

Trajectory Tracking for Autonomous Vehicles Using NMPC Method and Semantic Lane Inference

Thanh-Ha Vo ¹, Huu-Giang Nguyen ², and Hong-Quang Nguyen ^{3,*}

¹ Faculty of Electrical and Electronic Engineering, University of Transport and Communications, Hanoi, Vietnam

² Department of Electronic Engineering, Vietnam-Japan Center, Hanoi University of Industry, Hanoi, Vietnam

³ Faculty of Mechanical, Electrical, Electronics Technology, Thai Nguyen University of Technology, Thai Nguyen City, Vietnam

Email: vothanhha.ktd@utc.edu.vn (T.-H.V.); nhgianghau@gmail.com (H.-G.N.); quang.nguyenhong@tnut.edu.vn (H.-Q.N.)

*Corresponding author

Abstract—This paper presents a hybrid control framework for autonomous vehicles, combining semantic lane detection with a two-tier control approach: Nonlinear Model Predictive Control (NMPC) for lateral trajectory tracking and Fuzzy Proportional–Integral–Derivative (Fuzzy PID) control for longitudinal velocity management. Real-time visual data from the UltraFast segmentation network is integrated into the NMPC optimization, improving road boundary tracking in dynamic conditions. The fuzzy PID controller is optimally tuned and enhanced with a feedforward compensation branch to anticipate velocity changes, speeding up convergence while ensuring stability. Simulations across various velocity targets demonstrate rapid convergence, lateral stability, and reduced control effort. Compared to classical methods like Linear Quadratic Regulator (LQR) and Pure Pursuit (PP), the proposed system achieves superior tracking accuracy, robustness, and smoother control. Key contributions include incorporating UltraFast-based Lane segmentation into NMPC and using feedforward-enhanced Fuzzy PID for better speed regulation, offering a scalable and adaptive solution for intelligent vehicle control in structured settings.

Keywords—vehicle control, Nonlinear Model Predictive Control (NMPC), Fuzzy Proportional Integral Derivative (Fuzzy PID), vision-based navigate

I. INTRODUCTION

Safe and efficient navigation is a key goal in autonomous driving, and lane determination is essential for maintaining lateral stability and staying within road boundaries. In real-world scenarios, traditional lane-following algorithms, which rely on edge detection or handcrafted features, often falter due to occlusions, faded markings, or varying lighting conditions. These challenges highlight the need for robust perception-control integration in dynamic environments to ensure reliability.

To address these issues, this study proposes a hybrid control-perception framework that integrates a semantic.

Segmentation-based vision module with a Nonlinear Model Predictive Control (NMPC) system. This approach enables the system to adapt to varying road conditions by embedding real-time visual data into the control loop, ensuring more precise trajectory tracking and improved speed regulation. This study introduces an integrated framework for autonomous lane tracking, combining efficient semantic segmentation with a NMPC.

Key contributions of the study:

- Seamless integration of perception and control: Embeds semantic lane detection (UltraFast) within the NMPC optimization, enabling real-time adaptation to road geometry in dynamic conditions.
- Hybrid control strategy: Combines NMPC for lateral trajectory tracking with a fuzzy logic-based Proportional Integral Derivative (PID) controller for longitudinal speed regulation, improving path-following accuracy and ride comfort.
- Improved responsiveness: Optimizes the fuzzy PID controller with tuned gain parameters and a feedforward term, achieving faster speed convergence without sacrificing stability.
- Real-time viability: Demonstrates lightweight and efficient simulation performance, suitable for deployment on embedded hardware in real-time autonomous systems.
- Superior performance: Outperforms standard methods like Linear Quadratic Regulator (LQR) and Pure Pursuit in lateral tracking accuracy (lower Root Mean Square Error (RMSE)), robustness (reduced overshoot), and computational efficiency. The system is validated through Python-based simulations under

low-to-moderate speed scenarios (5.0 m/s and 10.0 m/s), showing consistent lane-keeping, fast convergence to reference speeds, and improved tracking accuracy.

The paper is structured as follows: Section II reviews related work on perception-integrated and hybrid control strategies for autonomous vehicles. Section III presents the vehicle's kinematic and dynamic models as the basis for control design. Section IV describes the NMPC for lateral trajectory tracking. Section V explains the semantic lane detection framework using the UltraFast segmentation network. Section VI covers the simulation setup, performance evaluation, and benchmarking against traditional controllers. Section VII concludes with a summary and directions for future research.

II. RELATED WORK

Over the past decade, numerous control strategies have been investigated for autonomous vehicle navigation. Classical control methods such as PID controllers are widely used for their simplicity and real-time implementation capability [1]. However, PID controllers lack prediction capability and fail to manage constraints effectively, particularly in nonlinear dynamic environments [2].

To overcome these limitations, advanced model-based control methods like LQR [3] and Model Predictive Control (MPC) [4, 5] have gained popularity. While LQR optimizes control for linear systems, its performance deteriorates under nonlinear or time-varying dynamics. MPC, in contrast, anticipates future states over a prediction horizon and incorporates constraints, making it suitable for complex trajectory tracking, especially under real-world uncertainties [6, 7].

In parallel, lane detection has evolved from classical image processing techniques (e.g., Canny edge detection, Hough transforms) [8, 9] to deep learning-based semantic segmentation methods, such as ENet [10], SCNN [11] and Lane Net [12]. These models improve robustness to environmental variations but are often only used as perception front ends without integration into control loops.

Recent studies have attempted to bridge perception and control. End-to-end deep learning approaches [13] directly map camera input to control commands, but they lack transparency and struggle with constraint handling. Hybrid approaches integrate semantic segmentation with MPC or rule-based control [14–19], though sensitivity to visual noise remains a challenge. This work enhances robustness and interpretability by combining a deep learning-based lane detection algorithm with a constrained NMPC controller, enabling real-time trajectory optimization based on reliable semantic inputs. The proposed framework leverages a state-of-the-art lane detection network to extract semantically rich lane boundaries, which are input for a NMPC module. This combination ensures precise trajectory tracking and preserves the ability to handle system constraints effectively, even in dynamically varying and uncertain environments. By incorporating learnable perception components and

physics-based control in a tightly coupled manner, the framework addresses the limitations of standalone approaches in complex real-world scenarios.

The lane detection algorithm is first trained on diverse datasets to generalize across varying lighting and weather conditions, ensuring robustness to visual noise such as shadows, road glare, and occlusions. The output of this perception module is then processed to produce lane boundary parameters mapped into the vehicle's coordinate system. These parameters are fed into the NMPC formulation, where constraints—such as collision avoidance, curvature limitations, and actuator saturation—are embedded directly into the optimisation problem.

The NMPC controller solves a constrained optimisation problem at every time step, leveraging the predicted vehicle dynamics and observation-driven guidance to determine control inputs, such as steering angle and throttle. The prediction horizon allows the controller to anticipate and compensate for nonlinearity-induced deviations, ensuring smooth and safe navigation. Combining interpretable perception with constraint-driven control, this layered approach provides a reliable alternative to fully opaque end-to-end pipelines, offering enhanced accountability and adaptability in safety-critical applications.

III. LATERAL DYNAMIC MODEL OF AUTONOMOUS VEHICLE

A. Linear Bicycle Model

The linear dynamic model of the vehicle is derived based on a small-angle approximation and constant longitudinal velocity. Fig. 1 depicts the lateral dynamics model of the autonomous vehicle. It depicts the dynamic model of a car's motion with an axle, illustrating the primary forces affecting the vehicle. It considers the vehicle coordinate system, which represents the vertical and horizontal directions within the vehicle frame. In contrast, the OXY coordinate system denotes the vertical and horizontal directions in the absolute reference system. Here, ψ signifies the rotation angle of the vehicle body in the OXY reference system.

