Minimum Curvature Trajectory Planning for Autonomous Vehicles in a Hierarchical Framework

Dániel Losonczi[1], Árpád Fehér[2], Szilárd Aradi[3], and László Palkovics[4]

Abstract— **This paper presents an approach for global trajectory planning using quadratic optimization and dynamic programming. In the global route planning phase, the goal is to find a route with minimal curvature for any road geometry, while a speed profile is determined by dynamic programming based on the vehicle's dynamic constraints. The method has a low computational cost and is well-suited for integration within a hierarchical system architecture, where the trajectory can be further refined with local route planning to enable the vehicle to adapt to changing environmental conditions. The developed method ensures vehicle stability while achieving the highest possible speed. The approach is implemented in the Frenet coordinate system within a self-developed simulation environment that allows for a comprehensive evaluation of vehicle dynamics, trajectory feasibility, and performance metrics. The results show that the proposed method can generate smooth and dynamically feasible trajectories while balancing curvature minimization, safety, and computational efficiency, making it well-suited for real-world applications.**

I. INTRODUCTION

For autonomous vehicles, route planning and the execution of maneuvers that ensure safe operation are of paramount importance. This entails numerous requirements. To ensure operational safety in real-world conditions, the trajectory must consider the vehicle's dynamics and parameters so the controller can carry out the target task. Fast operation and computational efficiency are essential. Considering the problem, due to the nonlinearity of the vehicle, finding the optimal solution for generating a route is PSPACE-hard [1]. Solving these problems in real time poses a significant challenge in the route planning of highly automated vehicles, as there is no general optimal solution. Therefore, practical approaches often rely on simplifications or heuristics to achieve suboptimal solutions that are quick and safe to implement.

[1]Dániel Losonczi is with the Systems and Control Laboratory, HUN-REN Institute for Computer Science and Control (HUN-REN SZTAKI), Kende utca 13-17., H-1111 Budapest, Hungary. losonczi.daniel@sztaki.hun-ren.hu

[2]Árpád Fehér is with the Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics (BME), Műegyetem rkp. 3., H-1111 Budapest, Hungary. feher.arpad@kjk.bme.hu

[3]Szilárd Aradi is with the Department of Control for Transportation and Vehicle Systems, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics (BME), Műegyetem rkp. 3., H-1111 Budapest, Hungary, and the Systems and Control Laboratory, HUN-REN Institute for Computer Science and Control (HUN-REN SZTAKI), Kende utca 13-17., H-1111 Budapest, Hungary. aradi.szilard@kjk.bme.hu

[4]László Palkovics is with Széchenyi István University, Egyetem tér 1., H-9026 Győr, Hungary. laszlo.palkovics@sze.hu

This study presents a method that employs a quadratic programming-based optimization approach [2] to generate a minimum-curvature route and uses dynamic programming to determine the speed profile. Our proposed global route planning method can be easily integrated into the hierarchical structure presented by [3]. According to this framework, autonomous driving functions are organized into interacting layers. The system consists of four distinct hierarchical levels. At the highest level, the global route planning layer designs a route from the vehicle's current position to the designated destination, using tools such as directed graphs or network-based algorithms. This is followed by the behavioral decision-making layer, whose task is to make strategic decisions and determine the vehicle's immediate objectives, for example, keeping a lane or executing a lane change. In addition, this layer coordinates the motion planning process based on the output of the route planner and the dynamically changing environment. Afterward, the motion planning layer develops a local trajectory, that is tailored to the desired maneuver, which is executed by the local feedback control layer, ensuring the planned trajectory is implemented precisely and accurately.

II. RELATED WORK

Autonomous driving has been a significant focus of research for years, evolving from urban and highway navigation to high-speed autonomous race cars. These vehicles push the limits of dynamics and drive advancements in trajectory planning, decision-making, and control systems under extreme conditions, leading to a wide range of methods that address the complexities of this problem and highlight its importance in research and industry.

