JOURNAL OF THEORETICAL
AND APPLIED MECHANICS
https://doi.org/10.15632/jtam-pl/208127
ONLINE FIRST

THE MODEL-BASED ALGORITHM FOR AUTONOMOUS VEHICLE PATH FOLLOWING

Michał BRZOZOWSKI

Department of Combustion Engines and Vehicles, University of Bielsko-Biala, Bielsko-Biala, Poland
mbrzozowski@ubb.edu.pl

This paper presents a proprietary steering algorithm for path following, whose advantage lies in its ability to be applied with vehicle dynamic models of varying complexity. The proposed algorithm was implemented using models with 3 and 10 degrees of freedom, which had been previously verified. The results were compared with those obtained using the geometric pure pursuit (PP) algorithm. Both algorithms require path approximation. In this study, path approximation was conducted using B3 functions. The presented computer simulation results indicate that the proposed steering angle selection algorithm demonstrates greater accuracy than the PP algorithm.

Keywords: autonomous vehicle; vehicle dynamics; vehicle modelling.

1. Introduction

One of the more challenging problems in the deployment of highly automated vehicles is the provision of an appropriate control system that enables the execution of a predefined driving path. This task involves determining the steering angle trajectory of the front wheels based on a known velocity profile in such a way that the vehicle follows the desired path. Numerous vehicle control methods exist, and their comparisons and benchmarking can be found in works such as Liu *et al.* (2021) and Diachuk and Easa (2022). Path-following methods generally fall into three main categories: geometric methods, model-based methods, and learning-based (artificial intelligence) methods.

Among the popular geometric methods, which rely on the curvature of the path, the pure pursuit (PP) algorithm stands out. Originally formulated in the 1990s (Coulter, 1992), it is still widely used in autonomous driving applications (Gámez-Serna *et al.*, 2017). Huang *et al.* (2018) present an application of the PP algorithm, focusing on the determination of the constant l_d, which is essential for the proper functioning of the algorithm and is related to vehicle speed. Setting the constant too high reduces trajectory-tracking accuracy, which becomes critical during maneuvers such as U-turns. Conversely, a too-low value leads to oscillations in the steering angle. This algorithm continues to be used in vehicle control tasks, for example in (An *et al.*, 2025).

Another well-known geometric control algorithm is Stanley control (SC), whose application was described by Yang *et al.* (2017), with particular attention to issues related to path curvature computation, which significantly influences the behavior of geometric algorithms. An alternative

Ministry of Science and Higher Education
Republic of Poland

This publication has been funded by the Polish Ministry of Science and Higher Education under the Excellent Science II programme "Support for scientific conferences".
The content of this article was presented during the 61st Symposium "Modelowanie w mechanice" (Modelling in Mechanics), Szczyrk, Poland, March 2–5, 2025.

use of the SC algorithm was proposed by Amer *et al.* (2018), where it was combined with optimization methods. Cibooglu *et al.* (2017) present a hybrid approach combining the PP and SC algorithms to enhance precision. Other, less common geometric algorithms include the one described in the monograph by Rajamani (2012). A comprehensive comparison of four geometric algorithms is provided by Brzozowski (2025).

The advantages of geometric algorithms include their ease of implementation, low computational requirements, and high effectiveness in simple scenarios (e.g., low-curvature routes). Their disadvantages include poor performance at higher speeds, moderate trajectory-tracking accuracy, and sensitivity to the tuning of auxiliary/scaling parameters. Additionally, these methods do not account for the vehicle dynamics model.

Model-based methods (using either kinematic or dynamic vehicle models) describe the vehicle's motion. Typical examples include optimization-based approaches such as model predictive control (MPC) and the linear quadratic regulator (LQR). These methods are also referred to as optimal control strategies.

The LQR method uses a feedback gain matrix to minimize a cost function (Li *et al.*, 2019; Lee *et al.*, 2019). While LQR provides greater accuracy than geometric algorithms, it is less computationally efficient. A significant drawback is the necessity of linearization and the inability to handle constraints.

Another class of optimization-based methods is represented by MPC, which predicts the future behavior of the vehicle over a sequence of subintervals within the total planning horizon based on a mathematical model. MPC performs real-time optimization to follow a reference path while satisfying constraints. Variants of this model are discussed in (Zhang *et al.*, 2019) and (Fu *et al.*, 2022), where the algorithm is adapted to specific needs. MPC's advantages include its ability to incorporate constraints, predict and avoid problems in advance, and work with nonlinear vehicle models. It is considered a highly accurate control method, although its disadvantages include high complexity and low computational efficiency.

A large class of control approaches consists of learning-based methods. These do not require the development of an explicit vehicle model and are suitable when detailed information about the vehicle or its environment is unavailable, but large amounts of driving data exist. MPC is often used as a preliminary tool for training such models. An example of reinforcement learning–based control can be found in (Cao *et al.*, 2023). Among learning-based methods, fuzzy logic–based approaches are also noteworthy, such as in (Elsayed *et al.*, 2018).

In this study, a proprietary algorithm was applied to solve the path-following task. Although it does not directly fall into any of the previously mentioned categories, it requires a dynamic vehicle model for its operation. Therefore, it can be classified alongside methods such as LQR and MPC. It is assumed that the equations of motion take the following form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{f}\left(\mathbf{q}, \dot{\mathbf{q}}, F(t), u\right), \tag{1.1}$$

where $\mathbf{M}$ – inertia matrix, $\mathbf{q}$ – vector of generalized coordinates, u – control parameter.