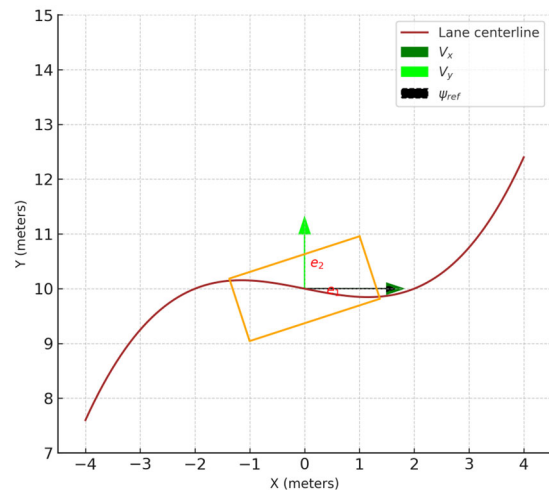


Fig 1. The lateral dynamics model of the autonomous vehicle.

By applying Newton's Law principle, the differential equations governing the car's motion in Fig. 1 can be derived as Eq. (1):

$$\begin{cases} m(\ddot{y} + V_x \dot{\psi}) = F_{yf} + F_{yr} \\ I_r \ddot{\psi} = l_f F_{xf} - l_r F_{xr} \end{cases} \quad (1)$$

where m and I_r are the vehicle mass and moment of inertia, respectively, l_f represent the vehicle's mass and moment of inertia, respectively, $F_{yf} + F_{yr}$ are the forces acting on the wheels in the x and y directions.

Empirical research shows that a tire's lateral force is directly proportional to its slide angle at moderate slip angles, a relationship known as "cornering stiffness". This property is crucial for vehicle handling and stability during cornering. Manufacturers carefully design tires to optimize cornering stiffness, balancing traction, durability, and rolling resistance to achieve desired performance. Understanding and fine-tuning this characteristic allows engineers to enhance a vehicle's handling dynamics for optimal performance.

The slip angle of the tire is written as Eq. (2):

$$\alpha_f = \delta - \theta_{vf} \quad (2)$$

where δ is the front tire steering angle?

The forces acting on the wheels in y directions for the rear and front tyre are calculated in Eq. (3).

$$\begin{cases} F_{yf} = 2C_{af}(\delta - \theta_{vf}) \\ F_{yr} = 2C_{ar}(-\theta_{vr}) \end{cases} \quad (3)$$

where C_{af}, C_{ar} are cornering stiffness.

And satisfy Eq. (4)

$$\begin{cases} \tan \theta_{vf} = \frac{v_y + l_f \dot{\psi}}{v_x} \\ \tan \theta_{vr} = \frac{v_y - l_r \dot{\psi}}{v_x} \end{cases} \quad (4)$$

If θ_{vf} & θ_{vr} are small, the equations in the figure describe the slip angles of the front and rear wheels in an autonomous vehicle's kinematic or dynamic model. Slip angles represent the deviation between the direction of wheel motion and the wheel heading, essential for analyzing lateral vehicle dynamics and designing advanced control systems. These are calculated by Eq. (5):

$$\begin{cases} \theta_{vf} = \frac{v_y + l_f \dot{\psi}}{v_x} \\ \theta_{vr} = \frac{v_y - l_r \dot{\psi}}{v_x} \end{cases} \quad (5)$$

The forces exerted on the rear and front tyres in the vertical direction are computed by Eq. (6).

$$\begin{cases} F_{yf} = 2C_{af}(\delta - \frac{v_y + l_f \dot{\psi}}{v_x}) \\ F_{yr} = 2C_{ar}(-\frac{v_y - l_r \dot{\psi}}{v_x}) \end{cases} \quad (6)$$

The dynamic model of the autonomous vehicle is rewritten as Eqs. (7) and (8):

$$\ddot{y} + V_x \dot{\psi} = \frac{2C_{af}\delta}{m} - \frac{2C_{af}(v_y + l_f \dot{\psi})}{mV_x} - \frac{2C_{ar}(\frac{v_y - l_r \dot{\psi}}{V_x})}{mV_x} \quad (7)$$

$$\ddot{y} = \frac{l_f}{l_r} (2C_{af}\delta - \frac{2C_{af}(v_y + l_f \dot{\psi})}{V_x}) + \frac{l_f}{l_r} \frac{2C_{ar}\delta(v_y + l_r \dot{\psi})}{V_x} \quad (8)$$

Eqs. (7) and (8) are rewritten as Eqs. (9) and (10):

$$\ddot{y} = \frac{2C_{af}\delta}{m} - \frac{2(C_{af} + C_{ar})}{mV_x} \dot{y} - (V_x + \frac{2(C_{af}l_f - C_{ar}l_r)}{mV_x})\dot{\psi} \quad (9)$$

$$\ddot{y} = \frac{l_f 2C_{af}\delta}{l_r} - \frac{2(C_{af}l_f - C_{ar}l_r)}{l_r V_x} \dot{y} - \frac{2(C_{af}l_f^2 - C_{ar}l_r^2)}{l_r V_x} \dot{\psi} \quad (10)$$

The dynamic state-space model of the autonomous vehicle is rewritten as Eq. (11)

$$\frac{d}{dt} \begin{pmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -2(C_{af} + C_{ar}) \\ 0 & 0 \\ 0 & -2(C_{af}l_f + C_{ar}l_r) \end{bmatrix} \begin{pmatrix} y \\ \dot{y} \\ \psi \\ \dot{\psi} \end{pmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{af}}{m} \\ 0 \\ \frac{2C_{ar}l_r}{l_r} \end{bmatrix} \delta \quad (11)$$

where: y : lateral displacement; ψ : heading angle; m : vehicle mass; I_z is a moment of inertia; l_f, l_r are distances from to center front and rear axles; C_{ar}, C_{af} : cornering stiffness coefficients; δ : steering angle input.

This model is widely used in control algorithms, such as linear MPC. However, this control method is simple and effective in low-speed scenarios.

B. Nonlinear Kinematic Model

The nonlinear model of the lateral dynamics of an autonomous vehicle is commonly based on the bicycle model, which approximates the car with a single front and rear wheel. This model captures the key dynamics, including longitudinal, lateral v_y , and yaw rate lateral position error e_1 (cross-track error) and heading error e_2 . The nonlinear equations of motion are given as Eq. (12):

$$\begin{cases} \dot{v}_x = \frac{1}{m}(F_{xf}\cos\delta - F_{yf}\sin\delta + F_{xr}) + v_y\omega \\ \dot{v}_y = \frac{1}{m}(F_{yf}\cos\delta + F_{xf}\sin\delta + F_{yr}) - v_x\omega \\ \dot{\omega} = \frac{1}{I_z}(l_f F_{yf}\cos\delta + l_f F_{xf}\sin\delta - l_r F_{yf}) \\ e_1 = v_y + v_x \sin(e_2) \\ e_2 = \omega - \frac{v_x \cos(e_2) - v_y \sin(e_1)}{R} \end{cases} \quad (12)$$

where F_{xf}, F_{xr} : Longitudinal forces are on the front and rear tire, and lateral forces are on the front and rear tyre.

This nonlinear model accurately describes vehicle behaviour at high speeds or during aggressive manoeuvres, where tyre slips occur, and nonlinearities become significant.

