

Real-time Mixed-Integer Quadratic Programming for Driving Behavior-Inspired Speed Bump Optimal Trajectory Planning

Van Nam Dinh, Dinh Tu Duong, Van Du Phan, The Anh Mai, Van Chuong Le, Thai Son Dang, Ho Sy Phuong, Hung Cuong Ta and Van Vy Phan

Abstract This paper proposes a novel methodology for trajectory planning in autonomous vehicles (AVs), addressing the complex challenge of negotiating speed bumps within a unified Mixed-Integer Quadratic Programming (MIQP) framework. By leveraging Model Predictive Control (MPC), we develop trajectories that optimize both the traversal of speed bumps and the overall passenger comfort. A key contribution of this work is the formulation of speed bump handling constraints that closely emulate human driving behavior, seamlessly integrating these with broader road navigation requirements. Through extensive simulations in varied urban driving environments, we demonstrate the efficacy of our approach, highlighting its ability to ensure smooth speed transitions over speed bumps while maintaining computational efficiency suitable for real-time deployment. The method's capability to handle both static road features and dynamic constraints with expert human driving represents a significant step forward in trajectory planning for urban autonomous driving applications.

1 Introduction

Autonomous vehicles (AVs) are poised to transform transportation, offering improvements in safety, efficiency, and accessibility [1]. Figure 1 illustrates the architecture of an autonomous driving system, which is structured hierarchically and comprises several core components that work together to facilitate safe and efficient navigation, particularly in scenarios such as speed reduction zones. Figure 1-(a) illustrates the traditional approach to autonomous driving, where the framework

School of Engineering and Technology
Vinh University, Vinh, Viet Nam,
e-mail: namdv@vinhuni.edu.vn, duongdinhtu@vinhuni.edu.vn, dupv@vinhuni.edu.vn,
theanh@vinhuni.edu.vn, chuonglv@vinhuni.edu.vn, sondt@vinhuni.edu.vn,
hophuong@vinhuni.edu.vn, cuongth@vinhuni.edu.vn, 215752021610071@vinhuni.edu.vn

is structured into sequential layers: Mapping and Localization, Decision Making, Motion Planning, Path Tracking and Control, and Actuator Control. Each module functions independently, relying on preceding layers for input, which can introduce delays and reduce adaptability in dynamic environments [2]. Additionally, recent works have made significant strides in localization techniques for enhancing perception and state estimation in autonomous systems [3, 4]. Cadena et al. [5] provides a comprehensive review of SLAM (Simultaneous Localization and Mapping) techniques, addressing both traditional methods and recent advancements. Furthermore, recent studies [6] introduce advanced learning-based solutions for state estimation applied to dynamic, real-world environments. These works are pivotal as they integrate machine learning with traditional state estimation, enabling the system to adapt to a variety of scenarios, including complex and rapidly changing urban environments. By integrating these works, we establish a robust basis for state estimation and perception, underscoring their collective impact on enabling autonomous vehicles to make accurate, real-time decisions.

This work focuses on the motion planning control for AVs [1]. Two major enhancements are introduced on the right side of the architecture, as shown in Fig. 1(b). The first is the MIQP-based Motion Planning module, which employs Mixed-Integer Quadratic Programming to generate optimal trajectories [7, 8]. This is particularly useful for navigating complex scenarios with discrete decisions, such as speed bumps or construction zones [9]. The second enhancement is a tracking-based NMPC module, utilizing Nonlinear Model Predictive Control to ensure robust tracking of the planned trajectory while adapting in real-time to changing conditions [10]. Integrating these advanced motion planning and control techniques with high-level decision-making allows the autonomous system to handle complex driving situations with greater safety and efficiency, ensuring smooth operation across diverse driving conditions.

Navigating speed bumps is a critical yet often underappreciated component of trajectory planning for autonomous vehicles (AVs), especially in urban settings [11]. Speed bumps are traffic-calming measures implemented to slow vehicles down and enhance safety in areas with significant pedestrian or cyclist activity. While human drivers manage these obstacles with relative ease, they pose substantial challenges for autonomous systems due to the need to simultaneously balance safety, ride comfort, and adherence to traffic regulations [11, 12]. One of the core challenges lies in decelerating sufficiently to minimize impact while maintaining efficient travel times. Human drivers, particularly those with less experience, may brake too abruptly or fail to accelerate smoothly afterward, resulting in jerky motions that compromise passenger comfort. In contrast, autonomous systems have the potential to optimize these transitions through real-time trajectory adjustments, considering both the vehicle's dynamics and the surrounding environment.

