,m} - \dot{y}_{T,m}\right)^2\right] + C_1^{\psi}\left[\dot{\psi}_{c,m} - \dot{\psi}_{T,m}\right]^2, \tag{21}$$

$x_{c,m} = x_c(t_m)$, $y_{c,m} = y_c(t_m)$, $\psi_{c,m} = \psi_c(t_m)$ are the coordinates of the vehicle's center of mass and its yaw angle for $t = t_m$, and $x_{T,m} = x_T(t_m)$, $y_{T,m} = y_T(t_m)$, $\psi_{T,m} = \psi_T(t_m)$ are the desired path coordinates and the tangent angle to the vehicle trajectory, calculated according to the path approximation algorithm at time t_m. Simultaneously, the quantities $\dot{x}_{c,m}$, $\dot{y}_{c,m}$ and $\dot{\psi}_{c,m}$ are assumed to be determined as the solution to the initial value problem (17). The constants $C_1^{x,y}$ and C_1^{ψ} were determined through computational experiments. To calculate $\dot{\Delta}^2(\delta_m)$ the equation of motion (17) must be integrated with δ_m defined by (12). The quadratic form

(21) reaches its minimum when:

$$\frac{\partial \dot{\Delta}^2(\delta_m)}{\partial \delta} = 2\left\{C_1^{x,y}\left[\left(\dot{x}_{c,m} - \dot{x}_{T,m}\right)\frac{\partial \dot{x}_{c,m}}{\partial \delta} + \left(\dot{y}_{c,m} - \dot{y}_{T,m}\right)\frac{\partial \dot{y}_{c,m}}{\partial \delta}\right] + C_1^{\psi}\left(\dot{\psi}_{c,m} - \dot{\psi}_{T,m}\right)\frac{\partial \dot{\psi}_{c,m}}{\partial \delta}\right\} = 0, \tag{22}$$

where: $\frac{\partial p}{\partial \delta}$, $p = \left[\dot{x}_{c,m}, \dot{y}_{c,m}, \dot{\psi}_{c,m}\right]$ is the derivative of the function p with respect to δ for $t = t_m$.

In this paper, these derivatives are calculated using the five-point finite difference method with equidistant spacing, assuming that:

$$\frac{\partial p}{\partial \delta}_{\delta=\delta_m} = \frac{p(\delta_m - 2\delta_h) - 8p(\delta_m - \delta_h) + 8p(\delta_m + \delta_h) - p(\delta_m + 2\delta_h)}{12\delta h}, \tag{23}$$

To calculate $\frac{\partial p}{\partial \delta}$ it is therefore necessary to integrate five times the equation of motion in the interval $\langle t_{m-1}, t_m\rangle$.

The δ_m is determined using Newton's method, assuming:

$$\delta_m^0 = \delta_{m-1}, \tag{24}$$

$$\delta_m^{(i)} = \delta_m^{(i-1)} - \frac{\chi_i(\delta_m)}{\chi_i'(\delta_m)}, \tag{25}$$

where: $\chi_i(\delta_m) = \frac{1}{2}\frac{\partial \Delta^2(\delta_m)}{\partial \delta}$, $\chi_i'(\delta_m) = \frac{\partial \chi_i(\delta_m)}{\partial \delta}$.

This iterative process was continued until the condition $|\delta_m^{(i)} - \delta_m^{(i+1)}| < \varepsilon$ or $i = i_{\max}$ was satisfied. The derivative of $\chi_i'(\delta_m)$ was also calculated using the five-point finite difference method.

4.3. Bisection method

The third proposed method is the bisection method.

In this approach, the objective is to find the minimum of the functional:

$$\Omega = \varepsilon_y(t_m). \tag{26}$$

Simultaneously, the quantities required for the evaluation of ε_y assumed to be determined as the solution to the initial value problem. Subject to the conditions (15) imposed on δ_m and assuming a constant value of δ_m over the interval $\langle t_{m-1}, t_m\rangle$.

Since, according to (26), the function ε_y takes positive values when the vehicle is on the left-hand side of the path and negative values when it is on the right-

hand side, the problem of finding $\min_{\delta_m} \varepsilon_y$ reduces to locating the root of the function $\varepsilon_y(\delta_m)$. To this end, a classical approach can be applied, consisting in first identifying an interval $\langle \delta_{m,L},\ \delta_{m,R} \rangle$ within which the function changes sign, and then successively reducing this interval.