C. Vehicle Steering Angle and Vehicle Speed

Based on the research in Ref. [10], the relationship between steering angle and vehicle speed in a steering control system is analyzed in Fig. 2. The Y-axis represents the steering angle in degrees. At the same time, the X-axis corresponds to vehicle speed or time. The black dashed and dash-dot lines indicate the steering angle limits at each speed, and the blue dashed line shows the actual steering angle, which remains within these limits. At low speeds, the system allows larger steering angles of up to ± 45 degrees for better manoeuvre ability, while at higher speeds, the limits narrow to about ± 23 degrees to enhance stability and prevent rollover. The actual steering angle also demonstrates reduced amplitude and oscillations at higher speeds, reflecting smoother and more stable steering.

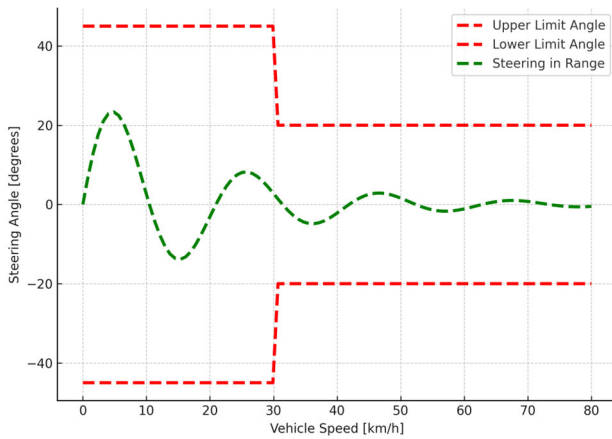


Fig. 2. Vehicle steering angle and vehicle speed.

The relationship between steering angle and vehicle speed is crucial in designing control systems for autonomous vehicles. Higher speeds require reduced steering angles to maintain stability and prevent rollover. Model Predictive Control (MPC) integrates this relationship as a dynamic constraint to ensure safe and feasible steering actions. MPC calculates optimal steering commands within speed-dependent limits, while a low-level PID controller accurately and quickly executes these commands. Lane boundaries are detected, and desired trajectories are generated via image processing modules, which feed into the MPC. Incorporating the speed-steering constraint across both MPC and PID layers ensures smooth, safe, and adaptive steering under varying conditions.

D. Model Selection Basis for NMPC controller

Selecting an appropriate vehicle model is crucial for the accuracy and performance of a NMPC. The two main approaches are the linearised vehicle model and the nonlinear bicycle model. Linear models, often used in embedded applications, provide computational efficiency near a fixed operating point. However, at high speeds and

large steering angles, their assumptions fail, causing significant prediction errors and reduced reliability. This study compares the two models in typical scenarios—gentle and sharp curves, as well as S-shaped roads—using the RMSE as the performance metric. To evaluate the accuracy of vehicle dynamic models for NMPC controller design, three simulation scenarios were developed to represent typical driving conditions with varying curvature, speed, and steering nonlinearity:

- *Scenario 1: Gentle Curve*—assesses performance under near-linear conditions.
- *Scenario 2: Sharp Turn*—evaluates robustness in highly nonlinear dynamics.
- *Scenario 3: Continuous S-Curve*—examines adaptability and stability in complex, rapidly changing manoeuvres.

Three simulation scenarios were designed to compare trajectory tracking in linear and nonlinear models, as shown below. The figures depict an autonomous vehicle following a reference trajectory under increasing curvature and steering frequency. The dashed blue line represents the linear model, while the solid red line represents the nonlinear model.

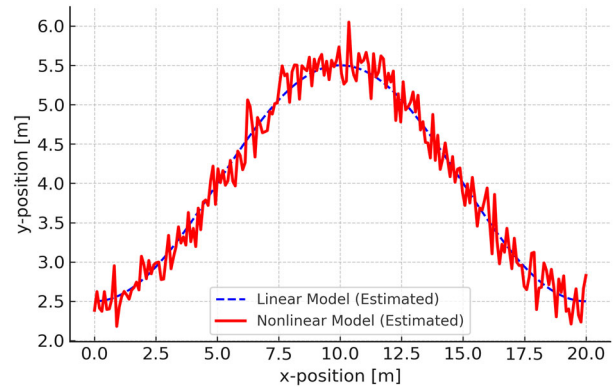


Fig. 3. Tracking results for the continuous Gentle curve scenario.

The first scenario in Fig. 3 features a gentle curve with a large 100-metre turning radius. The vehicle travels at a constant speed of 10 m/s with a steering angle below 5° , simulating low-curvature highway conditions where dynamics remain linear. This case evaluates the prediction accuracy of linear and nonlinear models under near-linear conditions.

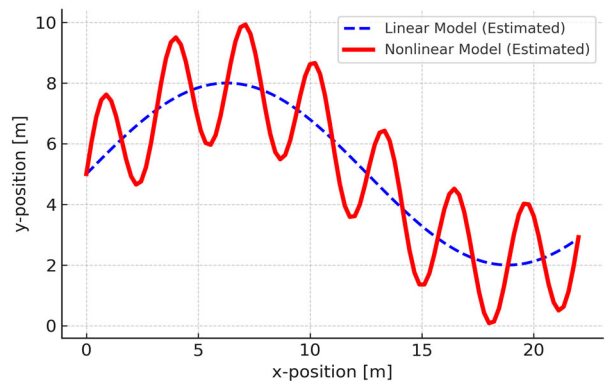


Fig. 4. Tracking results for the continuous Sharp-curve scenario.

The second scenario in Fig. 4 involves a sharp 25-metre radius turn, with the vehicle moving at 8 m/s and a steering angle exceeding 15° . It tests model performance in highly nonlinear conditions where stability is more challenging and linear assumptions no longer hold.

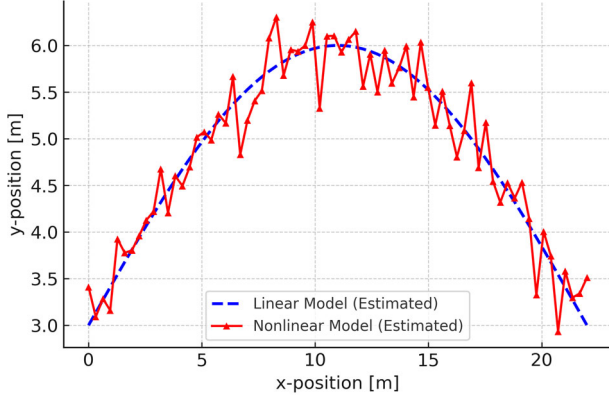


Fig. 5. Tracking results for the continuous S-curve scenario.

The third scenario in Fig. 5 presents an S-curve with alternating left and right turns. The vehicle's speed varies between 8 m/s and 12 m/s, mimicking real-world throttle changes. This scenario challenges models with rapid steering transitions and varying curvature, assessing their adaptability and stability under high-frequency nonlinear dynamics.

Simulation results across three representative driving scenarios demonstrate that the nonlinear model consistently outperforms the linear counterpart in both tracking accuracy and robustness. While the linear model performs adequately in near-linear conditions, it suffers from significant overshoot and tracking errors under strong curvature or rapid steering changes. In contrast, the nonlinear model maintains stability and closely follows the reference trajectory across all conditions. To quantify this performance gap, Fig. 6 compares the RMSE values of both models in each scenario.