Trajectory planning for AVs has been a major area of research, with a focus on ensuring safe and efficient navigation in complex, dynamic environments. Various strategies have been proposed, ranging from graph-based methods to continuous optimization techniques, each tackling different facets of the problem [1]. Dolgov et al. [13] introduced the hybrid A* algorithm for path planning in semi-structured en-

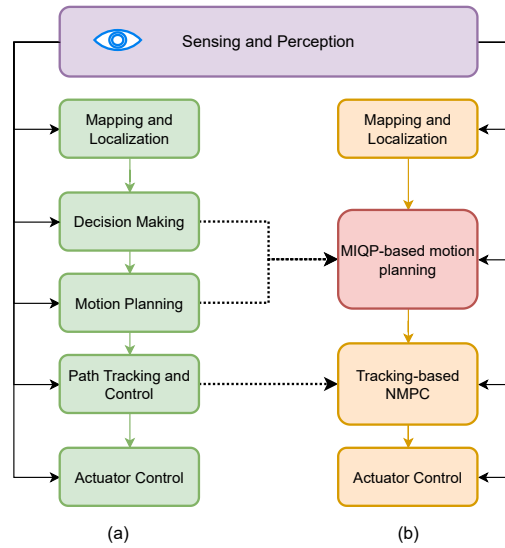


Fig. 1 The Hierarchical Autonomous Driving Framework. (a) presents the traditional solution for planning and control; (b) shows our adapted solution, incorporating MIQP-based motion planning for decision-making and motion planning.

vironments. Although this method handled scenarios with road boundaries and tight passages effectively, it did not explicitly account for dynamic obstacles or vehicle speed constraints, making it less suitable for navigating speed bumps or other irregular road conditions. Similarly, Ferguson et al. [14] developed an anytime D* search algorithm for urban driving, offering real-time path planning performance in environments with static obstacles. However, this approach did not address continuous dynamic constraints, limiting its ability to manage the speed modulation required for negotiating speed bumps.

More recently, Model Predictive Control (MPC) has gained attention as a robust tool for trajectory planning, owing to its capacity to incorporate complex vehicle dynamics and constraints into the optimization process. Li et al. [15] applied MPC to lane-change maneuvers, producing smooth, collision-free trajectories. Despite demonstrating the real-time effectiveness of MPC in trajectory generation, their work did not consider obstacle avoidance in dense environments or account for speed bump negotiation, which necessitates precise speed control to ensure passenger comfort.

Mixed-Integer Programming (MIP) has been widely employed to handle discrete decision-making in trajectory planning. Schouwenaars et al. [16] used Mixed-Integer Linear Programming (MILP) to address aircraft collision avoidance. While this offered a robust framework for handling discrete constraints, the challenges posed by ground vehicle navigation—such as road boundaries, speed limits, and speed bumps—were not considered. Adapting MIP to AV trajectory planning intro-

duces additional complexities, requiring the integration of both discrete obstacles (e.g., lane boundaries) and continuous variables (e.g., speed and position).

The issue of speed bump negotiation in autonomous driving has received limited attention. Handling speed bumps requires AVs to adjust their speed to ensure passenger comfort and protect the vehicle’s suspension. This task becomes more complicated when speed bumps are combined with other constraints, such as dynamic obstacles or road boundaries. A study by Navarro et al. [11] explored using MPC to manage speed bumps in a parking lot setting, but their approach focused on low-speed environments and did not address the challenges posed by speed bumps in more dynamic urban driving scenarios.

In this paper, we propose an advanced trajectory planning method incorporating speed bump negotiation into the larger problem of autonomous driving in urban environments. While most existing methods for AV trajectory planning focus primarily on obstacle avoidance and lane-keeping, we specifically address the unique challenges posed by speed bumps. Our approach uses a Model Predictive Control (MPC) framework to generate optimal trajectories that balance speed regulation, comfort, and safety. By formulating the problem as a Mixed-Integer Quadratic Programming (MIQP) optimization, we can seamlessly integrate speed bump handling into the broader task of urban navigation.

The contributions of this paper are as follows:

- A detailed analysis of vehicle dynamics related to speed bump traversal, highlighting the need for speed modulation to minimize discomfort and mechanical stress.
- A novel MIQP-based optimization framework that integrates speed bump negotiation into a unified trajectory planning solution for urban driving.
- Extensive simulations demonstrating the system’s ability to handle speed bumps smoothly while maintaining efficient, safe, and comfortable vehicle motion.

By focusing on speed bump negotiation within the context of AV trajectory planning, we address an important gap in the current literature. The proposed approach ensures that autonomous systems can handle the diverse range of road conditions found in urban environments, contributing to safer, more comfortable, and more efficient AV operations. The rest of this paper is organized as follows: Section 2 presents our MIQP formulation and vehicle model. Section 3 discusses our MPC-based trajectory generation method. Section V presents our experimental setup and results, followed by a conclusion in Section VI.

2 Methodology

This work assumes that the autonomous vehicle’s perception system can reliably detect speed bumps and provide accurate information about their positions within a predefined range. Specifically, the detection range is defined as $[d_{\min}, d_{\max}]$, where

d_{\min} and d_{\max} represent the minimum and maximum distances from the vehicle at which speed bumps can be perceived.