The linearization of the system would consist in transforming these equations into the following form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}, F(t)) + \mathbf{G}u, \tag{1.2}$$

where $\mathbf{G}$ – input (or control) distribution matrix, $\mathbf{h}$ – vector of generalized forces.

In the case where the control parameter is $u = \delta$ (the steering angle of the front wheels), it can be determined in one of the following ways:

- as the result of an optimization process (as in LQR, NLQR, MPC, or NMPC methods),
- as a solution to the problem of selecting the steering angle δ by repeated integration of the equations of motion, as proposed in this study. This method does not require linearization.

The algorithm proposed in this study (referred to as MPC/B) is an approach that enables the determination of the steering angle δ, using a vehicle model of arbitrary complexity, without the need for linearization or the application of optimization methods. Control inputs are determined in stages – within sequentially occurring subintervals of the total simulation time. To evaluate the proposed control algorithm, simulation studies were conducted, comparing the results obtained using the MPC/B algorithm to those achieved with the PP algorithm. Two vehicle dynamic models were formulated, and the path was approximated using third-degree spline functions.

Although the geometric PP algorithm does not require a dynamic model to compute the steering angle, it does require information about the vehicle's position in space. In the case of simulation studies, this positional information is provided by the vehicle dynamics model.

2. Nonlinear vehicle models

This study presents two vehicle models with 3 and 10 degrees of freedom (DoF). The 10-DoF model is a three-dimensional model. It is formulated under the assumption that the vehicle is treated as a rigid body with 6 degrees of freedom (the chassis), to which four rotating wheels are attached (Fig. 1).

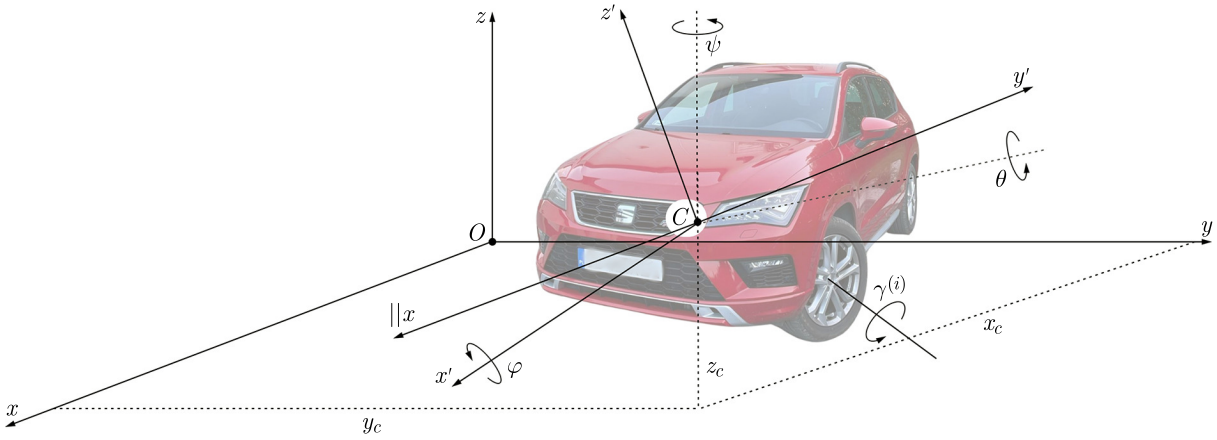

Fig. 1. Diagram of the 10 degrees of freedom model.

To derive the equations of motion for the chassis along with the attached concentrated masses (including wheels and suspensions), a formalism based on the Newton-Euler equations was applied (Blajer, 1998). The Newton-Euler equations for the chassis take the following form:

$$m\dot{\mathbf{V}}_c = \sum_i \mathbf{F}_i, \qquad \frac{\mathrm{d}\mathbf{k}_c}{\mathrm{d}t} = \sum_i \mathbf{M}_{i_c}, \tag{2.1}$$

where $\mathbf{V}_c$ – the velocity vector of the body center of mass, $\mathbf{k}_c$ – angular velocity of the body relative to the center of mass C, $\mathbf{F}_i$ – external forces acting on the body, $\mathbf{M}_{i_c}$ – moments of external forces relative to the center of mass C, m – total mass of the vehicle including the wheels.

The equations of motion for the wheels can be written in the form:

$$I^{(i)}\ddot{\gamma}^{(i)} = \sum_j M_j'^{(i)}, \quad i = 1, 2, 3, 4, \tag{2.2}$$

where $I^{(i)}$ – moment of inertia of the wheel about its axis of rotation, $\gamma^{(i)}$ – wheel rotation angle, $\sum_j M_j'^{(i)}$ – sum of moments of forces acting on the wheel about its axis of rotation.

The vector of generalized coordinates comprising the body and the four wheels of the vehicle thus takes the form:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_n \\ \boldsymbol{\Gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_c \\ \boldsymbol{\Phi} \\ \boldsymbol{\Gamma} \end{bmatrix}, \tag{2.3}$$

where

$$\mathbf{r}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, \qquad \boldsymbol{\Phi} = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} q_4 \\ q_5 \\ q_6 \end{bmatrix}, \qquad \boldsymbol{\Gamma} = \begin{bmatrix} \gamma^{(1)} \\ \gamma^{(2)} \\ \gamma^{(3)} \\ \gamma^{(4)} \end{bmatrix} = \begin{bmatrix} q_7 \\ q_8 \\ q_9 \\ q_{10} \end{bmatrix}.$$