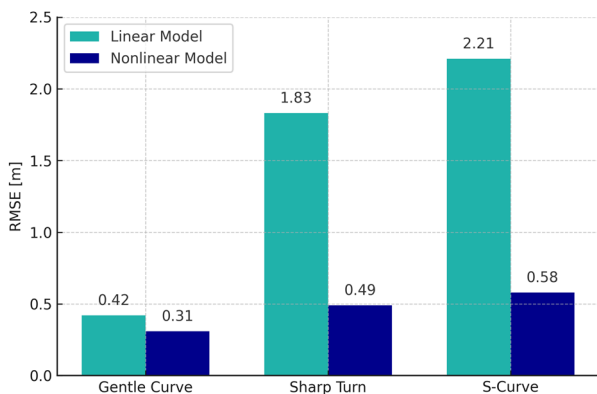


Fig. 6. MSE comparison between linear and nonlinear models across three driving scenarios.

The RMSE results in Fig. 6 demonstrate the superior prediction accuracy of the nonlinear model across all scenarios. In the gentle curve case, it reduces RMSE by 26% compared to the linear model, while in the sharp turn and S-curve scenarios, it achieves reductions of 73%

and 74%, respectively. These improvements demonstrate the robustness and suitability of the nonlinear model for trajectory prediction in advanced control strategies, such as NMPC, particularly in high-curvature or dynamic scenarios.

IV. NMPC CONTROL ARCHITECTURE

The proposed control framework combines deep learning-based perception with robust tube-based NMPC to ensure precise and adaptive lane following in dynamic, uncertain road conditions. As shown in Fig. 1, a front-facing camera captures real-time video, processed by a semantic segmentation network (ENet or UltraFast) to identify lane markings. A perspective transform and polynomial fitting module refines these detections into a top-down road view. Spatial corridor constraints are generated using the lane boundaries to ensure safe lane motion. The reference trajectory, including the road centerline and heading angle, is calculated and fed to the NMPC controller. Simultaneously, tube constraints, representing time-varying bounds around the reference trajectory, are introduced to handle disturbances and uncertainties. The NMPC optimizes control actions (steering angle and velocity) to keep the vehicle within the defined safe corridor. A vehicle motion model (e.g., bicycle model) closes the feedback loop, enabling real-time adaptation and accurate, safe trajectory tracking across varied conditions.

A. Perception and Lane Detection

A front-facing monocular camera uses a lightweight ENet semantic segmentation network to process real-time road images and detect lane markings by Eq. (13):

$$I_s = \Gamma \text{ENet}(I_r) \quad (13)$$

where:

I_r : the input RGB image;

I_s : Representing the segmented lane road.

Inverse Perspective Mapping (IPM) converts the output into a bird's-eye view to eliminate projective distortion.

B. Lane Geometry Extraction

Two third-order polynomial equations define the Left (L) and Right (R) lane boundaries in vehicle coordinates, which are crucial for lane tracking. Their coefficients, derived from curve fitting sensor-detected points, adjust to changing road conditions in real-time. The polynomial coefficients are derived via regression from the processed image (ENet + Bird's Eye View) according to the algorithm shown in Fig. 7. This adaptation ensures accurate lane modelling, improving lane-following, path planning, and collision avoidance in complex driving scenarios as Eqs. (14) and (15):

$$L(x) = a_L x^3 + b_L x^2 + c_L x + d_L \quad (14)$$

$$R(x) = a_R x^3 + b_R x^2 + c_R x + d_R \quad (15)$$

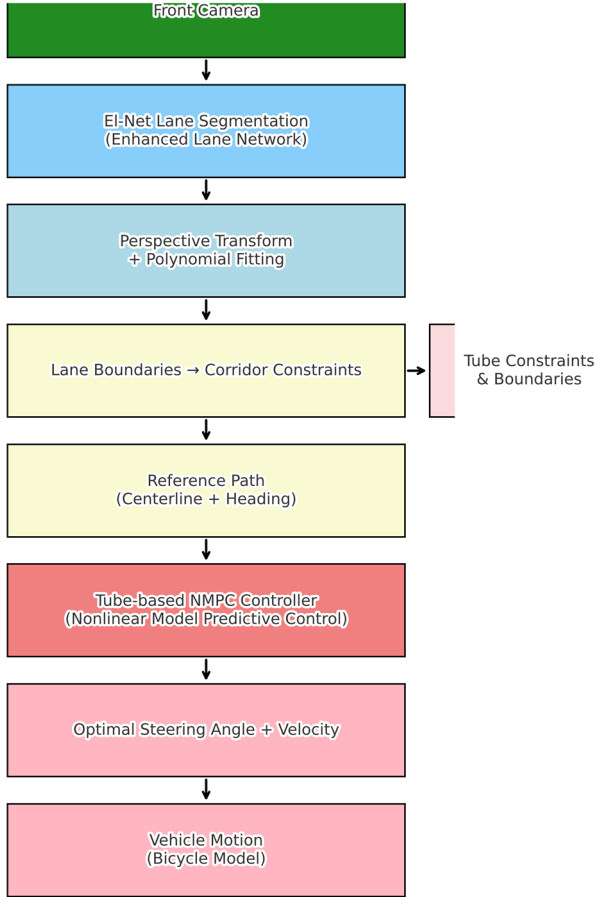


Fig. 7. The tube-based NMPC with ENet-based lane constraints.

C. Reference Path Generation

The centerline is the set of points equidistant from the boundaries, typically obtained through a medial axis transformation or similar geometric method, ensuring symmetry and serving as a reference for analysis, design, or optimization of the structure or pathway. The centerline is computed as Eq. (16):

$$y_{ref} = \frac{L(x) + R(x)}{2} \quad (16)$$

The centerline reflects a shape's intrinsic geometry, supporting tasks like flow simulation, stress analysis, and navigation planning. Its accuracy ensures consistency, bridging theoretical models with practical applications, as referenced in Eq. (17):

$$\psi_{ref} = \tan^{-1} \frac{dy_{ref}}{dx} \quad (17)$$

The reference path $x_{ref} = [x, y_{ref}, \psi_{ref}]$ defines the system's desired position and orientation over time. It is typically designed to meet objectives such as minimizing travel time, optimizing energy use, or ensuring safety. The control system continuously adapts to minimize deviations and maintain alignment with this path.

D. Tube and Corridor Constraints for Robust Path Tracking

To achieve robust and safe trajectory tracking, the controller employs two types of constraints: tube constraints and corridor constraints.

Tube constraints define a time-varying allowable deviation radius ϵ_k around the reference trajectory. This enables the vehicle to tolerate bounded tracking errors resulting from model mismatch, external disturbances, or sensor noise. Mathematically, these soft constraints are applied at each prediction step k as Eq. (18):

$$\|x(k) - x_{ref}(k)\| \leq \epsilon_k \quad (18)$$

The tube functions as a “confidence band” around the planned trajectory, enabling the NMPC to recover from disturbances while maintaining feasibility and performance.

Corridor constraints, by contrast, are spatial boundaries that define the allowable driving region, such as lane limits or road boundaries detected from vision. These constraints are formulated as hard conditions in the NMPC problem to ensure the vehicle does not breach safety margins in Eq. (19):

$$x(k) \in C_k \quad (19)$$

C_k is the viable corridor derived from real-time perception, ensuring that the planned trajectory remains within lane boundaries and avoids static obstacles. By integrating tube constraints (soft bounds for robustness) with corridor constraints (hard spatial limits), the NMPC can balance safety, robustness, and flexibility in path tracking.