2.1 Vehicle Dynamics and Linearize Non-honolomic Constraints

2.1.1 Vehicle Dynamics

Generally, the vehicle's dynamic model is a non-linear system [2, 12], that can be represented as:

$$\dot{\mathbf{x}} = f(\mathbf{x}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \frac{v}{L} \tan(\sigma) \\ a \end{bmatrix} \quad (1)$$

where, x and y are the coordinates of the vehicle's position, θ is the heading angle of the vehicle; v is the linear velocity of the vehicle, σ is the steering angle, L is the wheelbase of the vehicle, a is the acceleration of the vehicle.

However, nonlinear systems are challenging to handle in MIQP-MPC. To enable real-time performance, we model the vehicle as a third-order linear point-mass system [9]. The state vector $\mathbf{x}(k)$ and control input $\mathbf{u}(k)$ at time step k are defined as:

$$\begin{aligned} \mathbf{x}(k) &= [x(k), v_x(k), a_x(k), y(k), v_y(k), a_y(k), \theta(k)]^\top \\ \mathbf{u}(k) &= [j_x(k), j_y(k)]^\top \end{aligned} \quad (2)$$

where (x, y) is the position, (v_x, v_y) are velocities, (a_x, a_y) are accelerations, θ is the heading angle, and (j_x, j_y) are jerks in the longitudinal and lateral directions, respectively.

The discrete-time dynamics are given by:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (3)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_d \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_d \end{bmatrix} \quad (4)$$

with

$$\mathbf{A}_d = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} \frac{1}{6}\Delta t^3 \\ \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix} \quad (5)$$

and Δt is the time step. The heading angle update can be approximated as:

$$\theta(k+1) = \theta(k) + \Delta t \cdot \frac{v_y(k)}{v_x(k)} \quad (6)$$

The vehicle dynamics are modeled using a discrete-time kinematic model:

$$\begin{cases} x(k+1) = x(k) + v_x(k)\Delta t + \frac{1}{2}a_x(k)\Delta t^2 + \frac{1}{6}j_x(k)\Delta t^3 \\ y(k+1) = y(k) + v_y(k)\Delta t + \frac{1}{2}a_y(k)\Delta t^2 + \frac{1}{6}j_y(k)\Delta t^3 \\ v_x(k+1) = v_x(k) + a_x(k)\Delta t + \frac{1}{2}j_x(k)\Delta t^2 \\ v_y(k+1) = v_y(k) + a_y(k)\Delta t + \frac{1}{2}j_y(k)\Delta t^2 \\ a_x(k+1) = a_x(k) + j_x(k)\Delta t \\ a_y(k+1) = a_y(k) + j_y(k)\Delta t \end{cases} \quad (7)$$

where Δt is the time step.

2.1.2 Linearize Non-holonomic Constraints

The lateral velocity $v_y(k)$ is related to the longitudinal velocity $v_x(k)$ and the steering angle θ by:

$$v_y = v_x \tan(\theta)$$

Given the steering angle constraints θ_{\min} and θ_{\max} , the lateral velocity $v_y(k)$ must satisfy:

$$v_y(k) \in [v_x(k) \tan(\theta_{\min}), v_x(k) \tan(\theta_{\max})]$$

The lateral acceleration $a_y(k)$ is the time derivative of the lateral velocity:

$$a_y = \frac{d}{dt}(v_x \tan(\theta)) \approx v_x \frac{d}{dt}(\tan(\theta))$$

Assuming ω represents the steering rate, we have:

$$a_y \approx v_x \omega$$

Therefore, the lateral acceleration constraint is:

$$a_y(k) \in [-v_x(k)\omega_{\max}, v_x(k)\omega_{\max}]$$

These constraints approximate the nonholonomic constraints of the vehicle by bounding the possible values of lateral velocity and acceleration based on the vehicle's longitudinal velocity and steering capabilities.

Therefore, the vehicle is subject to various constraints:

$$v_y(k) \in [v_x(k) \tan(\theta_{\min}), v_x(k) \tan(\theta_{\max})] \quad (8a)$$

$$a_y(k) \in [-v_x(k)\omega_{\max}, v_x(k)\omega_{\max}] \quad (8b)$$

where the last two constraints approximate the nonholonomic constraints of the vehicle.

2.2 MIQP Objective Function, Constraints, and Algorithm

We formulate trajectory planning for autonomous vehicles as a Mixed-Integer Quadratic Programming based-MPC (MIQP-MPC) [17]. This framework allows for the simultaneous modeling of the vehicle's continuous dynamics and the discrete decision-making processes involved in obstacle avoidance, lane changing, and handling complex road scenarios like speed bumps. The MIQP-MPC provides an effective means of ensuring optimal navigation under such conditions. The following provides a detailed explanation of each MIOCP formulation as follows,

$$\min_{\mathbf{x}, \mathbf{u}} \sum_{i=0}^N \frac{1}{2} \begin{bmatrix} \mathbf{x}(i) \\ \mathbf{u}(i) \end{bmatrix}^T \mathbf{H}(i) \begin{bmatrix} \mathbf{x}(i) \\ \mathbf{u}(i) \end{bmatrix} + \begin{bmatrix} \mathbf{q}(i) \\ \mathbf{r}(i) \end{bmatrix}^T \begin{bmatrix} \mathbf{x}(i) \\ \mathbf{u}(i) \end{bmatrix} \quad (\text{a})$$