To determine the road reaction forces on the wheels, the brush model (Pacejka & Sharp, 1991), modified and described in detail in (Rajamani, 2012), was applied. A detailed description of the 10 degrees of freedom dynamic model and the tire model used can be found in (Brzozowski, 2025). In vehicle dynamics modeling for autonomous driving, the most commonly used dynamic model is the planar model with 3 degrees of freedom, also known as the bicycle or moped model (Gillespie, 1992; Ajanović *et al.*, 2023). Figure 2 shows the vehicle representation in the 3 degrees of freedom model. This model accounts for the vehicle's displacement in the xy plane and the yaw angle ψ around z'-axis of the local coordinate system $\{C\}'$ which is parallel to the z-axis of the road coordinate system $\{O\}$. The vehicle dynamics are described by the components of the vector shown in Fig. 2:

$$\mathbf{q} = \begin{bmatrix} V_x' \\ V_y' \\ \psi \end{bmatrix}, \tag{2.4}$$

where V_x', V_y' – the components of the vehicle velocity vector in the local coordinate system $\{C\}'$, ψ – yaw angle.

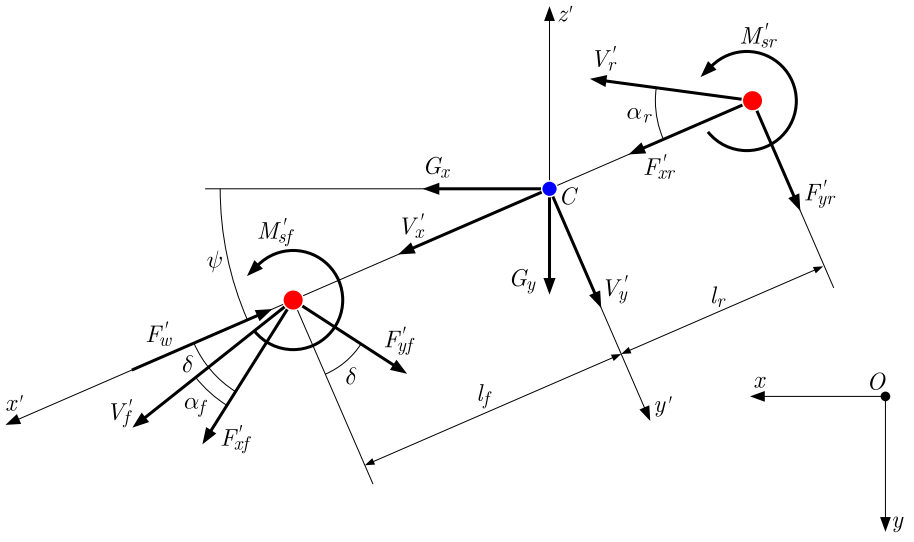

Fig. 2. Diagram of the bicycle model (Gillespie, 1992).

The equations of motion with generalized coordinates as in (2.4) take the form:

$$\begin{aligned} m\left(\dot{V}'_x - \dot{\psi}V'_y\right) &= F'_{xf}c\delta - F'_{yf}s\delta + F'_{xr} - F'_w + G_x c\psi + G_y s\psi, \\ m\left(\dot{V}'_y + \dot{\psi}V'_x\right) &= F'_{xf}s\delta + F'_{yf}c\delta + F'_{yr} - G_x\, s\psi + G_y c\psi, \\ I_{z'}\ddot{\psi} &= \left(F'_{xf}\, s\delta + F'_{yf}c\delta\right) l_f - F'_{yr}l_r + M'_{sf} + M'_{sr}, \end{aligned} \tag{2.5}$$

where $I_{z'}$ – mass moment of inertia of the vehicle about the axis z', M'_{sf}, M'_{sr} – self-aligning moments, F'_{xf}, F'_{yf}, F'_{xr}, F'_{yr} – components of the tire-road interaction forces acting on the vehicle's wheels, F'_w – drag force, l_f, l_r – distance of the front and rear wheel axles from the vehicle's center of mass, G_x, G_y – components of the gravitational force (equal to zero when the road is not inclined).

To determine the tire-road interaction forces, formulas are used that relate the forces in the road plane to the friction coefficients and normal reactions.

Below are the verification results of both models through comparison of own calculations with results obtained using the CarSim software. The maneuver of a double lane change at a vehicle speed of 80 km/h was simulated (total simulation time $t_K = 12$ s, maximum lateral displacement of the vehicle at time $t = 4.9$ s was 3.69 m/s^2). Vehicle parameters were based on the CarSim A-Class Hatchback. The assumed front wheel steering angle is shown in Fig. 3a. Figure 3b presents the calculated vehicle trajectory.

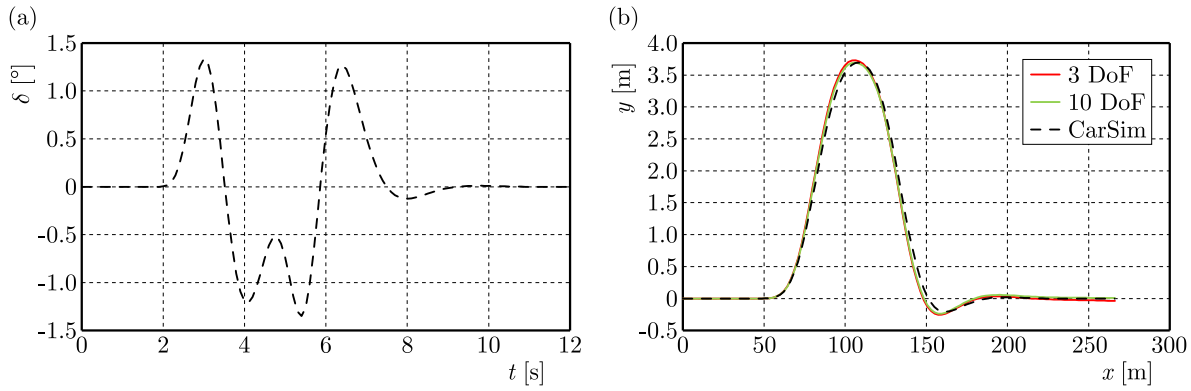

Fig. 3. Model validation: (a) assumed front wheel steering angle; (b) vehicle trajectory, own models and CarSim.