Safety is a fundamental aspect of autonomous driving, with several frameworks addressing its challenges. The Holistic Safety Framework, introduced by [4], presents an interdisciplinary approach by integrating safety engineering, human-computer interaction, legal frameworks, and social acceptance. It deals with issues such as hardware fault tolerance and fail-operational autonomy, which are critical for ensuring safety in conventional road traffic and high-speed racing environments. Similarly, [5] proposed a safety-focused framework combining a finite state machine with nonlinear model predictive control (MPC), a widely utilized technique for autonomous vehicle maneuvers.

In the context of trajectory planning, [6] introduces an MPC-based framework for autonomous overtaking, integrating a graph-based route and velocity optimization to minimize collision risks in multi-vehicle scenarios. Furthermore, [7] compares three model-based lateral controllers for autonomous vehicles, emphasizing passenger comfort while maintaining accurate trajectory tracking.

Various trajectory generation methods have also been explored. For instance, [8] presents a trajectory generation algorithm for autonomous vehicles using quartic Bézier curves, generating curvature and velocity profiles separately to ensure curvature continuity, boundedness, and adherence to acceleration and sideslip constraints. Similarly, [9] introduced a B-spline trajectory optimization method tailored for autonomous racing, emphasizing curvature minimization. On the other hand, [10] proposes an artificial potential function (APF) framework for local path planning, addressing issues like non-reachable goals, obstacle collisions, and narrow passage problems.

Reinforcement learning (RL) is another advanced methodology in autonomous driving. As introduced in [11], end-to-end planning leverages deep learning to map sensory inputs directly to control commands, improving adaptability and simplifying traditional pipelines. Similarly, [12] developed a feed-forward neural network trained on racing line data from optimal control simulations across various circuits, enabling accurate and computationally efficient racing line predictions. Another study, [13] combined a single RL agent for geometric trajectory generation with a model-predictive controller for lateral control. Additionally, [14] focused on accelerating RL agent training to enhance its practicality and real-world applicability, demonstrating its potential to reduce training time while maintaining robust performance.

Dynamic trajectory planning is a critical task in many aspects of autonomous driving, requiring the generation of optimal paths and speed profiles in real-time. [15] highlights the importance of balancing objectives such as safety, smoothness, comfort, and consistency, often achieved by leveraging the Frenet coordinate system to simplify the problem of trajectory representation relative to the road geometry. Similarly, [16] introduces the concept of state-time space, a novel framework for dynamic trajectory planning that integrates dynamic constraints and moving obstacles into a unified representation. Accurate environmental understanding is vital for this task, and SLAM-based mapping techniques discussed in [17] provide robust and adaptable map representations that complement the use of the Frenet coordinate system, allowing autonomous systems to navigate safely and efficiently in complex environments.

III. Contributions of the paper

In this paper, we present a robust global path-planning framework that generates minimum curvature trajectories through quadratic programming-based optimization. The solution is built on a multi-component simulation environment that includes a path generator, a nonlinear, planar single-lane vehicle model with a dynamic wheel model, a lateral MPC controller to minimize tracking errors, and a longitudinal PID controller based on a predefined velocity profile. During the velocity profile generation, we rely on the vehicle's dynamic model, considering normal forces, the Pacejka tire model, and other factors to define an optimal yet safe speed range. Our simulation experiments show that the developed approach results in smoother and more stable vehicle dynamics, with fewer extreme steering variations, yaw rate fluctuations, and errors.

IV. Environment

A unique simulation environment comprising several essential components was developed for designing and testing the local trajectory planning method. The first is a road generator that adheres to standard road design guidelines, constructing tracks using straight segments, curves, and clothoid transitions. The second component is a nonlinear planar single-track vehicle model, which includes a dynamic wheel model, as detailed in [18]. This model is utilized for the EGO vehicle to ensure accurate behavior prediction. A lateral Model Predictive Control (MPC) system is the third component, relying on a linear dynamic model to minimize tracking errors and incorporating a yaw-rate profile constraint. Complementing this is a longitudinal PID controller, which uses a predefined speed profile generated by the local planner to regulate speed. The system also integrates transformation solutions to facilitate planning within the Frenet frame. Finally, for the visual representation of the environment, a 2D graphical interface has been provided.