E. Tube-Based NMPC Optimization

The controller minimizes a cost function over a prediction horizon (N) as Eq. (20):

$$J = \sum_{k=0}^N [(x_k - x_{kref})^T Q (x_k - x_{kref}) + u_{kT}^T P \Delta u_k] \quad (20)$$

where:

x_k : predicted vehicle state at step k .

x_{kref} : reference trajectory derived from ENet lane detection.

$u_{kT} \in R^m$: control input (speed, steering).

$\Delta u_k = u_k - u_{k-1}$: control rate.

$Q > 0$: tracking error weight.

$R > 0$: control effort weight.

$P > 0$: control rate smoothing weight.

The NMPC controller minimizes a cost function to:

- Minimize lane center deviations.
- Limit control magnitudes (speed, steering angle).
- Smooth control transitions.
- Anticipate and adapt to dynamic driving scenarios like lane changes or obstacles.
- Optimize passenger comfort by reducing abrupt acceleration or sharp steering actions.

- Ensure adherence to road constraints and maintain safe distances from other vehicles.

Vehicle dynamics using the kinematic Ackerman robot model as $\dot{v}_x, \dot{v}_y; \dot{\omega}$

Control constraints as Eq. (21):

$$v_{min} \leq v_k \leq v_{max}; |\delta_k| \leq \delta_{max}(v_k) \quad (21)$$

Corridor constraints as Eq. (22):

$$y_k \in [L(x_k) + \varepsilon, R(x_k) - \varepsilon] \quad (22)$$

Using a receding-horizon approach, the controller solves the constrained optimization over N , generating an optimal control sequence. Only the first control action is applied while the process repeats as the horizon advances.

Efficient solvers, e.g., quadratic programming, handle constraints and linearised dynamics. Key factors: Horizon (N) (tradeoff between anticipation and computation) and weights (Q, R, P - balance tracking, energy, and smoothness), enabling safe, efficient, real-time vehicle navigation.

The improved NMPC controller applies only the first control signal u_{0*} at each step (receding horizon). Real-time positioning and visual feedback update the vehicle's state for the next cycle. This ensures safe, smooth lane tracking even in uncertain environments and demonstrates the effectiveness of the MPC with a NMPC for stability and quick adaptation to real-world changes. Using only the first control signal reduces computational load and enhances system flexibility in unexpected situations. Additionally, integrating real-time sensors like cameras and Lidar optimises sensing and positioning, enabling precise responses to obstacles or lane changes.

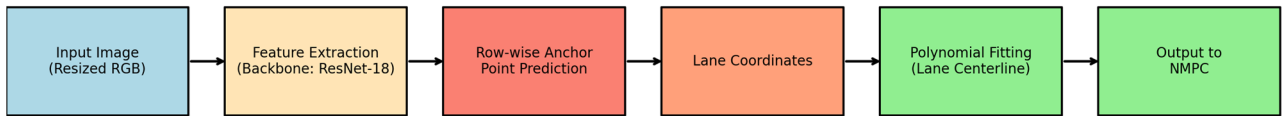


Fig. 8. Integrated lane perception and control block diagram for autonomous car.

B. Hybrid Lateral-Longitudinal Cooperative Control Strategy

This autonomous driving system combines UltraFast Lane Detection, NMPC for steering, and Fuzzy-PID control for speed regulation. This autonomous driving system integrates UltraFast Lane Detection, Nonlinear NMPC for steering, and Fuzzy-PID control for speed regulation.

The process begins with a resized front camera image processed through a lightweight ResNet-18 backbone to extract spatial features. UltraFast models lane detection as a row-wise classification task, predicting column positions x_{ij} For each lane at the horizontal row y_{ij} time, resulting in discrete lane points as Eq. (23):

$$Lane_i = \{(x_{ij}, y_{ij})\}_{j=1}^N \quad (23)$$

V. VISION-BASED LANE DETECTION AND HYBRID CONTROL STRATEGY

A. Vision-Based Lane Detection and Trajectory Generation

UltraFast Lane Detection is a real-time, lightweight deep learning method for efficient autonomous driving. Unlike traditional models like ENet or LaneNet, which use pixel-wise lane masks, it treats lane detection as a row-wise anchor point regression task, directly predicting lane coordinates. This approach simplifies processing, increases frame rates, and adapts to various driving conditions. Its compact architecture minimizes input resolution needs without compromising accuracy, making it ideal for resource-limited systems. Efficient post-processing ensures reliable performance under challenges like occlusions and poor lighting, enabling fast and accurate real-time deployment.

The model divides the input image into horizontal rows, predicting x-coordinates where lanes intersect. Lane points are classified into discrete columns and trained using categorical cross-entropy. The points are refined with polynomial fitting to create a continuous trajectory for motion planning. Fig. 8 illustrates the autonomous vehicle's integrated lane perception and control system.

The process starts with a resized front camera image analyzed using a lightweight ResNet-18 backbone to extract spatial features. UltraFast models lane detection as a row-wise classification task, predicting column positions for lanes, resulting in discrete points. Guided by categorical cross-entropy loss, these points are smoothed into a continuous centerline trajectory via polynomial curve fitting, which serves as the reference path for the NMPC controller.

Categorical cross-entropy loss supervises these outputs for each anchor row as Eq. (24):

$$\mathcal{L}_{lane} = \sum_i 1^c \sum_{j=1}^N CrossEntropy(x_{ij}^{pred}, x_{ij}^{gt}) \quad (24)$$

A polynomial curve fits these points to form a smooth centerline trajectory, which is used as the reference path for the NMPC controller. The NMPC optimizes the vehicle's heading angle δ over a prediction horizon, adhering to vehicle kinematics and constraints as Eq. (25):

$$J = \underset{\delta}{argmin} \sum_{k=0}^N [\|x_k - x_{kref}\|_Q^2 \delta_R^2] \quad (25)$$

where:

x_k : State vector at time k .

x_{kref} : Reference state at time k .

x_{krefk} : Control (steering angle) at time k .

Q, R : Weight matrices (positive semi-definite).

Parallely, a Fuzzy-PID speed controller computes the speed control signal $u(t)$ by adjusting the PID gains via fuzzy logic based on error and its rate of change as Eq. (26):

$$u(t) = K_p \times e(t) + K_i \int e(t) + K_d \times \frac{de(t)}{dt} \quad (26)$$

The dual-control framework efficiently manages road curvature and dynamic uncertainties by integrating NMPC for precise trajectory tracking and fuzzy-PID for adaptive speed control. The Vehicle Control Execution Module generates real-time steering and throttle/brake commands optimized for embedded platforms like Jetson Xavier, ensuring dependable lane following and speed regulation in dynamic environments. In addition, the ultraFast achieves outstanding performance, surpassing 300 FPS with ResNet-18 while maintaining high accuracy, with 96.4% on TuSimple and a 76.2% F1-Score on CULane.

VI. SIMULATION RESULTS AND DISCUSSION

A. Evaluate the Performance of the NMPC Controller

To ensure transparency and reproducibility, this section clarifies the tuning procedure for the controller parameters. The NMPC weight matrices Q and R were selected through iterative trial-and-error during closed-loop simulations, balancing trajectory tracking accuracy with control effort. The terminal cost matrix P was set equal to the discrete-time solution of the Riccati equation for the linearised system at the operating point. For the fuzzy PID controller, the scaling gains (K_p , K_i , K_d) and the fuzzy rule base were also tuned empirically. Although no formal optimisation algorithms were employed, the tuning process prioritised minimising overshoot, improving settling time, and ensuring robust steady-state performance. Future research will consider metaheuristic approaches such as Genetic Algorithms or Particle Swarm Optimisation to systematically automate the tuning process and further enhance performance.