To assess the accuracy of the formulated models, the following error metrics were used:
– mean absolute error

$$\bar{\varepsilon} = \frac{1}{n}\sum_{i=1}^{n} |\varepsilon_i|\,, \tag{2.6}$$

– relative error

$$\varepsilon_{\%} = \left|\frac{w_o^{\max} - w_m^{\max}}{w_o^{\max}}\right| \cdot 100\,[\%], \tag{2.7}$$

where $\varepsilon_i = \varepsilon(t_i) = w_{o,i} - w_{m,i}$, $w_{o,i}$ – reference value obtained using the CarSim software, $w_{m,i}$ – value obtained according to the own model, $w_o^{\max} = \max\limits_{1\le i\le n} |w_{o,i}|$, $w_m^{\max} = \max\limits_{1\le i\le n} |w_{m,i}|$, n – number of compared values.

The results presented in Table 1 indicate that both own models yield errors $\bar{\varepsilon}$ in displacements on the order of several centimeters over a route nearly 270 meters long. The displacements and

Table 1. Calculated mean $\overline{\varepsilon}$ and percentage $\varepsilon_{\%}$ differences between the own models and CarSim.

Model (DoF)	$\overline{\varepsilon}$				$\varepsilon_{\%}$			
	x [m]	y [m]	ψ [°]	$\dot{\psi}$ [°/s]	x	y	ψ	$\dot{\psi}$
3	0.150	0.057	0.219	0.392	0.06	0.97	2.65	2.05
10	0.150	0.052	0.217	0.381	0.06	0.04	2.11	0.90

angular velocities ψ and $\dot{\psi}$ show larger errors between the own models and the CarSim software. However, these do not exceed 3 %. Therefore, both presented models can be considered valid. The 3 degrees of freedom model was previously verified in (Brzozowski & Drąg, 2023), and the 10 degrees of freedom model in (Brzozowski, 2025).

3. Path approximation

Proper path planning has a decisive impact on the path-following task (Zhong *et al.*, 2025; Guo *et al.*, 2025). There are many path planning methods. A review of these can be found in (Katrakazas *et al.*, 2015; Paden *et al.*, 2016). In this work, approximation using cubic B-spline functions (B3) was applied. It is assumed that the function approximating the path $f(x)$ has the form:

$$f(x) = \sum_{i=-1}^{n+1} a_i \varphi_i(x), \tag{3.1}$$

where a_i – coefficients, φ_i – cubic B-spline basis functions (B3), n – number of subintervals into which the interval $\langle A, B\rangle$ is divided. In the case where the interval $\langle A, B\rangle$ is divided into equal segments of length h, it takes the values:

$$x_i = ih \quad \text{for} \quad i = -3, -2, -1, 0, 1, ..., n, n+1, n+2, n+3, \tag{3.2}$$

functions $\varphi_i(x)$ are defined as follows:

$$\varphi_i(x) = \begin{cases} 0 & \text{when} \quad x < x_{i-2}, \\ (x - x_{i-2})^3 & \text{when} \quad x \in \langle x_{i-2}, x_{i-1}\rangle, \\ -3(x - x_{i-1})^3 + 3h(x - x_{i-1})^2 + 3h^2(x - x_{i-1}) + h^3 & \text{when} \quad x \in \langle x_{i-1}, x_i\rangle, \\ 3(x - x_{i+1})^3 + 3h(x - x_{i+1})^2 - 3h^2(x - x_{i+1}) + h^3 & \text{when} \quad x \in \langle x_i, x_{i+1}\rangle, \\ -(x - x_{i+2})^3 & \text{when} \quad x \in \langle x_{i+1}, x_{i+2}\rangle, \\ 0 & \text{when} \quad x > x_{i+2}. \end{cases} \tag{3.3}$$

The function $\varphi_i(x)$ along with its characteristic values is shown in Fig. 4.

The coefficients a_i present in Eq. (3.1) for $i = -1, 0, 1..., n, n+1$ are determined by minimizing the functional:

$$\Omega(a_{-1}, ..., a_{n+1}) = \sum_{k=0}^{m} [f(x_k) - y_k^e]^2 \to \min, \tag{3.4}$$

where y_k^e – measured value $y(x_k)$, $m+1$ – number of measurement points.

After transformations, to determine the $n+3$ coefficients $a_{-1}, ..., a_{n+1}$, a system of $n+3r$ linear algebraic equations of the form is obtained:

$$\sum_{i=-1}^{n+1} a_i \left[\sum_{k=0}^{m} \varphi_i(x_k^e)\, \varphi_j(x_k^e)\right] = \sum_{k=0}^{m} y_k^e \varphi_j(x_k^e) \tag{3.5}$$

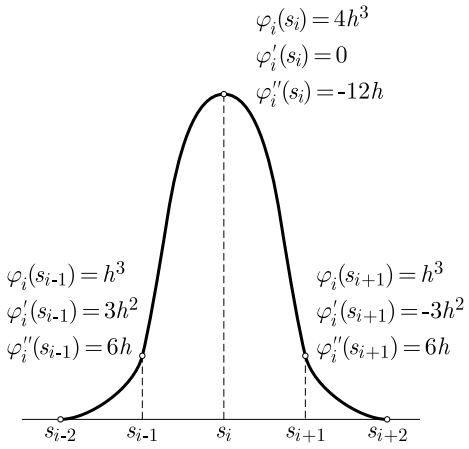

Fig. 4. Basis functions $\varphi_i(x)$.

for $j = -1, 0, 1, ..., n, n+1$.