V. Global trajectory planning

Trajectory planning plays a crucial role in autonomous vehicle navigation, particularly in high-speed scenarios such as autonomous racing [19], where the primary objective is to achieve the shortest possible lap time. The major challenge is to generate a smooth, feasible, and dynamically valid trajectory that minimizes curvature while ensuring safety. The method involves formulating the problem as a quadratic programming (QP) task [20], where the control points of the trajectory are optimized to achieve smoothness and feasibility. The approach leverages the differences between the inner and outer boundaries of the track to define curvature and smoothness constraints, ensuring that the resulting path adheres to the track limits. The execution time of planning the global trajectory implemented in Python is 0.05314 seconds for a 500-meter section when running on a laptop equipped with a Ryzen 5600H processor.

1. Quadratic Cost Function

The quadratic cost function for the optimization problem is given as:

$$\tau^\top \mathbf{H}_C \tau + \mathbf{B}_C^\top \tau \tag{1}$$

where:

- H_C: Quadratic cost matrix representing the curvature and geometric properties of the path,
- B_C: Linear cost vector incorporating path smoothness and edge deviations,
- τ: Optimization variables representing the proportional position between the inner and outer boundaries.

2. Curvature and Smoothness Terms (H_C)

The curvature and smoothness terms are computed based on the differences between the inner and outer path boundaries. For this, the matrices h_x and h_y represent the curvature costs in the x and y directions, respectively. They are constructed using second-order finite difference approximations over three consecutive points $(i-1, i, i+1)$:

$$\mathbf{h_x} = \begin{bmatrix} dx_{m1}^2 & -2dx_{m1}dx & dx_{m1}dx_{p1} \\ -2dxdx_{m1} & 4dx^2 & -2dxdx_{p1} \\ dx_{p1}dx_{m1} & -2dx_{p1}dx & dx_{p1}^2 \end{bmatrix}, \quad (2)$$

$$\mathbf{h_y} = \begin{bmatrix} dy_{m1}^2 & -2dy_{m1}dy & dy_{m1}dy_{p1} \\ -2dydy_{m1} & 4dy^2 & -2dydy_{p1} \\ dy_{p1}dy_{m1} & -2dy_{p1}dy & dy_{p1}^2 \end{bmatrix}. \quad (3)$$

Here, h_x is derived from the horizontal displacements dx between the inner and outer track boundaries, while h_y is constructed similarly from the vertical displacements dy. These matrices contribute to the total curvature cost used in the trajectory optimization process.

The total curvature matrix is:

$$\mathbf{H}_C[i-1:i+2, i-1:i+2] += \mathbf{h}_x + \mathbf{h}_y. \quad (4)$$

3. Smoothness Correction (B_C)

The vectors b_x and b_y represent the second finite difference of the inner track coordinates, scaled by a weighted product with the corresponding dx and dy offsets. These expressions account for local curvature variations.

Adding these terms gives:

$$\mathbf{B}_C[i-1:i+2] += \mathbf{b}_x + \mathbf{b}_y. \quad (5)$$

4. Path Optimization with Constraints

The optimization is subject to the following box constraints:

$$0.05 \leq \tau_i \leq 0.95 \quad \forall i, \quad (6)$$

which can be reformulated in matrix form as:

$$\mathbf{G}\tau \leq \mathbf{h}, \quad \text{where} \quad \mathbf{G} = \begin{bmatrix} I \\ -I \end{bmatrix}, \quad \mathbf{h} = \begin{bmatrix} ub \\ -lb \end{bmatrix} \quad (7)$$

Where, $I \in \mathbb{R}^{N \times N}$ is the identity matrix, and the vectors ub and lb represent the element-wise upper and lower bounds for the optimization variable τ. This formulation ensures that the optimized trajectory remains within the track boundaries, maintaining a predefined safety margin from the inner and outer edges.