$$\text{s.t. } \mathbf{x}(i+1) = [\mathbf{A}(i) \ \mathbf{B}(i)] \begin{bmatrix} \mathbf{x}(i) \\ \mathbf{u}(i) \end{bmatrix} + \mathbf{a}(i), \quad \forall i \in \mathbb{Z}_0^{N-1}, \quad (\text{b})$$

$$\underline{\mathbf{x}}(i) \leq \mathbf{x}(i) \leq \bar{\mathbf{x}}(i), \quad \mathbf{u}(i) \leq \mathbf{u}(i) \leq \bar{\mathbf{u}}(i), \quad \forall i \in \mathbb{Z}_0^N, \quad (\text{c})$$

$$\underline{\mathbf{c}}(i) \leq [\mathbf{C}(i) \ \mathbf{D}(i)] \begin{bmatrix} \mathbf{x}(i) \\ \mathbf{u}(i) \end{bmatrix} \leq \bar{\mathbf{c}}(i), \quad \forall i \in \mathbb{Z}_0^N, \quad (\text{d})$$

$$u_j(i) \in \mathbb{Z}, \quad \forall j \in I(i), \quad \forall i \in \mathbb{Z}_0^N. \quad (\text{e})$$

Finally, the MIOCP (9) is converted into the following MIQP formulation:

$$\min_{\mathbf{z}} \frac{1}{2} \mathbf{z}^T \mathbf{H} \mathbf{z} + \mathbf{h}^T \mathbf{z} \quad (\text{a})$$

$$\text{s.t. } \mathbf{G} \mathbf{z} \leq \mathbf{g}, \quad \mathbf{F} \mathbf{z} = \mathbf{f}, \quad (\text{a})$$

$$z_j \in \mathbb{Z}, \quad j \in \mathcal{I}, \quad (\text{a})$$

where \mathbf{z} includes all optimization variables, and the index set \mathcal{I} denotes the integer variables.

The adoption of Gurobi¹ for solving the MIQP allows our trajectory planning framework to manage the intricate balance between discrete decision-making and continuous optimization, ensuring real-time feasibility in the context of autonomous driving [18]. Its performance is especially beneficial for resource-constrained platforms, enabling our model predictive control (MPC) approach to handle complex dynamic environments while maintaining computational efficiency.

¹ <https://www.gurobi.com/>

2.2.1 MIQP Objective Function

The objective function for the optimal control problem over a prediction horizon of N steps is given by:

$$J = \sum_{k=0}^{N-1} \left(q_1(v_x(k) - v_r)^2 + q_2 a_x(k)^2 + q_3(y(k) - y_r)^2 + q_4 v_y(k)^2 + q_5 a_y(k)^2 + r_1 j_x(k)^2 + r_2 j_y(k)^2 \right) \quad (11)$$

where q_1, \dots, q_5 and r_1, r_2 are weighting coefficients, v_r is the reference velocity, and y_r is the reference lateral position. The state variables $v_x(k)$, $a_x(k)$, $y(k)$, $v_y(k)$, $a_y(k)$ represent the longitudinal and lateral velocities, accelerations, and positions of the system, while the control inputs $j_x(k)$ and $j_y(k)$ represent the longitudinal and lateral jerks at time step k .

To convert the objective function J into the standard Mixed-Integer Optimal Control Problem (MIOCP) format, we first define the decision vector $\mathbf{v}(k)$ at each time step k as:

$$\mathbf{v}(k) = [x(k) \ v_x(k) \ a_x(k) \ y(k) \ v_y(k) \ a_y(k) \ j_x(k) \ j_y(k)]^\top$$

Next, we express the quadratic terms in (11) in matrix form. The quadratic terms can be rewritten as:

$$J = \sum_{k=0}^{N-1} \mathbf{v}(k)^\top \mathbf{H} \mathbf{v}(k) + \mathbf{q}^\top \mathbf{v}(k)$$

where \mathbf{H} is the matrix containing the quadratic coefficients, and \mathbf{q} contains the linear terms arising from the reference velocity v_r and reference lateral position y_r . The matrix \mathbf{H} is defined as:

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 \end{bmatrix}$$

The vector \mathbf{q} is given by:

$$\mathbf{q} = \begin{bmatrix} 0 \\ -2q_1v_r \\ 0 \\ -2q_3y_r \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Finally, the constant terms (such as $q_1v_r^2$ and $q_3y_r^2$) are ignored since they do not affect the optimization problem. In summary, the objective function J has been successfully transformed into the quadratic form required for Mixed-Integer Optimal Control Problems, where each term can now be efficiently handled by an appropriate solver:

$$J = \sum_{k=0}^{N-1} \left[\mathbf{v}(k)^\top \mathbf{H} \mathbf{v}(k) + \mathbf{q}^\top \mathbf{v}(k) \right]$$