This is a system of $n+3r$ linear algebraic equations. Solving this system enables the determination of the coefficients of the function f in formula (3.1). In problems related to approximating the vehicle's trajectory, the values of the function $f(x)$ and its derivatives at the beginning and end of the approximation interval are generally known, which enables the determination of up to 6 coefficients among $a_{-1}, ..., a_{n+1}$. To account for cases where the approximated path is not a function in the mathematical sense, the following procedure was applied.

If the points $(x_0^e, y_0^e), ..., (x_m^e, y_m^e)$ are sufficiently dense, the distance traveled by the vehicle can be approximately calculated as follows:

$$s_i^e = \sum_{j=1}^{i} \sqrt{\left(x_j^e - x_{j-1}^e\right)^2 + \left(y_j^e - y_{j-1}^e\right)^2}. \tag{3.6}$$

Assuming the vehicle speed is greater than zero, the values s_i^e form an increasing sequence. Therefore, the coordinates x and y can be treated as functions of the variable sss (in the mathematical sense). To determine the approximating functions $x(s)$ and $y(s)$, two problems analogous to the one presented above need to be solved, assuming:

$$\begin{aligned} &x_i^e = s_i^e \quad \text{and} \quad y_i^e = x_i^e \quad \text{when calculating the coefficients of the function} \quad x(s), \\ &x_i^e = s_i^e \quad \text{and} \quad y_i^e = y_i^e \quad \text{when calculating the coefficients of the function} \quad y(s). \end{aligned} \tag{3.7}$$

It is necessary to take into account the initial and boundary conditions for each of the functions $x(s)$, $y(s)$.

4. Own MPC/B algorithm for selecting the front wheel steering angle $\delta(t)$

We assume that the nonlinear equation describing the dynamics of the autonomous vehicle takes the form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \delta(t)), \tag{4.1}$$

where $\mathbf{M}(\mathbf{q})$ – inertia matrix, $\mathbf{q}$, $\dot{\mathbf{q}}$ – vectors of vehicle coordinates and velocities, $\delta(t)$ – function describing the steering angle trajectory of the vehicle's front wheels (to be determined). It is assumed that the vehicle's path and velocity profile are known. The integration interval $\langle 0, T \rangle$ for the equations of motion (4.1) is divided into subintervals of length

$$\Delta t = mh_c, \tag{4.2}$$

where h_c – the integration step of the equations of motion, m_c – the multiple of the integration step.

Assuming that at $t = t_0$ the steering angle is known:

$$\delta_0 = \delta(t_0) \tag{4.3}$$

and the initial conditions:

$$q_0 = q(t_0), \qquad \dot{q}_0 = \dot{q}(t_0). \tag{4.4}$$

The sought value of the steering angle is denoted as

$$\delta_m = \delta(t_0 + \Delta t) = \delta(t_m) \tag{4.5}$$

which changes linearly over the interval $\langle t_0, t_0 + \Delta t\rangle = \langle t_0, t_m\rangle$ according to the relation:

$$\delta(t) = \delta_0 + \frac{\delta_m - \delta_0}{\Delta t}(t - t_0). \tag{4.6}$$

It is assumed that the quantity δ_m should minimize the expression:

$$\begin{aligned}\Delta^2(\delta_m) = {} & C_0^{x,y}\left[(x_{c,m} - x_{T,m})^2 + (y_{c,m} - y_{T,m})^2\right] + C_0^{\psi}[\psi_m - \psi_{T,m}]^2 \\ & + C_1^{x,y}\left[(\dot{x}_{c,m} - \dot{x}_{T,m})^2 + (\dot{y}_{c,m} - \dot{y}_{T,m})^2\right] + C_1^{\psi}[\dot{\psi}_m - \dot{\psi}_{T,m}]^2,\end{aligned} \tag{4.7}$$

where $x_{c,m} = x_c(t_m)$, $y_{c,m} = y_c(t_m)$, $\psi_m = \psi(t_m)$. They are the coordinates of the vehicle's center of mass and its yaw angle, calculated using the vehicle dynamics model at time t_m, whereas $x_{T,m} = x_T(t_m)$, $y_{T,m} = y_T(t_m)$, $\psi_{T,m} = \psi_T(t_m)$. They are the desired path coordinates and the tangent angle to the vehicle trajectory, calculated according to the path approximation algorithm at time t_m.

To calculate $\Delta_m^2(\delta_m)$, the equation of motion (4.1) must be integrated with $\delta(t)$ defined by (4.6). The quadratic form (4.7) reaches its minimum when:

$$\begin{aligned}\frac{\partial \Delta^2(\delta_m)}{\partial \delta} = 2\Bigg\{ & C_0^{x,y}\left[(x_{c,m} - x_{T,m})\frac{\partial x_{c,m}}{\partial \delta} + (y_{c,m} - y_{T,m})\frac{\partial y_{c,m}}{\partial \delta}\right] + C_0^{\psi}(\psi_m - \psi_{T,m})\frac{\partial \psi_m}{\partial \delta} \\ & + C_1^{x,y}\left[(\dot{x}_{c,m} - \dot{x}_{T,m})\frac{\partial \dot{x}_{c,m}}{\partial \delta} + (\dot{y}_{c,m} - \dot{y}_{T,m})\frac{\partial \dot{y}_{c,m}}{\partial \delta}\right] \\ & + C_1^{\psi}\left(\dot{\psi}_m - \dot{\psi}_{T,m}\right)\frac{\partial \dot{\psi}_m}{\partial \delta}\Bigg\} = 0,\end{aligned} \tag{4.8}$$

where $\frac{\partial p}{\partial \delta}\, p \in \Omega = \left\{x_{c,m}, y_{c,m}, \psi_m, \dot{x}_{c,m}, \dot{y}_{c,m}, \dot{\psi}_m\right\}$ is the derivative of the function p with respect to δ for $t = t_m$.