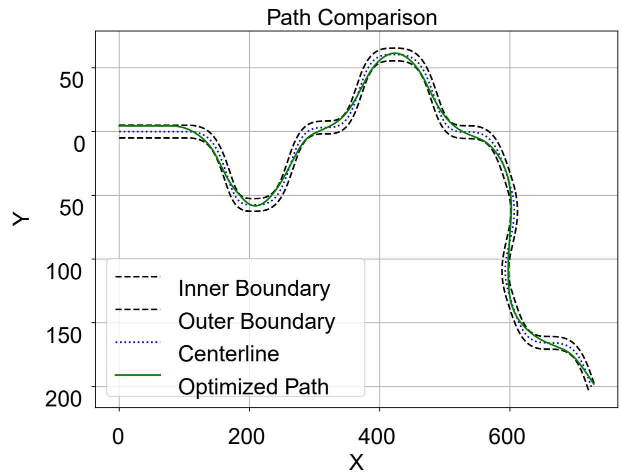

Fig. 1: Optimized path

5. Optimized Path

Finally, the optimized path as seen in Fig.1 is computed as:

$$\mathbf{P}_x = \mathbf{X}_{\text{inner}} + \boldsymbol{\tau}_{\text{curv}} \cdot (\mathbf{X}_{\text{outer}} - \mathbf{X}_{\text{inner}}), \quad (8)$$

$$\mathbf{P}_y = \mathbf{Y}_{\text{inner}} + \boldsymbol{\tau}_{\text{curv}} \cdot (\mathbf{Y}_{\text{outer}} - \mathbf{Y}_{\text{inner}}). \quad (9)$$

Here, $\tau_{\text{curv}} \in \mathbb{R}^N$ represents the optimized vector of lateral weights that define the position of each trajectory point between the inner and outer track boundaries. As seen in Fig. 2, the overall curvature of the optimized path has been significantly reduced. The extreme values have decreased, and fewer peaks are reached than the centerline curvature. Moreover, the rate of curvature change has also been noticeably reduced.

This reduction implies that the required steering angle inputs will be less aggressive and more gradual, leading to a smoother vehicle motion. Additionally, the constant sections in the centerline curvature can be attributed to the fact that our track generator consists of straight segments connected by clothoid transition curves for cornering.

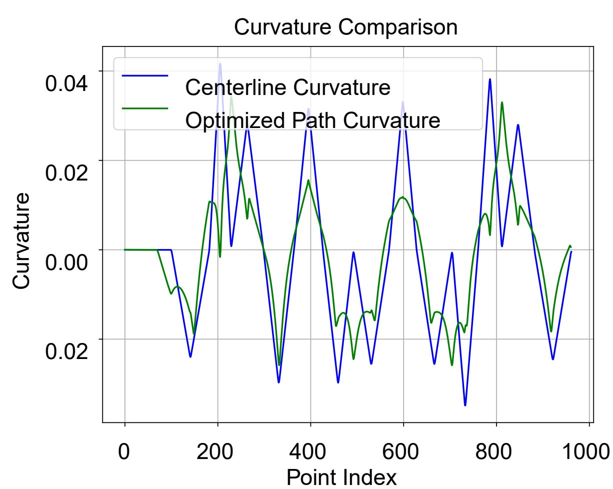

Fig. 2: Curvature comparison

VI. Velocity profile based on vehicle model

The vehicle longitudinal velocity profile is determined along the entire length of the track based on the vehicle model. The speed profile is optimized to maximize lateral acceleration, which includes the maximization of the yaw rate. First, the normal forces acting on the front and rear axles are calculated based on equilibrium conditions while considering the known track curvature.

The normal forces at the front and rear axles are computed using the vehicle's mass, gravitational acceleration, and the distances from the center of gravity to each axle. These forces are then used to calculate the peak factors for both the front and rear axles based on the tire-road friction coefficient.

$$F_{z,\text{front}} = \frac{mgL_r}{L} \qquad F_{z,\text{rear}} = \frac{mgL_f}{L} \tag{10}$$

where m is the vehicle mass, L_f is the distance from the front axle to the center of gravity, L_r is the distance from the rear, and L is the wheelbase.