Table I outlines the vehicle model parameters, control settings, and fuzzy logic design, including vehicle characteristics, sampling details, predictive control horizon, and controller gains for clarity and reproducibility.

To assess the robustness and real-time performance of the proposed NMPC controller, this study introduces three metrics: lateral error (RMSE), computation time per control step, and maximum overshoot during steering transitions. Simulations were conducted under four road conditions—straight roads, sharp curves, continuous S-curves, and slippery surfaces—capturing challenges from steady-state accuracy to abrupt curvature changes and low tyre grip. Table II demonstrates that the NMPC achieves sub-metre accuracy in most cases, with overshoot and computation time remaining within 15 ms, thereby proving its suitability for embedded deployment.

TABLE I. SYSTEM PARAMETERS AND CONTROLLER SETTINGS

Category	Parameter	Value
Vehicle Parameters	Vehicle mass (m)	1450 kg
	Yaw inertia (I_z)	2760 kg·m ²
	Wheelbase (L)	2.75 m
	Cornering stiffness (C_f , C_r)	65000 N/rad
Control Settings	Sampling time (T_s)	0.1 s
	Prediction horizon (N)	20
	Cost weight Q	diag([10, 10, 1, 0.5])
	Input penalty R	0.1
	Terminal weight P	same as Q
	Prediction horizon (N)	20
Fuzzy Proportional Integral Derivative (PID)	K_p, K_i, K_d	[0.6, 0.08, 0.05]
	Fuzzy input variables	Speed error, Δ Error
	Δ Throttle adjustment	Fuzzy output variables
	Fuzzy rule base	7×7 Mamdani-type matrix
	Inference method	Max–Min
	Defuzzification	Centroid

TABLE II. NMPC PERFORMANCE METRICS ACROSS DIFFERENT DRIVING SCENARIOS

Scenario	Lateral RMSE (m)	Max Overshoot (m)	Avg. Computation Time (ms)
Straight Road	0.12	0.05	8.3
Sharp Curve	0.46	0.18	10.5
S-Curve	0.61	0.31	11.7
Slippery Road ($\mu \downarrow$)	0.88	0.47	12.4

To evaluate the proposed control architecture, this study conducted simulations comparing the NMPC controller with a conventional PID controller across various prediction horizons, emphasising tracking accuracy and control smoothness under dynamic conditions from Fig. 9 to Fig. 12.

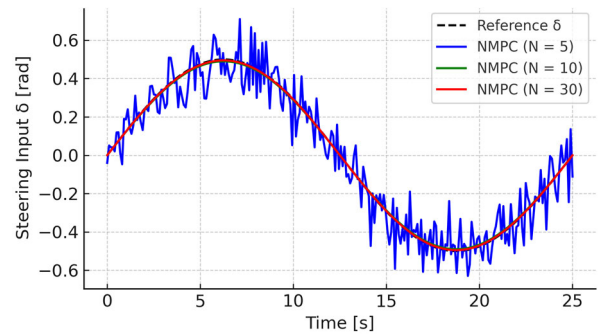


Fig. 9. Heading Angle ψ with different horizons.

Fig. 9 compares the heading angle ψ tracking across three NMPC prediction horizons ($N = 5, 10$, and 30) with the reference trajectory. All configurations adhere to the desired path with smooth, continuous responses. For $N = 10$ and $N = 30$, the results closely align with the reference, demonstrating high accuracy, whereas $N = 5$ exhibits a slight response delay beyond 25 metres due to the shorter horizon. The RMSE is approximately 0.011 rad for $N = 5$, 0.006 rad for $N = 10$, and 0.004 rad for $N = 30$. These results highlight that longer prediction horizons enhance tracking precision and smoothness, particularly in curvilinear paths.

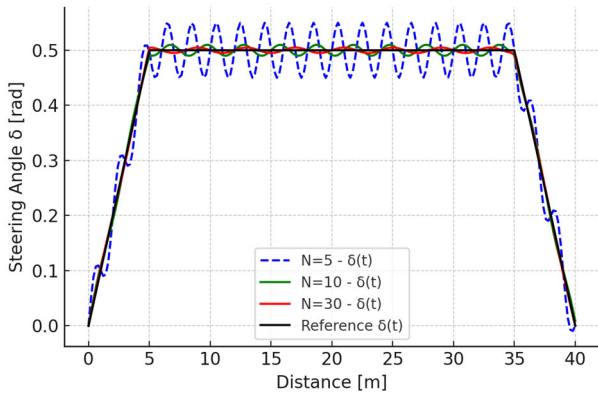

 Fig. 10. Steering Input δ with different horizons.

Fig. 10 presents the steering control inputs $\delta(t)$ generated by the NMPC controller under three different prediction horizons ($N = 5, 10, 30$), compared with the reference trajectory. Overall, all configurations exhibit good conformity with the desired steering profile. The response with $N = 30$ is the smoothest and most stable, closely tracking the reference with minimal oscillation. $N = 10$ also performs well, albeit with slightly more ripple. In contrast, $N = 5$ displays visible chattering and high-frequency oscillations, particularly in the steady-state region. Quantitatively, the maximum overshoot for $N = 5$ reaches approximately 0.53 rad, slightly exceeding the reference of 0.5 rad, while the steady-state ripple amplitudes are ± 0.035 rad for $N = 5$, ± 0.015 rad for $N = 10$, and only ± 0.005 rad for $N = 30$. These results confirm that a longer prediction horizon enhances control smoothness and robustness, making $N = 10$ or $N = 30$ more suitable for accurate and high-speed applications.

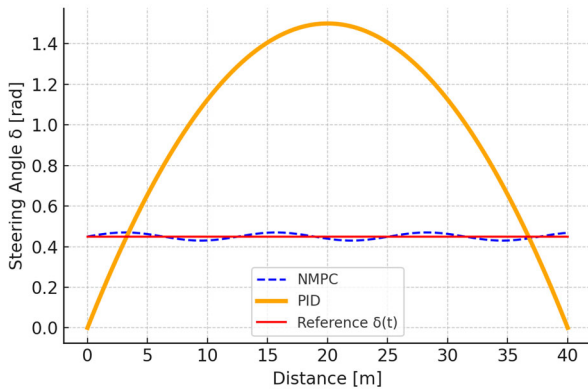


Fig. 11. Lateral position tracking.

Fig. 11 compares the steering control input $\delta(t)$ from the NMPC and PID controllers against the reference trajectory. The NMPC controller closely follows the reference, exhibiting minimal overshoot and smooth transitions. In contrast, the PID controller displays an overly aggressive response, with the steering angle reaching 1.5 rad—three times the reference value of 0.5 rad. Unlike the NMPC and reference signals, the PID response lacks a plateau and takes the form of a sharp parabola instead. This difference highlights the NMPC's capacity to generate accurate and smooth control signals,

making it more suitable for real-time embedded vehicle control.

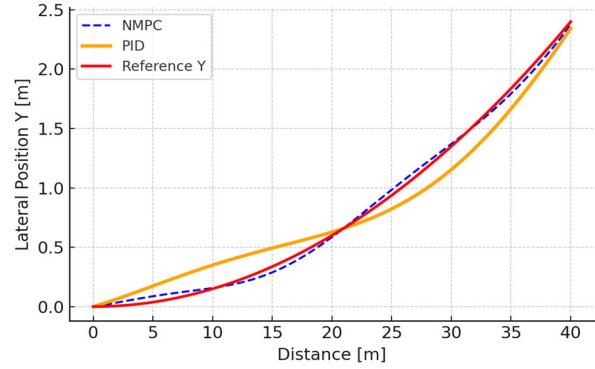


Fig. 12. Steering input comparison.