To calculate these quantities in this paper, the five-point finite difference method was applied, assuming:

$$\frac{\partial p}{\partial \delta}_{\delta=\delta_m} = \frac{p(\delta_0 - 2\delta_m) - 8p(\delta_0 - \delta_m) + 8p(\delta_0 + \delta_m) - p(\delta_0 + 2\delta_m)}{12\delta_m}. \tag{4.9}$$

To calculate $\frac{\partial p}{\partial \delta}$ for $p \in \Omega t$, it is therefore necessary to quadratically integrate the equation of motion $(3.7)_1$ in the interval $\langle t_0, t_m\rangle$.

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

$$\delta_m^0 = \delta_0, \qquad \delta_m^{(i)} = \delta_m^{(i-1)} - \frac{\gamma_i(\delta_m)}{\gamma_i'(\delta_m)}, \tag{4.10}$$

where

$$\gamma_i(\delta_m) = \frac{1}{2}\frac{\partial \Delta^2(\delta_m)}{\partial \delta}, \qquad \gamma_i'(\delta_m) = \frac{\partial \gamma_i(\delta_m)}{\partial \delta}.$$

The iterative process was conducted until one of the conditions was met:

$$i = i_{\mathrm{MAX}}, \qquad \left|\frac{\gamma_i(\delta_m)}{\gamma_i'(\delta_m)}\right| < \mathrm{EPS}, \tag{4.11}$$

where i_{MAX} and EPS are quantities defining the maximum number of iterations and the absolute error in determining the δ_m, respectively.

The derivative of $\gamma_i'(\delta_m)$ was also calculated using the five-point finite difference method.

In summary, to determine δ_m, it is necessary, according to the proposed algorithm, to integrate the equations of motion (4.1) over the interval $\langle t_0, t_m \rangle$ at most:

$$N = i_{\mathrm{MAX}} 5\mathrm{times}. \tag{4.12}$$

5. Simulation research

Simulation studies were performed for a loop in which the trajectory (Fig. 5) is described by the formulae:

$$\begin{cases} x = a\sin(s), \\ y = a\sin(s)\cos(s), \end{cases}$$

where $a = 50$.

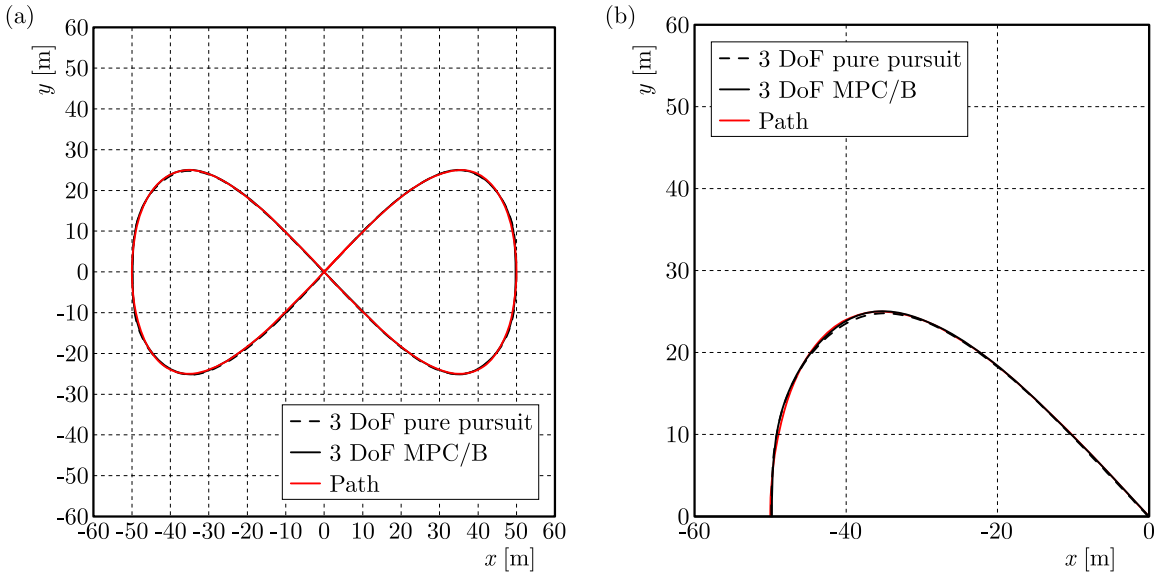

Fig. 5. Trajectory for the implementation of the "loop" maneuver: (a) total; (b) enlargement.

The execution time for the entire maneuver is $t_k = 39\,\mathrm{s}$, and the total distance is 304 metres. A constant speed of $v = 28\,\mathrm{km/h}$ was assumed.

In the present task, the constants $C_0^{x,y}$ and $C_1^{x,y}$ are taken as 10^3 and 10^2, while the constant L_d needed for the PP algorithm to work properly as 0.05.

The trajectory shown in Fig. 5 indicates that there is little difference between control using the PP algorithm and the proprietary algorithm. The proposed algorithm is slightly more accurate. Due to the small differences, the figure does not show the results of the calculation using the dynamics model with 10 DoF. The maximum values of the mapping error are shown in Table 2.