5. Results

The performance and effectiveness of the proposed steering angle selection methods were evaluated by comparing the results. Simulations were performed on a machine equipped with Intel Core i5-11400H processor @ 2.70GHz using Delphi (pascal). A comprehensive set of simulation experiments was carried out to assess the performance of the proposed methods. Two representative scenarios were selected for detailed analysis. The first scenario involves a trajectory with high curvature, intended to evaluate the control performance under demanding maneuvering conditions. The second scenario is based on a real-world road segment, enabling validation under realistic driving conditions with varying curvature and speed profiles. The first case is a test maneuver consisting of following the track described by equation (27) at a constant speed of $v = 7.8$ m/s. A bicycle dynamics model was applied. The test track can be described by the equation:

$$\begin{cases} x = a\sin(\omega), \\ y = a\sin(\omega)\cos(\omega), \end{cases} \qquad \text{where :} \quad \omega = vt, \quad a = 50. \tag{27}$$

The simulation interval $0 \leqslant t \leqslant 38$ s was discretized using a time step of $\Delta T = 0.2$ s. Consequently, 190 integration steps were required to solve the equations of motion (17).

The vehicle parameters are presented in Table 1.

Table 1: Vehicle parameters according to Cao et al. [16]

Symbol	Description	Value
m	Vehicle mass	1188 kg
I_z	Vehicle yaw moment of inertia	2243.1 kgm^2
l_f	Front Axle-C.G. distance	1.1281 m
l_r	Rear Axle-C.G. distance	1.4719 m
C_f	Front cornering stiffness	76744 N/rad
C_r	Rear cornering stiffness	119320 N/rad

The steering angle was constrained by limits of $\delta_{\min} = -20^\circ$ and $\delta_{\max} = 20^\circ$. The determined steering angle profiles are shown in Fig. 5.

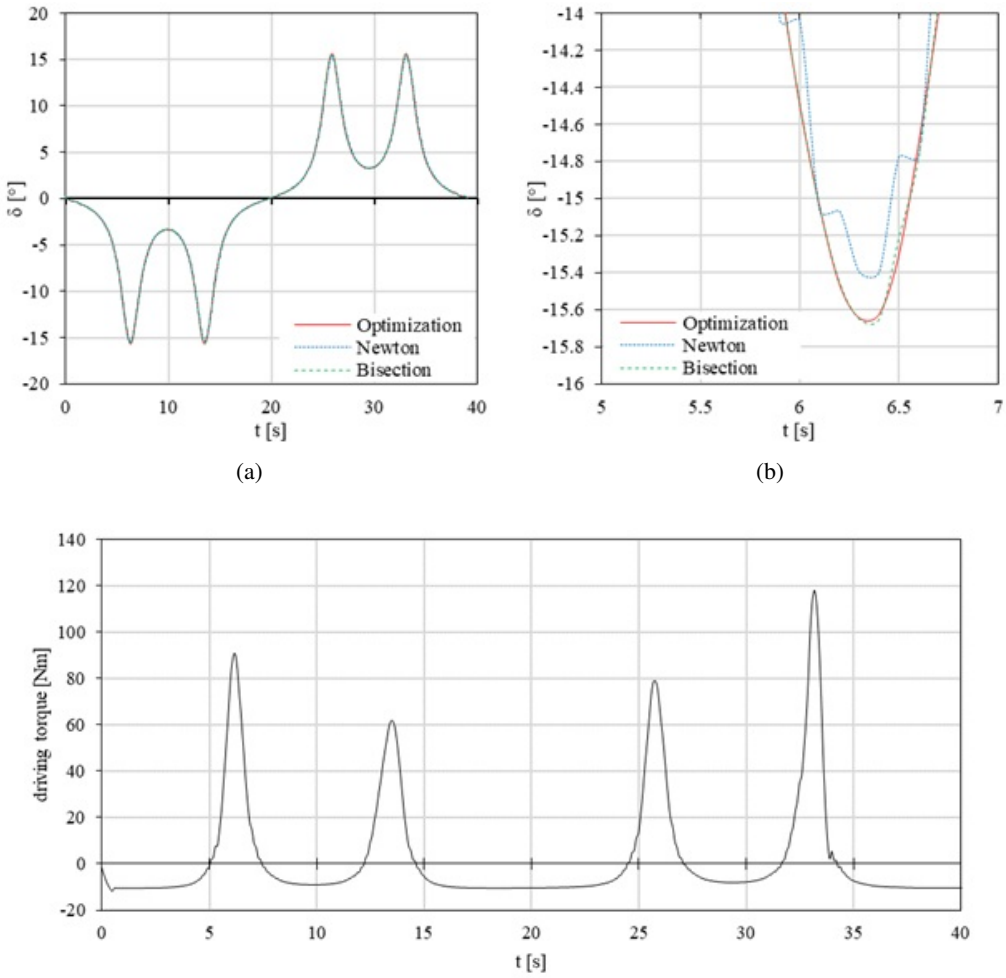

Figure 5: Control inputs: Front wheels steering angle profile (a) entire route; (b) magnified view; (c) driving torque (optimization)