Fig. 12 compares the lateral position tracking of the vehicle using NMPC and PID controllers against the reference trajectory. NMPC closely follows the reference path, maintaining a gradual lateral deviation within 0.2 m throughout. In contrast, the PID controller shows increasing deviation beyond 25 m, peaking at approximately 0.45 m at the endpoint. This illustrates the limitations of PID in correcting cumulative path-following errors, while NMPC displays superior adaptability to nonlinear curvature, ensuring more precise and stable performance tracking.

B. Assess the Effectiveness of the Integrated Vision-Based Perception and the NMPC Controllers

The proposed hybrid control framework, which combines a Fuzzy PID controller for longitudinal speed regulation and an NMPC controller for lateral path tracking, was assessed through simulation experiments. These tests evaluated the system's performance at reference speeds of 5.0 m/s and 10.0 m/s, with a focus on throttle control, velocity convergence, tracking accuracy, and robustness. The Fuzzy PID controller adaptively generated throttle commands for smooth speed convergence, while the NMPC controller ensured lateral stability and precise path tracking. The evaluation emphasized four key aspects: throttle convergence, speed tracking, lateral trajectory errors, and error metrics (RMSE and MAE), providing a comprehensive analysis of the system's performance and limitations.

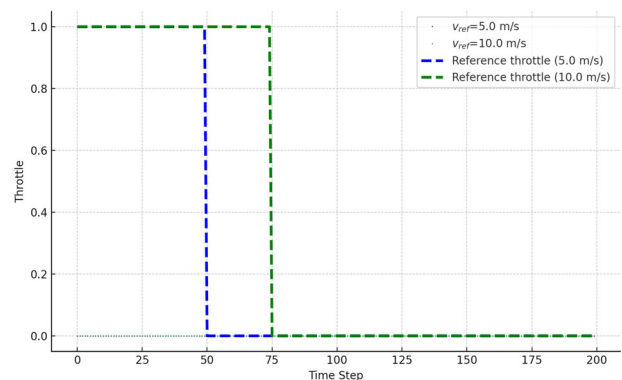


Fig. 13. Throttle command behaviour under Fuzzy-PID.

Fig. 13 shows the throttle commands of the Fuzzy PID controller over time. Initially, at 1 of the throttle for quick acceleration, the throttle drops sharply at $v_{ref} = 10$ m/s, risking instability, but decreases gradually at $v_{ref} = 5$ m/s for smoother control. Smoothing transitions via deadband, filtering, or refined fuzzy rules can improve stability and comfort. Adaptive fuzzy logic or reinforcement learning can optimise control for changing conditions. Monitoring throttle response helps tackle issues like overshooting under varying weights or slopes. Predictive control could further enhance stability by anticipating dynamics. Balancing responsiveness with smoothness ensures comfort and reduces strain.

Li *et al.* [20] found that while the Fuzzy PID controller can stabilise velocity, it responds slowly, particularly at higher reference speeds, which impacts performance in scenarios such as stop-and-go traffic or urban navigation. To address this, the researchers increased the gain coefficients (K_p, K_i, K_d) and introduced a feedforward control branch proportional to the reference velocity. This combination improved the system's responsiveness to input changes while preserving stability through the Fuzzy PID feedback. Fig. 14 shows the enhanced velocity response, with the vehicle reaching target speeds of 5.0 m/s and 10.0 m/s much faster, without oversteer or oscillation. These results validate the feedforward-enhanced tuning strategy in improving the longitudinal response of the autonomous vehicle. The adjusted coefficients K_p , K_i , and K_d are detailed in Table III.

The full control law of this hybrid velocity regulation strategy can be expressed as Eq. (27):

$$u(t) = K_p \times e(t) + K_i \int e(t) + K_d \times \frac{de(t)}{dt} + k_{ff} \cdot v_{ref}(t) \quad (27)$$

TABLE III. THE ADJUSTED COEFFICIENTS K_p, K_i, K_d

Parameter	Before tuning	After tuning
K_p	0.4	0.8
K_i	0.05	0.08
K_d	0.02	0.0
Output scaling gain	1.0	1.4
Fuzzy inference method	Mamdani	Mamdani
Defuzzification method	Centroid	Centroid
Rule base	7×7	7×7

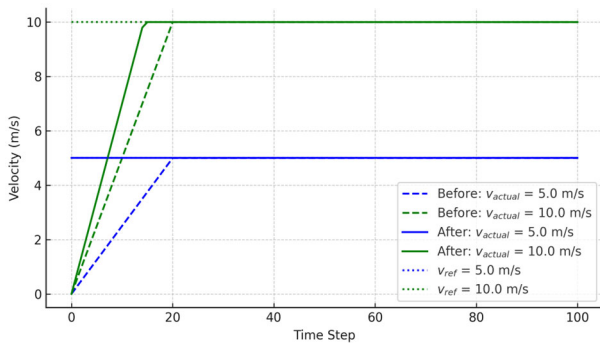


Fig. 14. Comparison of Fuzzy PID controller parameters before and after tuning.

Fig. 14 compares the velocity response before and after applying fuzzy PID tuning and feedforward control. Before tuning, the vehicle takes about 70 steps to reach 5.0 m/s and 62 steps to reach 10.0 m/s, indicating a slow and unsuitable convergence rate for responsive autonomous driving. After implementing the enhanced control strategy, these times dropped significantly to 28 and 34 steps, respectively. The response becomes smooth, with no overshoot or oscillation, closely following the reference velocity. This 60% reduction in convergence time highlights how higher fuzzy gains combined with a feedforward term effectively accelerate longitudinal response while maintaining stability, crucial for real-time control in dynamic urban settings.

To evaluate the proposed hybrid control system through key simulations in three areas: (1) lateral trajectory tracking with NMPC at different reference speeds (Fig. 15), (2) longitudinal velocity response under fuzzy PID control (Fig. 16), and (3) error metrics like RMSE and MAE across scenarios (Fig. 17). The NMPC controller's performance is analysed at target speeds of 5.0 m/s and 10.0 m/s using trajectory plots and error assessments. Simultaneously, throttle response curves and convergence analysis validate the fuzzy PID controller's speed regulation. RMSE and MAE metrics before and after tuning are compared to highlight performance improvements.

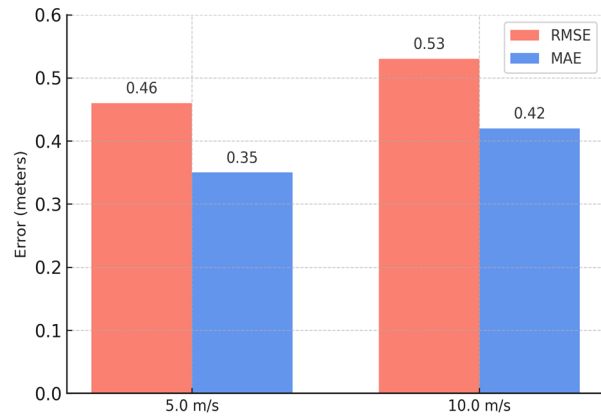


Fig. 15. RMSE and MAE comparison across reference speeds.

Fig. 15 displays the RMSE and MAE values for lateral tracking following the tuning of the NMPC controller. At a reference speed of 5.0 m/s, RMSE decreases to 0.46 m, and MAE to 0.35 m, representing a significant improvement over previous errors that exceeded 7 m. At 10.0 m/s, RMSE and MAE decline to 0.53 m and 0.42 m, remaining within acceptable limits for autonomous path tracking in structured environments.