2.2.2 Speed Bump Negotiation Mimicking Human Driving Behavior Constraints

In general, the state and control input variables are bounded:

$$\begin{aligned} \mathbf{x}_{\min} &\leq \mathbf{x}(k) \leq \mathbf{x}_{\max} \\ \mathbf{u}_{\min} &\leq \mathbf{u}(k) \leq \mathbf{u}_{\max} \end{aligned} \tag{12}$$

When approaching a speed bump, human drivers typically exhibit certain behaviors to ensure a comfortable and safe passage. Our model aims to mimic these behaviors through a set of logical constraints. We introduce binary variables $\delta_1(k)$, $\delta_2(k)$, and $\delta_3(k)$ to represent the logical conditions related to the speed bump, and $\text{turn_left}(k)$, $\text{turn_right}(k)$, and $\text{is_turning}(k)$ to represent turning behavior. The following constraints implement the logical conditions for speed bump negotiation:

$$\delta_1(k) = 1 \Leftrightarrow x(k) \geq x_{\text{bump.start}} \quad (13a)$$

$$\delta_2(k) = 1 \Leftrightarrow x(k) \leq x_{\text{bump.end}} \quad (13b)$$

$$\delta_3(k) = 1 \Leftrightarrow v_x(k) \leq v_{\text{max.bump}} \quad (13c)$$

$$\text{turn.left}(k) = 1 \Leftrightarrow v_y(k) \geq v_{\text{turn}} \quad (13d)$$

$$\text{turn.right}(k) = 1 \Leftrightarrow v_y(k) \leq -v_{\text{turn}} \quad (13e)$$

$$\text{is.turning}(k) = \text{turn.left}(k) \vee \text{turn.right}(k) \quad (13f)$$

$$\delta_1(k) + \delta_2(k) - \text{is.turning}(k) \leq 1 \quad (13g)$$

$$-\delta_1(k) + \delta_3(k) \leq 0 \quad (13h)$$

$$-\delta_2(k) + \delta_3(k) \leq 0 \quad (13i)$$

$$\delta_1(k) + \delta_2(k) - \delta_3(k) \leq 1 \quad (13j)$$

where $x_{\text{bump.start}}$ and $x_{\text{bump.end}}$ are the start and end positions of the speed bump, $v_{\text{max.bump}}$ is the maximum allowed speed over the bump, and v_{turn} is the threshold lateral velocity for considering a turn.

By incorporating these human-like behaviors into the optimization model, we aim to generate trajectories that satisfy the mathematical constraints and produce a driving style that feels natural and comfortable to passengers, enhancing the overall experience of autonomous navigation over speed bumps.

2.2.3 MIQP-MPC Algorithm

The complete MIQP-MPC algorithm for trajectory planning is summarized as follows:

Algorithm 1 MIQP-MPC for Trajectory Planning

```

Initialize state  $\mathbf{x}(0)$ 
while not at goal do
    Formulate MIQP problem with objective function and constraints
    Solve MIQP to obtain optimal control sequence  $\mathbf{u}^*(k : k + N - 1)$ 
    Apply first control input  $\mathbf{u}^*(k)$ 
    Update state  $\mathbf{x}(k + 1)$  based on vehicle dynamics
     $k \leftarrow k + 1$ 
end while

```

3 Experimental Results

3.1 Experimental Setup

To evaluate the proposed method, we implemented the MIQP-MPC algorithm using the Gurobi optimizer. We simulated various driving scenarios on a straight road with a width of 5 meters, including multiple obstacles and a speed bump. The simulation parameters were set as follows:

Parameter	Value
Simulation Parameters	
Prediction horizon (T)	20 seconds
Time step (Δt)	0.1 seconds
Number of time steps (N)	200
Scenario Setup	
Road width	2.0 meters
Initial x-position (x_0)	0 m
Initial y-position (y_0)	0.75 m
Initial longitudinal velocity (v_{x0})	10 m/s
Initial lateral velocity (v_{y0})	0 m/s
Reference speed (v_r)	10 m/s
Speed bump start ($x_{\text{bump.start}}$)	30 m
Speed bump end ($x_{\text{bump.end}}$)	35 m
Maximum speed over bump ($v_{\text{max.bump}}$)	5 m/s
Wheelbase (L)	2.7 m
Lateral velocity threshold for turning (v_{turn})	0.1 m/s
Cost Function Weights	
q_1 (longitudinal velocity deviation)	1
q_2 (longitudinal acceleration)	1
q_3 (lateral position deviation)	1
q_4 (lateral velocity)	2
q_5 (lateral acceleration)	4
r_1 (longitudinal jerk)	4
r_2 (lateral jerk)	4

Table 1 Simulation Parameters, Scenario Setup, and Cost Function Weights

3.2 Results and Discussion

The experiments demonstrated the effectiveness of the proposed MIQP-MPC approach in generating safe and comfortable trajectories for autonomous vehicles in complex scenarios involving obstacles and speed bumps.