Figure 5 presents the obtained steering wheel angle profiles, which are highly similar. A slight discrepancy can be observed between the Newton method and the remaining two methods. It should be emphasized that the torque trajectory presented in Fig. 5c corresponds to an idealized theoretical response. In practical real-time implementations, the control signal would require appropriate filtering and smoothing to account for actuator dynamics and reduce high-frequency oscillations. The driving torque profile remains identical for all considered methods, since it was assumed to be a known function of time. Table 2 presents the

simulation results obtained using presented methods, along with a comparison to the work of Cao et al. [16].

Table 2: Simulation results for first scenario

Steering method	Optimization	Newton	Bisection	Cao et al. [16]
RMSE [m]	0.0041	0.0653	0.0041	0.1151
Max error [m]	0.0190	0.1750	0.0190	n/d

All applied methods generate a very small mean path error (RMSE). Under the considered simulation conditions, the obtained tracking errors are smaller than those reported in [16]. The proposed methods produce similar errors. The maximum path tracking error does not exceed 0.175 m (Fig. 6) over a route length of 304 m.

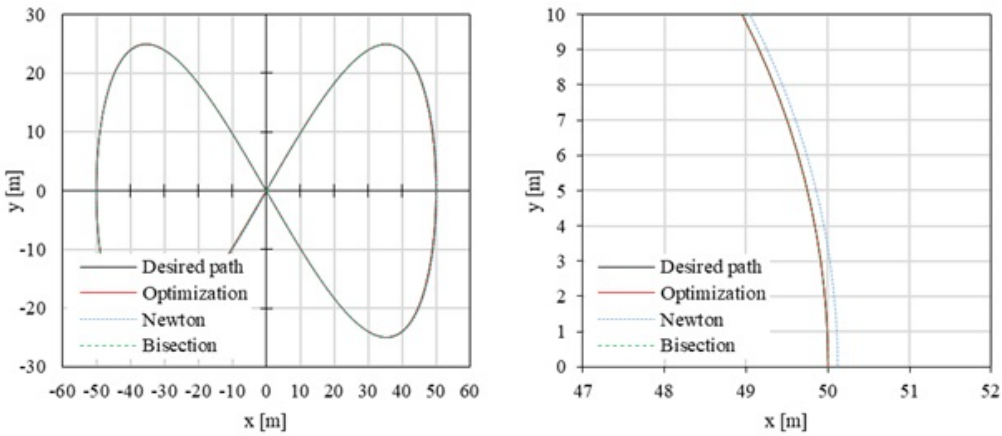

Figure 6: (a) Desired path and actual vehicle trajectory; (b) magnified view

The very low path tracking errors result not only from the applied control algorithm but also from the approximation method used. The fifth-degree function approximation enables the generation of a smoothly varying path.

To demonstrate differences in computation time and to test the control methods on a route with a variable speed profile, an additional simulation study was conducted. The maneuver, for which the results were compared, involves following a route that is a section of the bypass road around Bielsko-Biała (Poland) shown in Fig. 7a–7d.

The speed profile was based on the actual speed limits imposed along the presented route (Fig. 7b). The vehicle parameters were extended to enable simu-