Table 2. Mapping error.

Dynamics model	Steering algorithm	
	PP	MPC/B
3 DoF	0.298	0.098
10 DoF	0.409	0.384

The results indicate a higher accuracy of the proposed algorithm, especially when combined with a low complexity vehicle dynamics model. Figure 6 shows the course of the steering angle. The MPC/B algorithm determined a slightly larger steering angle than the PP algorithm. The maximum difference was 1.64°.

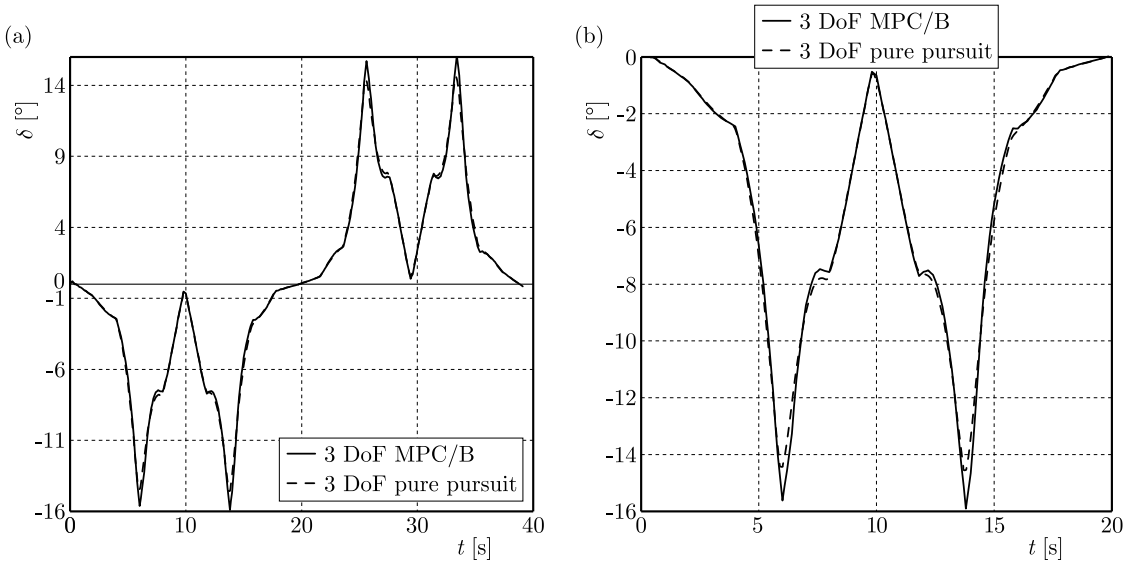

Fig. 6. Diagram of the course of the steering angle: (a) total; (b) magnification.

6. Results and discussion

Among the most popular vehicle control methods are: geometric, model-based (optimization) and machine learning. Developing control methods that ensure path realization is an important research problem. This paper compares the classical geometric algorithm PP with the proprietary MPC/B algorithm, which belongs to the group of MPC algorithms based on a model of vehicle dynamics. The proposed algorithm is more accurate than the PP algorithm. It has additional advantages, such as the ability to be used with any dynamics model or to be tuned for specific requirements. A weakness is the moderate computational efficiency. For the presented "loop" maneuver, the computation time with the PP algorithm was 2.14 s for MPC/B 4.94 s. Table 3 shows a synthesis of the conclusions and a comparison of the PP algorithm with the proposed own algorithm.

Table 3. Comparison of the PP algorithm and own algorithm MPC/B.

Dynamics model	PP	MPC/B
	Does not use	Any of the following may be used
Choice of constants	Limited tuning ability	Trajectory or yaw angle tuning possible
Precision	Moderate	High
Numerical effectiveness	High	Moderate

In future work, it seems expedient to compare the proposed algorithm with a standard MPC-type algorithm.

References

1. Ajanović, Z., Regolin, E., Shyrokau, B., Ćatić, H., Horn, M., & Ferrara, A. (2023). Search-based task and motion planning for hybrid systems: Agile autonomous vehicles. *Engineering Applications of Artificial Intelligence*, *121*, Article 105893. https://doi.org/10.1016/j.engappai.2023.105893

2. Amer, N.H., Zamzuri, H., Hudha, K., Aparow, V.R., Kadir, Z.A., & Abdin, A.F.Z. (2018). Path tracking controller of an autonomous armoured vehicle using modified Stanley controller optimized with particle swarm optimization. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *40*(2), Article 104. https://doi.org/10.1007/s40430-017-0945-z

3. An, L., Huang, X., Yang, P., & Liu, Z. (2025). Adaptive Bézier curve-based path following control for autonomous driving robots. *Robotics and Autonomous Systems*, *189*, Article 104969. https://doi.org/10.1016/j.robot.2025.104969

4. Blajer, W. (1998). *Methods of multibody system dynamic* (in Polish). Radom: Politechnika Radomska.

5. Brzozowski, M. (2025). *Motion planning for autonomous vehicles using dynamic models* (in Polish) [Unpublished doctoral dissertation]. Faculty of Mechanical Engineering and Computer Science, University of Bielsko-Biala, Poland.