Fig. 2 illustrates the speed profile during speed bump negotiation. The vehicle consistently reduced its speed when approaching the speed bump, maintaining a

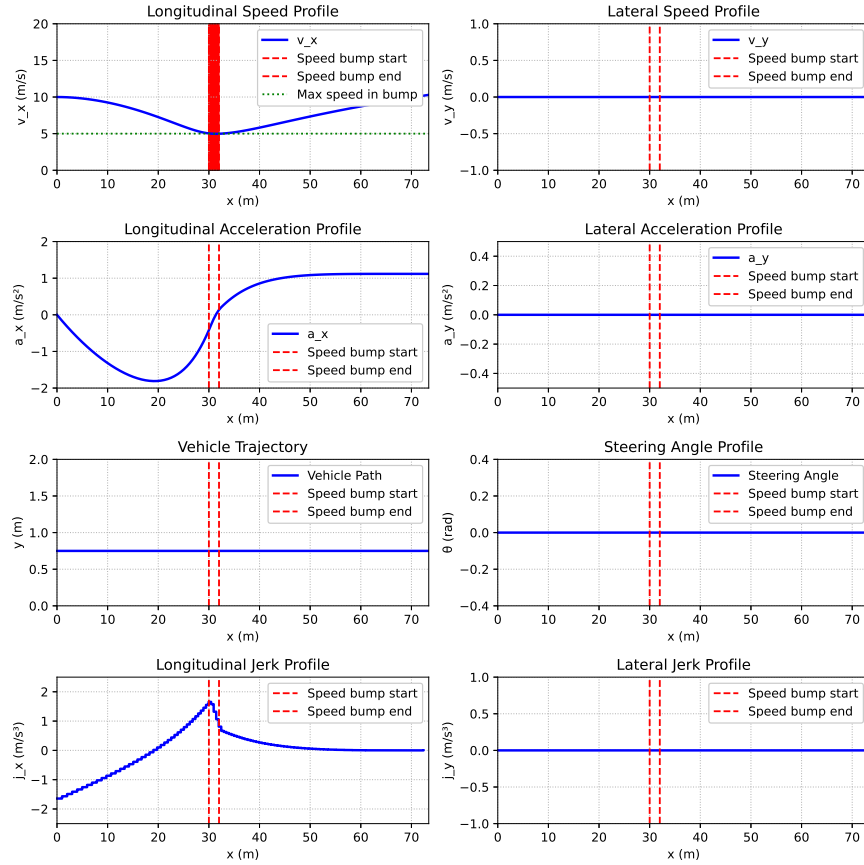


Fig. 2 The result with bumper width of 2m- no steering to reduce jerk

speed below the specified limit of 5 m/s while traversing the bump. Note that, in this experiment, we do not mimic human driving, and it goes straight to pass the bumper.

The subsequent experiment incorporates expert driving behavior, as in fig 3. At this time, the system steers the vehicle when it reaches the bumper.

The generated trajectories exhibited smooth changes in position, velocity, and acceleration, minimizing jerk and ensuring passenger comfort. Fig. 4 shows the acceleration and jerk profiles for a typical trajectory.

Last but not least, we conducted a study on the high speed before going to the bumper, the system can easily handle the problem and after that track the lateral lane, as shown in Fig. 5

Despite the computational complexity of MIQP and hardware, our implementation achieves solve times compatible with real-time operation in the tested scenarios. The computational time is about 10ms-100ms and was conducted on a laptop Intel