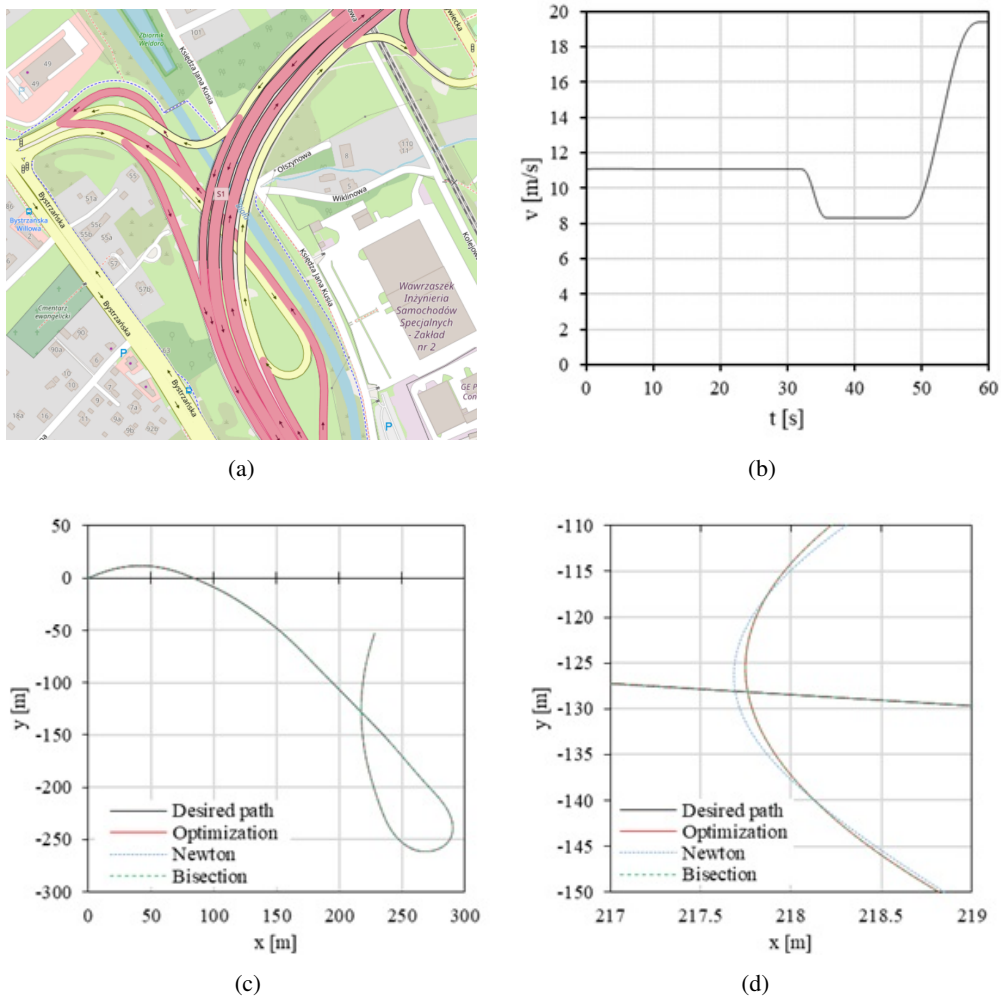

Figure 7: Bypass road of Bielsko-Biała city: (a) actual route – OpenStreetMap; (b) speed profile; (c) desired path and actual vehicle trajectory; (d) magnified view

lations using a spatial model (10 DoF). The parameters of the vehicle are provided in Table 3.

Similar to the previous case, B_5 functions were used to approximate the 672 m long trajectory. The resulting steering wheel angle trajectories are presented in Fig. 8.

Once again, the obtained steering angle profiles are very similar. The zoomed-in view reveals some minor differences, which, however, do not significantly affect the resulting path tracking error (Table 4). An advantage of the Newton method

Table 3: Vehicle parameters according to Chebly et al. [34]

Symbol	Description	Value
l_f	Front Axle-C.G. distance	1.195 [m]
l_r	Rear Axle-C.G. distance	1.513 [m]
b_f, b_r	Track width of front and rear wheels	1.4 [m]
$I_{x'}$	Mass moment of inertia about the x' axis	1100 [kg·m^2]
$I_{y'}$	Mass moment of inertia about the y' axis	2300 [kg·m^2]
$I_{z'}$	Mass moment of inertia about the z' axis	3300 [kg·m^2]
h	Height of the center of gravity	0.5 [m]
m	Vehicle mass	1719 [kg]
r_n	Wheel radius	0.32 [m]
I_w	Mass moment of inertia of the wheel about its rotation axis	1.02 [kg·m^2]
m_w	Wheel mass	12.2 [kg]
C_{af}, C_{ar}	Tire lateral slip stiffness coefficients for front and rear wheels	85275 F 68922 R [N/rad]
C_{bf}, C_{br}	Tire longitudinal stiffness coefficients of front and rear wheels	8500 F 8500 R [N/rad]
A	Vehicle frontal area	2.31 [m^2]
c_x	Aerodynamic drag coefficient	0.314
c^i	Equivalent radial stiffness coefficient of the wheel	$3.3 \cdot 10^4$ [N/m]

is its ability to generate a smooth steering angle profile, due to the assumption that the angle changes linearly within each interval.