A. Determination of Traction Forces

The longitudinal and lateral forces, based on the Pacejka tire model [21], are calculated at each step considering the slip ratio. From this, the maximum longitudinal force is derived, which is used to determine the vehicle's maximum acceleration and deceleration. The traction forces are computed using the Pacejka tire model, where λ represents the slip ratio, α the slip angle, and B, C, D, E are the Pacejka parameters.

$$F_x = D \sin\left(C \tan^{-1}(B\lambda - E(B\lambda - \tan^{-1} B\lambda))\right) \tag{11}$$

$$F_y = D \sin\left(C \tan^{-1}(B\alpha - E(B\alpha - \tan^{-1} B\alpha))\right) \tag{12}$$

The peak forces required for the Pacejka formula are based on the equations previously defined in Equation (10).

B. Vehicle Dynamic Limits

In addition, the vehicle's speed is regulated based on the maximum yaw rate and lateral acceleration, ensuring that the forces acting on the tires remain within the physical limits of the vehicle model. The track curvature is taken into account in these calculations, enforcing a speed profile that aligns with both the track geometry and the vehicle's dynamic capabilities.

The maximum yaw rate of the vehicle is defined as:

$$r_{\max} = \frac{F_{y,\text{rear}} \cdot L}{I_z} \tag{13}$$

The maximum velocity based on the yaw rate is computed as:

$$v_{\text{yaw}} = \frac{r_{\max}}{|\kappa|} \tag{14}$$

where κ is the track curvature and I_z is the vehicle moment of inertia.

However, this would have allowed significantly higher speeds, which are not feasible. Therefore, we chose to maximize lateral acceleration instead. The lateral acceleration limit is given by:

$$a_{y,\max} = \mu g \tag{15}$$

The velocity limit based on lateral acceleration is computed as:

$$v_{ay} = \sqrt{\frac{a_{y,\max}}{|\kappa|}} \tag{16}$$

C. Final Velocity

The optimal speed profile is calculated through an iterative process based on dynamic programming that progresses backward and forward while leveraging previously computed values to refine the next set of calculations.

In the first iteration, an initial maximum speed profile is determined at each path point, considering physical and dynamic constraints:

$$\mathbf{v}_{\max} = \min(\mathbf{v}_{\text{ay}}, \mathbf{v}_{\text{yaw}}) \tag{17}$$

In the second step, the braking limit is incorporated through a backward iteration, considering the deceleration limits. Next, the acceleration limit is taken into account through a forward iteration.

To obtain this, the maximum acceleration and deceleration are determined using the following calculations.

$$\mathbf{a}_{x,\max} = \frac{\mathbf{F}_{x,\max}}{m}, \qquad \mathbf{a}_{x,\min} = \frac{-\mathbf{F}_{x,\max}}{m} \tag{18}$$

The iterative update of the velocity is given by:

$$\mathbf{v}_i^{\text{sat}} = \min\left(\mathbf{v}_i, \sqrt{\mathbf{v}_{i-1}^2 + 2\mathbf{a}_x \cdot \Delta s}\right) \tag{19}$$

Here, v_i^{sat} is the updated velocity at path point i, computed by taking the minimum of the previously established velocity limit v_i and the kinematic constraint based on the maximum allowed acceleration or deceleration. The term v_{i-1} is the velocity at the previous path point, a_x represents the maximum longitudinal acceleration or deceleration (depending on the iteration direction), and Δs is the distance between consecutive path points. During a forward iteration, a_x is positive and increases the velocity, while in a backward iteration, a_x takes a negative value, effectively reducing the speed. This update ensures that the velocity profile respects the vehicle's physical acceleration capabilities throughout the path.

VII. Results

As a result of this study, we present a quadratic optimization-based trajectory planning framework that generates a minimum-curvature path while ensuring a dynamically feasible velocity profile determined by dynamic programming. The entire system follows Paden's architectural guidelines [3].

The core objective of the global trajectory planner is to produce a path that minimizes curvature, which inherently reduces excessive lateral accelerations and improves vehicle stability. However, a geometrically optimal trajectory alone is insufficient; it must be complemented by a velocity profile that accounts for dynamic feasibility, considering acceleration, deceleration, and tire force constraints. By integrating these two aspects, curvature minimization and dynamic feasibility, the planner generates a trajectory that is smooth and physically realizable.