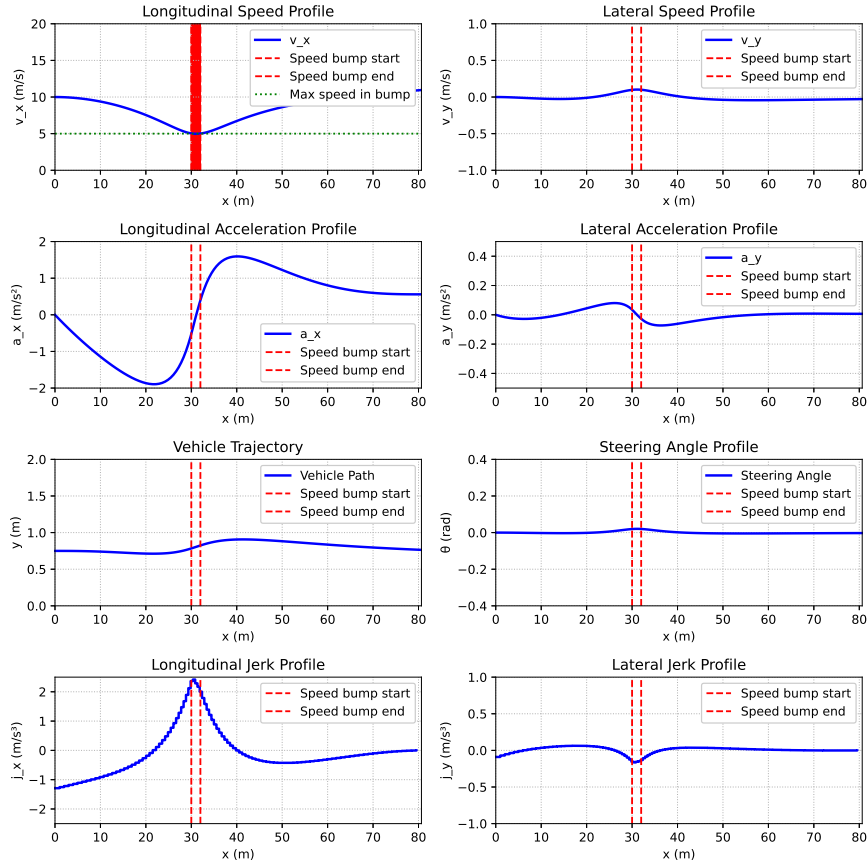


Fig. 3 The result with human behavior steering

core i5. However, the current formulation assumes perfect knowledge of obstacle positions and road conditions. Incorporating uncertainty estimation and robust optimization techniques could enhance the method’s performance in real-world conditions.

4 Conclusion

We propose a new approach to trajectory planning for autonomous vehicles. This approach incorporates speed bump negotiation using Mixed-Integer Quadratic Programming (MIQP) within a Model Predictive Control (MPC) framework. This method can produce safe, comfortable, and efficient trajectories, particularly in complex urban environments. This work contributes to a unified MIQP formulation that

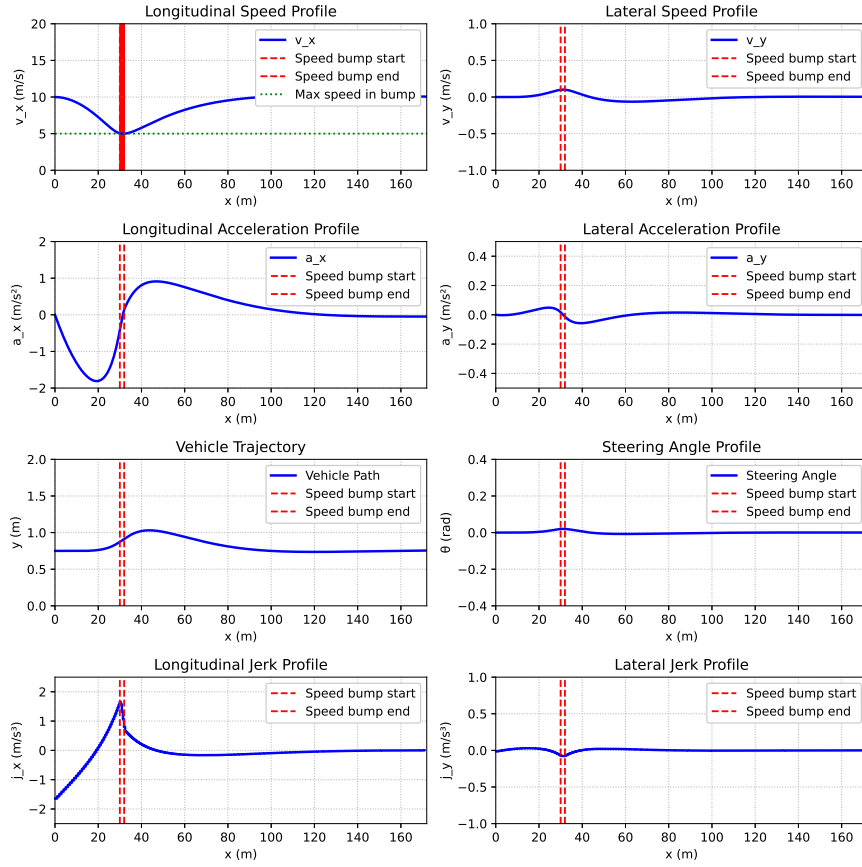


Fig. 4 The result with bumper 2m, longer run to converge to reference speed

simultaneously addresses speed bump negotiation, an MPC-based strategy that generates smooth trajectories while adhering to multiple constraints, and experimental validation demonstrating the method's effectiveness in simulated urban driving conditions. The results show that the proposed approach navigates complex environments with speed bumps while preserving both passenger comfort and computational efficiency. The ability to handle various constraints within a single optimization framework marks a significant step forward in autonomous vehicle trajectory planning. In future research, we aim to expand this method to encompass a wider range of urban driving scenarios, including interactions with obstacles, pedestrians, and other vehicles, thereby narrowing the gap between controlled simulations and real-world autonomous driving applications.