Table 4: Simulation results for second scenario

Steering method	Optimization		Newton		Bisection	
Vehicle model	3 DoF	10 DoF	3 DoF	10 DoF	3 DoF	10 DoF
RMSE [m]	0.052	0.052	0.065	0.065	0.052	0.052
Max error [m]	0.169	0.169	0.140	0.140	0.169	0.169
Computation time [s]	5.390	90.00	4.780	86.380	2.360	32.16

The bisection method is the fastest, as it does not require performing such complex calculations. Simulation times for the 3-DoF dynamic model are approximately twenty times shorter than those for the 10-DoF model. In the presented scenario the bisection method proposed in this study produces results with ac-

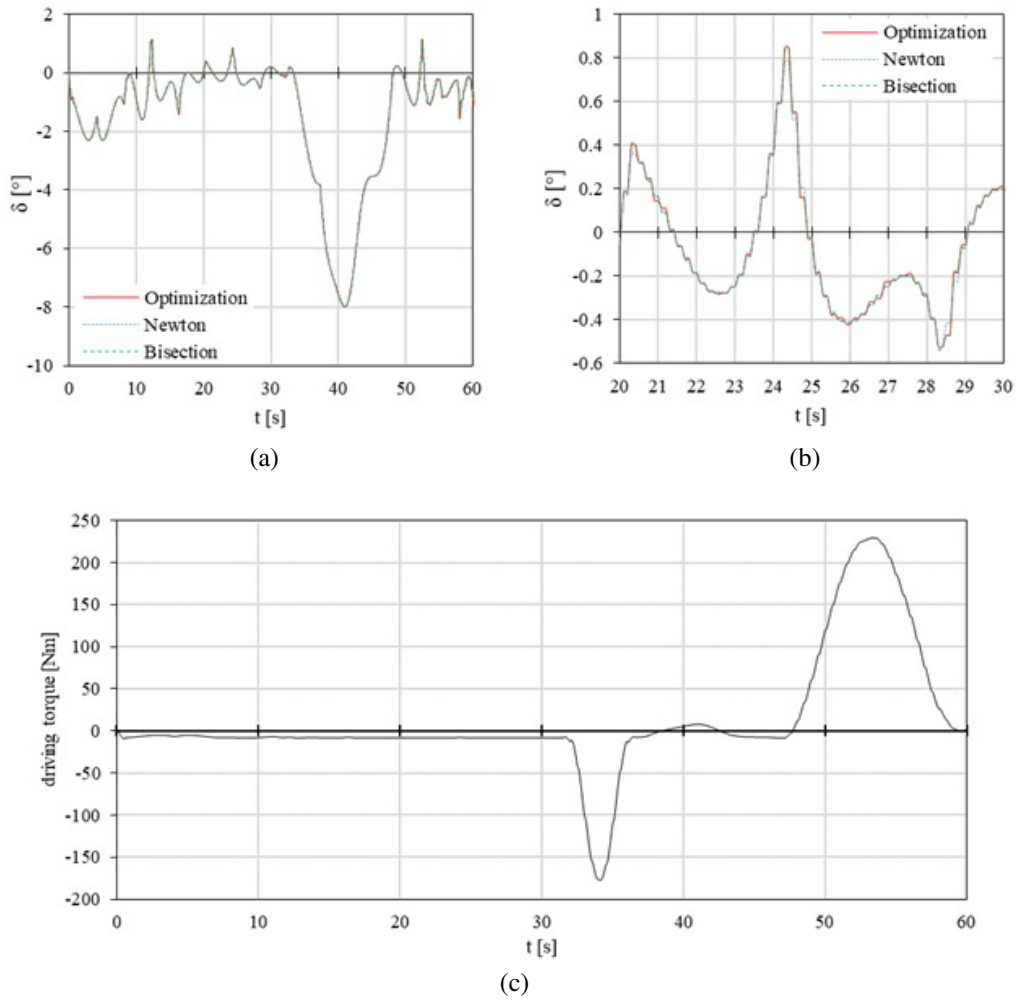

Figure 8: Control inputs: Front wheels steering angle profile: (a) entire route; (b) magnified view; (c) driving torque (optimization)

curacy similar to the optimization method, while reducing computation time by more than 50%.

For the purpose of evaluating the stability of the proposed control methods, the error profile is presented (Fig. 9).

All methods ensure bounded and stable tracking performance without noticeable divergence or oscillatory instability. The trajectory error remains largely similar across all methods. Minor fluctuations are observed at approximately $s \sim 400$ m, corresponding to cornering.