6. Brzozowski, M., & Drąg, Ł. (2023). Application of dynamic optimization for autonomous vehicle motion control. *Transport Problems*, *18*(2), 209-222. https://doi.org/10.20858/tp.2023.18.2.18

7. Cao, Y., Ni, K., Jiang, X., Kuroiwa, T., Zhang, H., Kawaguchi, T., Hashimoto, S., & Jiang, W. (2023). Path following for autonomous ground vehicle using DDPG algorithm: A reinforcement learning approach. *Applied Sciences*, *13*(11), Article 6847. https://doi.org/10.3390/app13116847

8. Cibooglu, M., Karapinar, U., & Söylemez, M.T. (2017). Hybrid controller approach for an autonomous ground vehicle path tracking problem. *2017 25th Mediterranean Conference on Control and Automation (MED)* (pp. 583–588). IEEE. https://doi.org/10.1109/MED.2017.7984180

9. Coulter, R.C. (1992). *Implementation of the pure pursuit path tracking algorithm* (Technical Report CMU-RI-TR-92-01). Robotics Institute, Carnegie Mellon University. https://www.ri.cmu.edu/pub_files/pub3/coulter_r_craig_1992_1/coulter_r_craig_1992_1.pdf

10. Diachuk, M., & Easa, S.M. (2022). Motion planning for autonomous vehicles based on sequential optimization. *Vehicles*, *4*(2), 344–374. https://doi.org/10.3390/vehicles4020021

11. Elsayed, H., Abdullah, B.A., & Aly, G. (2018). Fuzzy logic based collision avoidance system for autonomous navigation vehicle. *2018 13th International Conference on Computer Engineering and Systems (ICCES)* (pp. 469–474). IEEE. https://doi.org/10.1109/ICCES.2018.8639396

12. Fu, T., Zhou, H., & Liu, Z. (2022). NMPC-based path tracking control strategy for autonomous vehicles with stable limit handling. *IEEE Transactions on Vehicular Technology*, *71*(12), 12499–12510. https://doi.org/10.1109/TVT.2022.3196315

13. Gámez Serna, C., Lombard, A., Ruichek, Y., & Abbas-Turki, A. (2017). GPS-based curve estimation for an adaptive pure pursuit algorithm. In G. Sidorov, & O. Herrera-Alcántara (Eds.), *Lecture notes in computer science: Vol. 10061. Advances in computational intelligence* (pp. 497–511). Springer. https://doi.org/10.1007/978-3-319-62434-1_40

14. Gillespie, T.D. (1992). *Fundamentals of vehicle dynamics*. SAE International.

15. Guo, S., Gong, J., Shen, H., Yuan, L., Wei, W., & Long, Y. (2025). DBVSB-P-RRT*: A path planning algorithm for mobile robot with high environmental adaptability and ultra-high speed planning. *Expert Systems with Applications*, *266*, Article 126123. https://doi.org/10.1016/j.eswa.2024.126123

16. Huang, P., Zhang, Z., Luo, X., Zhang, J., & Huang, P. (2018). Path tracking control of a differential-drive tracked robot based on look-ahead distance. *IFAC-PapersOnLine*, *51*(17), 112–117. https://doi.org/10.1016/j.ifacol.2018.08.072

17. Katrakazas, C., Quddus, M., Chen, W-H., & Deka, L. (2015). Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, *60*, 416–442. https://doi.org/10.1016/j.trc.2015.09.011

18. Lee, K., Jeon, S., Kim, H., & Kum, D. (2019). Optimal path tracking control of autonomous vehicle: Adaptive full-state linear quadratic Gaussian (LQG) control. *IEEE Access*, *7*, 109120–109133. https://doi.org/10.1109/ACCESS.2019.2933895

19. Li, S., Wang, G., Zhang, B., Yu, Z., & Cui, G. (2019). Vehicle stability control based on model predictive control considering the changing trend of tire force over the prediction horizon. *IEEE Access*, *7*, 6877–6888. https://doi.org/10.1109/ACCESS.2018.2889997

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

21. Pacejka, H.B., & Sharp, R.S. (1991). Shear force development by pneumatic tyres in steady state conditions: A review of modelling aspects. *Vehicle System Dynamics*, *20*(3–4), 121–175. https://doi.org/10.1080/00423119108968983

22. Paden, B., Čáp, M., Yong, S.Z., Yershov, D., & Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, *1*(1), 33–55. https://doi.org/10.1109/TIV.2016.2578706

23. Rajamani, R. (2012). *Vehicle dynamics and control*. Springer. https://doi.org/10.1007/978-1-4614-1433-9

24. Yang, J., Bao, H., Ma, N., & Xuan, Z. (2017). An algorithm of curved path tracking with prediction model for autonomous vehicle. *2017 13th International Conference on Computational Intelligence and Security (CIS)* (pp. 405–408). IEEE. https://doi.org/10.1109/CIS.2017.00094

25. Yang, Y., Li, Y., Wen, X., Zhang, G., Ma, Q., Cheng, S., Qi, J., Xu, L., & Chen, L. (2022). An optimal goal point determination algorithm for automatic navigation of agricultural machinery: Improving the tracking accuracy of the Pure Pursuit algorithm. *Computers and Electronics in Agriculture*, *194*, Article 106760. https://doi.org/10.1016/j.compag.2022.106760

26. Zhang, B., Zong, C., Chen, G., & Li, G. (2019). An adaptive-prediction-horizon model prediction control for path tracking in a four-wheel independent control electric vehicle. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *233*(12), 3246–3262. https://doi.org/10.1177/0954407018821527

27. Zhong, J., Kong, D., Wei, Y., Hu, X., & Yang, Y. (2025). Efficiency-optimized path planning algorithm for car-like mobile robots in bilateral constraint corridor environments. *Robotics and Autonomous Systems*, *186*, Article 104923. https://doi.org/10.1016/j.robot.2025.104923

Manuscript received June 3, 2025; accepted for publication July 10, 2025; published online September 2, 2025.