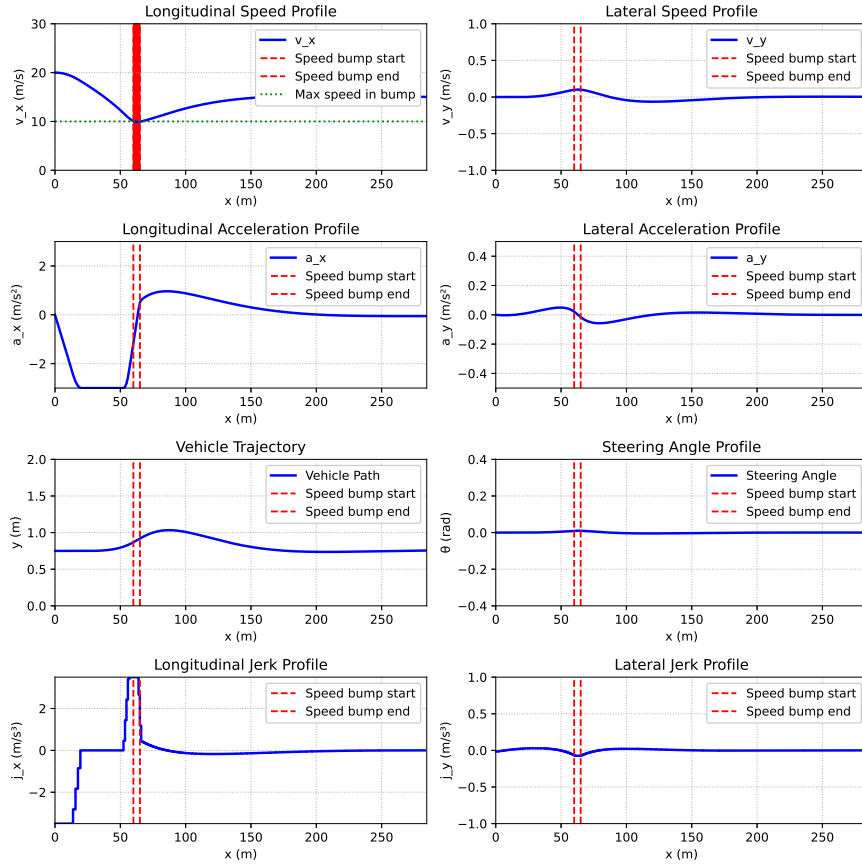


Fig. 5 The result handling high-speed over bumper

References

1. Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1(1):33–55, 2016.
2. Nam Dinh Van, Muhammad Sualeh, Dohyeong Kim, and Gon-Woo Kim. A hierarchical control system for autonomous driving towards urban challenges. *Applied Sciences*, 10(10):3543, 2020.
3. Timothy D Barfoot. *State estimation for robotics*. Cambridge University Press, 2024.
4. Dinh Van Nam, Phan Thanh Danh, Chung Huyk Park, and Gon-Woo Kim. Fusion consistency for industrial robot navigation: An integrated slam framework with multiple 2d lidar-visual-inertial sensors. *Computers and Electrical Engineering*, 120:109607, 2024.
5. Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

6. Dinh Van Nam and Kim Gon-Woo. Learning observation model for factor graph based-state estimation using intrinsic sensors. *IEEE Transactions on Automation Science and Engineering*, 20(3):2049–2062, 2022.
7. Jan Eilbrecht and Olaf Stursberg. Cooperative driving using a hierarchy of mixed-integer programming and tracking control. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 673–678. IEEE, 2017.
8. Yaoqiong Du, Yizhou Wang, and Ching-Yao Chan. Autonomous lane-change controller via mixed logical dynamical. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1154–1159. IEEE, 2014.
9. Xiangjun Qian, Florent Alché, Philipp Bender, Christoph Stiller, and Arnaud de La Fortelle. Optimal trajectory planning for autonomous driving integrating logical constraints: An miqp perspective. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 205–210, 2016.
10. Paolo Falcone, Francesco Borrelli, Jahan Asgari, Hongtei Eric Tseng, and Davor Hrovat. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on control systems technology*, 15(3):566–580, 2007.
11. J. Navarro and P. Luque. Real-time speed bump negotiation using model predictive control in low-speed conditions. In *2018 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2018.
12. J. Smith and A. Johnson. Challenges in urban autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 24(3):1000–1015, 2023.
13. Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Path planning for autonomous vehicles in unknown semi-structured environments. *The International Journal of Robotics Research*, 29(5):485–501, 2010.
14. Dave Ferguson, Thomas M Howard, and Maxim Likhachev. Motion planning in urban environments. In *Journal of Field Robotics*, volume 25, pages 939–960, 2008.
15. Shengbo Eben Li, Keqiang Li, and Jianqiang Wang. Model predictive multi-objective vehicular adaptive cruise control. *IEEE Transactions on Control Systems Technology*, 23(1):198–207, 2015.
16. Tom Schouwenaars, Bart De Moor, Eric Feron, and Jonathan How. Mixed integer programming for multi-vehicle path planning. *2001 European Control Conference (ECC)*, pages 2603–2608, 2001.
17. Rien Quirynen, Sleiman Safaoui, and Stefano Di Cairano. Real-time mixed-integer quadratic programming for vehicle decision-making and motion planning. *IEEE Transactions on Control Systems Technology*, 2024.
18. Calvin Nau, Prashant Sankaran, and Katie McConky. Comparison of parameter tuning strategies for team orienteering problem (top) solved with gurobi. In *IIE Annual Conference. Proceedings*, pages 1–6. Institute of Industrial and Systems Engineers (IISE), 2022.