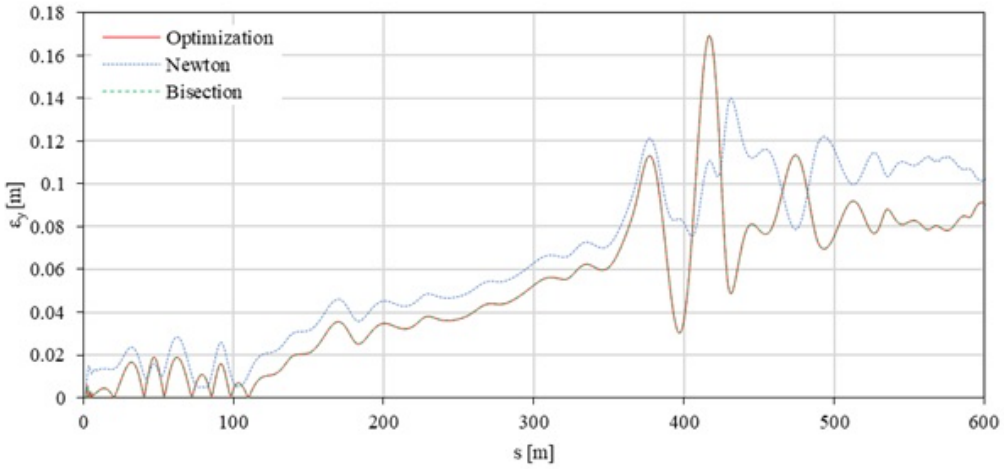

Figure 9: Tracking error ε_y as a function of the traveled distance s for the considered methods

To evaluate the robustness of the proposed methods, a sensitivity analysis regarding model uncertainties was performed. With a 10% mismatch in vehicle mass, presented methods maintained stable tracking with an RMSE increase of less than 6%.

6. Discussion

The proposed optimization-based method is robust and effective; however, its primary limitation is low numerical efficiency when used with high-fidelity models.

The Newton method demonstrates computational performance comparable to the bisection method while offering increased flexibility through the tuning of weighting coefficients $C_1^{x,y}$ and C_1^{ψ}. This flexibility allows the method to be adapted to different tracking accuracy and smoothness requirements. The method can be applied to vehicle dynamics models of varying complexity, subject to the resulting computational burden and available computational resources. Its main limitation is a moderate computational efficiency when higher-fidelity models are employed. A drawback of this method is that the convergence of the Newton-based approach depends on the initial guess and may deteriorate for highly nonlinear trajectories.

The bisection method exhibits very high computational efficiency due to its algorithmic simplicity. Despite relying on a scalar search procedure, the method achieves low path-tracking errors for the considered scenarios. However, the

resulting steering angle profiles may require additional smoothing in practical implementations to ensure actuator-friendly control signals.

The applied path approximation technique based on fifth-degree polynomial functions enables the generation of smooth reference trajectories, which positively influences the performance of the steering control algorithms.

The considered uncertainties are not explicitly modeled in the control formulation. Instead, their effect is indirectly evaluated through sensitivity analysis, in which selected model parameters (e.g., vehicle mass) are varied. The results indicate that the proposed method maintains stable tracking performance under moderate parameter variations.

A limitation of the proposed approach is the reduction of the control input to a single parameter, which may restrict performance in highly dynamic scenarios requiring rapid control variation.

7. Conclusions

The effectiveness of the proposed approach stems from several factors. First, reducing the control problem to a single scalar parameter simplifies the optimization task, which improves numerical stability and lowers computational effort. Second, the short prediction horizon helps limit the accumulation of modeling errors while allowing the control input to be updated frequently. In this way, the method can compensate for moderate inaccuracies without relying on an explicit disturbance model. Finally, the use of smooth trajectory approximation provides well-behaved reference signals, supporting stable and accurate tracking.

Based on the presented results, the following conclusions can be drawn:

- Path approximation using fifth-degree polynomial functions enables the generation of smooth trajectories within acceptable computational time.
- The imposed constraints on steering angle and rate ensure physical feasibility of the control inputs.
- The proposed Newton and bisection methods achieve trajectory-tracking performance comparable to the selected classical optimization approach under the considered assumptions and test scenarios. The agreement with previously published simulation results reported by Cao et al. [16] supports the validity of the obtained results.
- The Newton method offers a favorable compromise between flexibility and computational efficiency, enabling tuning for specific performance requirements and generating continuous steering angle trajectories.
- The bisection method significantly reduces computational effort compared to classical optimization techniques, making it particularly suitable for ap-

plications with strict real-time constraints, provided that additional smoothing of the steering signal is applied if necessary.