This framework is best formulated within the Frenet coordinate system, where the local trajectory is defined relative to the global reference path rather than in absolute coordinates. This representation allows the local trajectory planner to make adjustments efficiently while maintaining alignment with the global plan. If a near-optimal global trajectory has already been computed, it does not need to be recomputed entirely; instead, only minor updates are necessary to refine it based on real-time constraints and environmental changes. This hierarchical approach ensures that the trajectory remains both globally optimized and locally adaptable.

The validation of the results was performed in a simulation environment, where we evaluated the efficiency of the system on different routes by analyzing vehicle dynamics parameters, including acceleration characteristics and slip behavior.

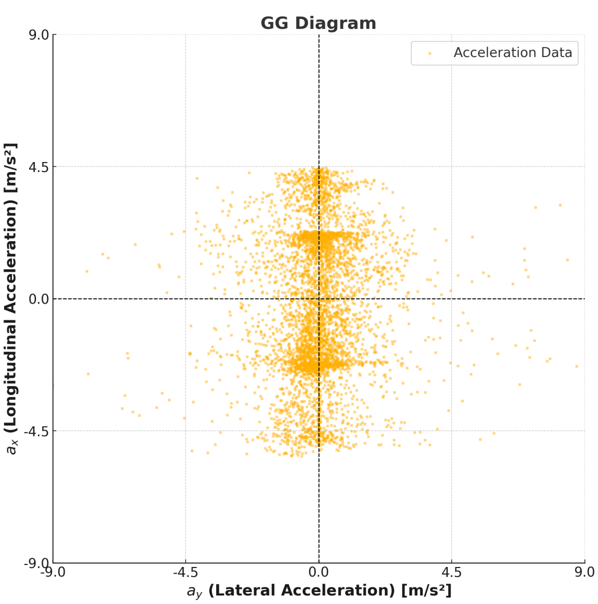

Fig. 3: GG diagram

The GG diagram (Fig. 3) illustrates the lateral and longitudinal acceleration distribution during the simulated runs. The results demonstrate that the acceleration limits were respected, ensuring vehicle stability. The concentration of data points along the vertical axis proves that the optimization process prioritized maintaining high longitudinal acceleration while minimizing excessive lateral acceleration, which is because of the general road vehicle model. Fig. 4 presents the slip values for multiple runs, highlighting the variations in vehicle grip performance across different scenarios. The results indicate that the slip remains within acceptable limits, with minor deviations observed in some runs. The vehicle maintained stable slip values, confirming its robustness.