- The comparison between vehicle models with 3 and 10 degrees of freedom indicates that, for relatively flat trajectories with moderate lateral accelerations, lower-order vehicle models can provide sufficient accuracy while offering substantial computational advantages.

Future research will focus on extending the proposed methods to more demanding driving scenarios and implementing the algorithms on a real test vehicle.

References

[1] J. Liu, Z. Yang, Z. Huang, W. Li, S. Dang and H. Li: Simulation performance evaluation of pure pursuit, Stanley, LQR, MPC controller for autonomous vehicles. *IEEE International Conference on Real-time Computing and Robotics (RCAR)*, (2021), 1444–1449. DOI: 10.1109/RCAR52367.2021.9517448

[2] M. Diachuk and S.M. Easa: Motion planning for autonomous vehicles based on sequential optimization. *Vehicles*, 4(2), (2022) 344–374. DOI: 10.3390/vehicles4020021

[3] R. Marino, S. Scalzi and M. Netto: Nested PID steering control for lane keeping in autonomous vehicles. *Control Engineering Practice*, **19**(12), (2011), 1459–1467. DOI: 10.1016/j.conengprac.2011.08.005

[4] R.C. Coulter: Implementation of the pure pursuit path tracking algorithm. *Technical Report CMU-RI-TR-92-01*, Robotics Institute, Carnegie Mellon University, 1992.

[5] C.G. Serna, A. Lombard, Y. Ruichek and A. Abbas-Turki: GPS-based curve estimation for an adaptive pure pursuit algorithm. *Mexican International Conference on Artificial Intelligence*, (2017), 497–511. DOI: 10.1007/978-3-319-62434-1_40

[6] L. An, X. Huang, P. Yang and Z. Liu: Adaptive Bézier curve-based path following control for autonomous driving robots. *Robotics and Autonomous Systems*, 189, (2025), 104969. DOI: 10.1016/j.robot.2025.104969

[7] Y. Yang, Y. Li, X. Wen, G. Zhang, Q. Ma, S. Cheng, J. Qi, L. Xu and L. Chen: An optimal goal point determination algorithm for automatic navigation of agricultural machinery. *Computers and Electronics in Agriculture*, **194**, (2022), 106760. DOI: 10.1016/j.compag.2022.106760

[8] X. Li, Z. Sun, D. Cao, D. Liu and H. He: Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles. *Mechanical Systems and Signal Processing*, **87**, (2017), 118–137. DOI: 10.1016/j.ymssp.2015.10.021

[9] K. Lee, S. Jeon, H. Kim and D. Kum: Optimal path tracking control of autonomous vehicle: Adaptive full-state linear quadratic gaussian control. *IEEE Access*, **7**, (2019), 109120–109133. DOI: 10.1109/ACCESS.2019.2933895

[10] S. Xu and H. Peng: Design, analysis, and experiments of preview path tracking control for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, **21**(1), (2020), 48–58. DOI: 10.1109/TITS.2019.2892926

[11] P. Falcone, F. Borrelli, J. Asgari, H.E. Tseng and D. Hrovat: Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, **15**(3), (2007), 566–580. DOI: 10.1109/TCST.2006.883268

[12] J.B. Rawlings and D.Q. Mayne: *Model Predictive Control, Theory and Design*. Madison, WI: Nob Hill Publishing, 2009.

[13] R. Nebeluk and P. Marusak: Efficient MPC algorithms with variable trajectories of parameters weighting predicted control errors. *Archives of Control Sciences*, **30**(2), (2020), 325–363. DOI: 10.24425/acs.2020.133502

[14] W. Zhao, H. Wei, Q. Ai, N. Zheng, C. Lin and Y. Zhang: Real-time model predictive control of path-following for autonomous vehicles towards model mismatch and uncertainty. *Control Engineering Practice*, **153**, (2024), 106126. DOI: 10.1016/j.conengprac.2024.106126

[15] G. Zhu and W. Hong: Autonomous vehicle motion control considering path preview with adaptive tire cornering stiffness under high-speed conditions. *World Electric Vehicle Journal*, **15**(12), (2024), 580. DOI: 10.3390/wevj15120580

[16] Y. Cao, K. Ni, Jiang X., T. Kuroiwa, H. Zhang, T. Kawaguchi, S. Hashimoto and W. Jiang: Path following for autonomous ground vehicle using DDPG algorithm: A reinforcement learning approach. *Applied Sciences*, **13**(11), (2023), 6847. DOI: 10.3390/app13116847