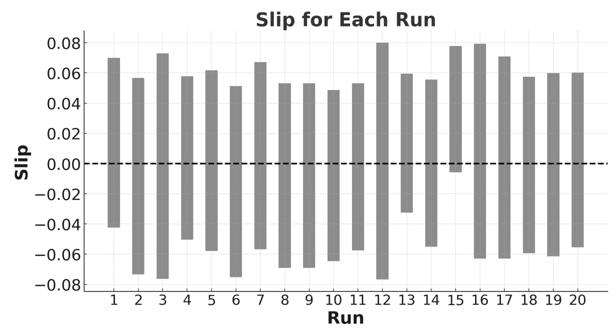

Fig. 4: Slip diagram

VIII. Conclusions

In conclusion, this paper presents a comprehensive global path-planning framework that optimizes trajectories with minimal curvature using quadratic programming. However, achieving the fastest lap times requires a strategic approach that goes beyond simply minimizing curvature. The optimal racing line is defined not just by the smoothest path but by the trajectory that allows the shortest completion time. This means that, in many cases, a late apex can be advantageous, as it enables earlier throttle application and maximizes acceleration out of a turn. Additionally, there is a constant trade-off between the length of the driven path and lateral acceleration. Although a wider arc might reduce lateral forces, it can also result in excessive travel distance, making a more aggressive, tighter line the faster option in certain scenarios. Through local trajectory planning, our goal is to further refine these principles and implement them dynamically, allowing real-time optimization based on environmental and vehicle constraints. Furthermore, efficient tire grip utilization is crucial, as a race car operates at the limits of adhesion, with the g-g diagram serving as a key tool for analyzing tire performance. However, in this case, our focus was on pushing the limits of a general road vehicle, assessing how these strategies can be adapted to real-world driving conditions, and ensuring both high performance and safety in practical applications. The velocity profile is generated based on the vehicle's dynamics, incorporating factors like normal forces and the Pacejka tire model to ensure optimal and safe speed adaptation. The simulation results demonstrated improved vehicle stability, smooth steering behavior, and reduced yaw rate fluctuations.

ACKNOWLEDGMENT

The research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems. (RRF-2.3.1-21-2022-00002)

REFERENCES

[1] J. H. Reif, "Complexity of the mover's problem and generalizations," in *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*. IEEE Computer Society, 1979, pp. 421–427.

[2] A. Rodriguez, "Rc car modelling and trajectory tracking control," *University of Southampton, September 2021*, 2021.

[3] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on intelligent vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[4] P. Koopman and M. Wagner, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 90–96, 2017.

[5] J. Palatti, A. Aksjonov, G. Alcan, and V. Kyrki, "Planning for safe abortable overtaking maneuvers in autonomous driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 508–514.

[6] B. Németh, T. Hegedűs, and P. Gáspár, "Model predictive control design for overtaking maneuvers for multi-vehicle scenarios," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 744–749.

[7] A. M. Bokor, A. Szabo, S. Aradi, and L. Palkovics, "Comparison of lateral controllers for autonomous vehicles based on passenger comfort optimization," *21st International Conference on Informatics in Control, Automation and Robotics*, 2024.

[8] C. Chen, Y. He, C. Bu, J. Han, and X. Zhang, "Quartic bézier curve based trajectory generation for autonomous vehicles with curvature and velocity constraints," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6108–6113.

[9] H. Xue, T. Yue, and J. M. Dolan, "Spline-based minimum-curvature trajectory optimization for autonomous racing," *arXiv preprint arXiv:2309.09186*, 2023.

[10] D. H. Kim and S. Shin, "Local path planning using a new artificial potential function composition and its analytical design guidelines," *Advanced Robotics*, vol. 20, no. 1, pp. 115–135, 2006.

[11] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 187–210, 2018.

[12] S. Garlick and A. Bradley, "Real-time optimal trajectory planning for autonomous vehicles and lap time simulation using machine learning," *Vehicle System Dynamics*, vol. 60, no. 12, pp. 4269–4289, 2022.

[13] Á. Fehér, Á. Domina, Á. Bárdos, S. Aradi, and T. Bécsi, "Path planning via reinforcement learning with closed-loop motion control and field tests," *Engineering Applications of Artificial Intelligence*, vol. 142, p. 109870, 2025.

[14] B. Kővári, B. Pelenczei, I. G. Knáb, and T. Bécsi, "Beyond trial and error: Lane keeping with monte carlo tree search-driven optimization of reinforcement learning," *Electronics*, vol. 13, no. 11, p. 2058, 2024.

[15] M. Wang, L. Zhang, Z. Wang, Y. Sai, and Y. Chu, "A real-time dynamic trajectory planning for autonomous driving vehicles," in *3rd Conference on Vehicle Control and Intelligence, CVCI 2019*, 2019.

[16] T. Fraichard, "Dynamic trajectory planning with dynamic constraints: A'state-time space'approach," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, vol. 2. IEEE, 1993, pp. 1393–1400.

[17] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[18] T. Hegedűs, B. Németh, and P. Gáspár, "Design of a low-complexity graph-based motion-planning algorithm for autonomous vehicles," *Applied Sciences*, vol. 10, no. 21, p. 7716, 2020.

[19] F. Braghin, F. Cheli, S. Melzi, and E. Sabbioni, "Race driver model," *Computers & Structures*, vol. 86, no. 13-14, pp. 1503–1516, 2008.

[20] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, 2020.

[21] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.