[17] H. Elsayed, B.A. Abdullah and G. Aly: Fuzzy logic based collision avoidance system for autonomous navigation vehicle. *International Conference on Computer Engineering and Systems (ICCES)*, (2018), 469–474. DOI: 10.1109/ICCES.2018.8639396

[18] O. Tutsoy and D.E. Barkan: Model-free adaptive control of the under-actuated robot manipulator with chaotic dynamics. *ISA Transactions*, **118**, (2021), 106–115. DOI: 10.1016/j.isatra.2021.02.006

[19] H. Wei, W. Zhao, Q. Ai, T. Huang and Y. Zhang: Deep reinforcement learning based active safety control for distributed drive electric vehicles. *IET Intelligent Transport Systems*, **16**(6), (2022), 813–824. DOI: 10.1049/itr2.12176

[20] O. Tutsoy and M. Brown: Reinforcement learning analysis for a minimum time balance problem. *Transactions of the Institute of Measurement and Control*, **38**(10), (2016), 1186–1200. DOI: 10.1177/0142331215581638

[21] C. Hu, Y. Qin, H. Cao, X. Song, J. Jiang, J. Rath and C. Wei: Lane keeping of autonomous vehicles based on differential steering with adaptive multivariable super-twisting control. *Mechanical Systems and Signal Processing*, **125**, (2019), 330–346. DOI: 10.1016/j.ymssp.2018.09.011

[22] M. Jošević, A. Katriniok, A. Riek and D. Abel: Disturbance estimation for longitudinal vehicle dynamics control at low speeds. *IFAC-PapersOnLine*, **50**(1), (2017), 987–993. DOI: 10.1016/j.ifacol.2017.08.204

[23] S. Zhou, Y. Wang, M. Zheng and I. Tomizuka: A hierarchical planning and control framework for structured highway driving. *IFAC-PapersOnLine*, **50**(1), (2017), 9101–9107. DOI: 10.1016/j.ifacol.2017.08.1705

[24] J. Dallas, K. Jain, Z. Dong, L. Sapronov, M. Cole, P. Jayakumar and I. Ersal: Online terrain estimation for autonomous vehicles on deformable terrains. *Journal of Terramechanics*, **91**, (2020), 11–22. DOI: 10.1016/j.jterra.2020.03.001

[25] F. Zhang, J. Gonzales, S.E. Li, F. Borrelli and K. Li: Drift control for cornering maneuver of autonomous vehicles. *Mechatronics*, **54**, (2018), 167–174. DOI: 10.1016/j.mechatronics.2018.05.009

[26] P. Polack: *Consistency and stability of hierarchical planning and control systems for autonomous driving*. PhD thesis, Université Paris Sciences et Lettres, 2018.

[27] L. Prochowski, M. Ziubiński, P. Szwajkowski, M. Gidlewski, T. Pusty and T.L. Stańczyk: Impact of control system model parameters on the obstacle avoidance by an autonomous car-trailer unit: research results. *Energies*, **14**, (2021), 1–31. DOI: 10.3390/en14102958

[28] W. Blajer: *Metody dynamiki układów wieloczłonowych*. Radom: Wydawnictwo Politechniki Radomskiej, 1998.

[29] H.B. Pacejka and R.S. Sharp: Shear force development by pneumatic tyres in steady state conditions: a review of modeling aspects. *Vehicle System Dynamics*, **20**(3–4), (1991), 121–176. DOI: 10.1080/00423119108968983

[30] R. Rajamani: *Vehicle Dynamics and Control*. New York: Springer, (012. DOI: 10.1007/978-1-4614-1433-9

[31] M. Brzozowski and Ł. Drąg: Application of dynamic optimization for autonomous vehicle motion control. *Transport Problems*, **18**(2), (2023), 209–222. DOI: 10.20858/tp.2023.18.2.18

[32] M. Cibooglu, U. Karapinar and M.T. Söylemez: Hybrid controller approach for an autonomous ground vehicle path tracking problem. *Mediterranean Conference on Control and Automation (MED)*, (2017), 583–588. DOI: 10.1109/MED.2017.7984180

[33] J.B. Rawlings, D.Q. Mayne and M. Diehl: *Model Predictive Control: Theory, Computation, and Design*. Madison: Nob Hill Publishing, 2017.

[34] A. Chebly, R. Talj and A. Charara: Coupled longitudinal/lateral controllers for autonomous vehicles navigation, with experimental validation. *Control Engineering Practice*, **88**, (2019), 79–96. DOI: 10.1016/j.conengprac.2019.05.001