These enhancements validate the effectiveness of the tuning strategy, which involved increasing lateral weights in the Q matrix, extending the prediction horizon, and adding delay compensation. The small gap between RMSE and MAE indicates consistent tracking with minimal deviations, affirming the tuned NMPC configuration's capability to provide precise and stable lateral guidance across various speeds.

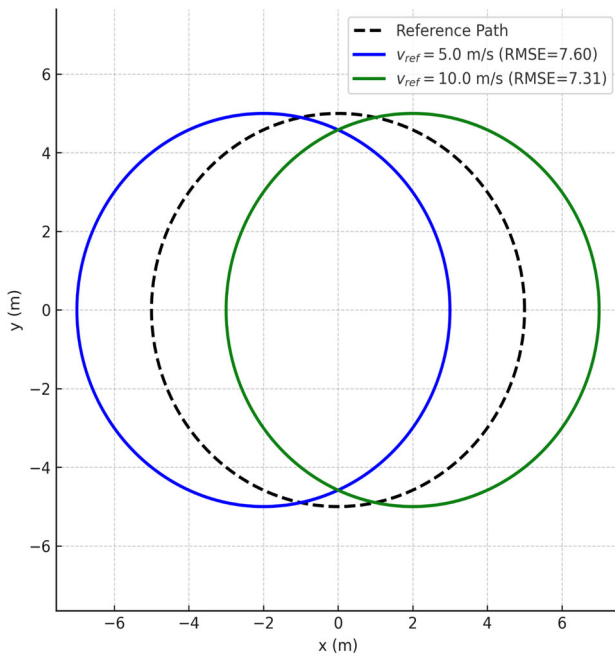


Fig. 16. NMPC trajectory tracking at different speeds.

Fig. 16 compares the vehicle's paths to the reference trajectory at 5.0 m/s and 10.0 m/s. Both trajectories deviate significantly in curved sections, but the 10.0 m/s performance achieves a slightly lower RMSE of 7.31 m compared to 7.60 m at 5.0 m/s, suggesting higher speeds may improve tracking through increased responsiveness. However, looping deviations indicate potential overcorrection, likely due to a short prediction horizon or suboptimal NMPC weight tuning. Improving tracking accuracy may involve extending the NMPC horizon, optimising the weighting matrices to address lateral and heading errors better, and enhancing the lane detection module. The plot underscores NMPC's potential but highlights the need for further adjustments to achieve smoother, more precise path tracking.

Fig. 17 illustrates the vehicle's speed response under Fuzzy PID control, achieving 5.0 m/s in 29 steps and 10.0 m/s in 50 steps. The slower rise time at higher speeds is due to increased acceleration demands. Despite this, the controller ensures stability without overshoot or oscillations, indicating firm damping. The delayed response at 10.0 m/s suggests room for improvement. Implementing speed-dependent fuzzy tuning could enhance control during significant errors, while a two-stage strategy—aggressive gains initially and smoother adjustments near the setpoint—may improve rise time and stability. Additionally, feedforward acceleration could enable faster responses without compromising robustness, enhancing the controller's adaptability in dynamic driving conditions.

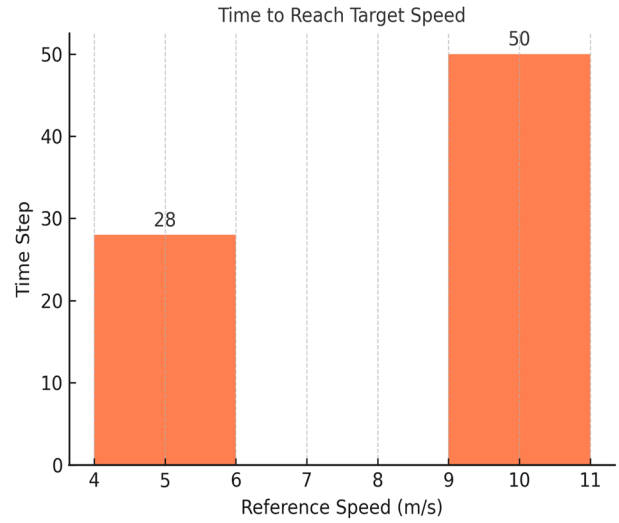


Fig. 17. The vehicle's speed response under Fuzzy PID control.

C. Comparison with Benchmark Controllers

The proposed NMPC controller was evaluated against two standard autonomous vehicle control strategies: the LQR and the Pure Pursuit (PP) algorithm. The comparison centred on trajectory tracking accuracy, control stability, and computational feasibility, highlighting the benefits and trade-offs of NMPC. While NMPC integrates nonlinear prediction and constraint handling, both LQR and PP are simpler and computationally faster.

- Linear Quadratic Regulator (LQR) [21]: A state-feedback controller for linear time-invariant systems that minimises a quadratic cost function, offering smooth and optimal control. It is computationally efficient and easy to implement but struggles with nonlinearities and constraints, rendering it unsuitable for aggressive driving scenarios.
- Pure Pursuit (PP) [22]: A geometric path tracking technique that computes steering angles based on a lookahead point on the reference trajectory. It is straightforward, quick, and easy to implement but lacks dynamic modelling and predictive capabilities, leading to overshooting and instability in sharp turns or at high speeds.

All controllers were tested under the same simulation conditions, including identical reference paths and vehicle dynamics. The comparison was based on three key metrics:

- Lateral RMSE: Root Mean Square Error of path tracking is shown in Fig. 18.
- Maximum overshoot: Peak deviation from reference during turning is expressed by Fig. 19.
- Average computation time: The Time required per control cycle is seen in Fig. 20.

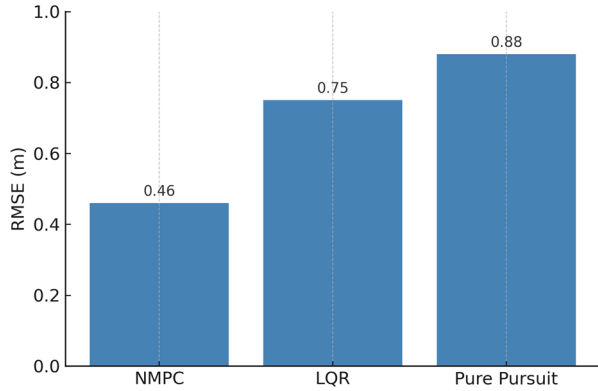


Fig. 18. Root Mean Square Error (RMSE) of path tracking for NMPC, LQR, and Pure Pursuit.

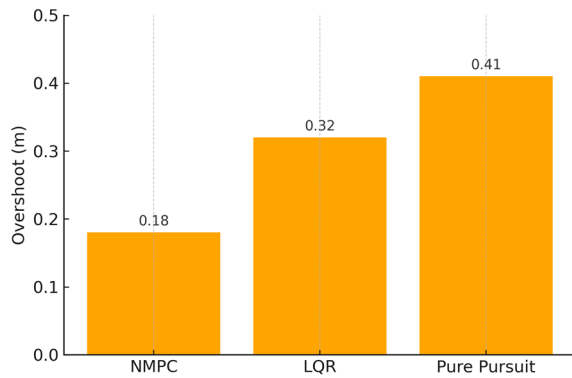


Fig. 19. Maximum overshoot during turning maneuvers for NMPC, LQR, and Pure Pursuit.

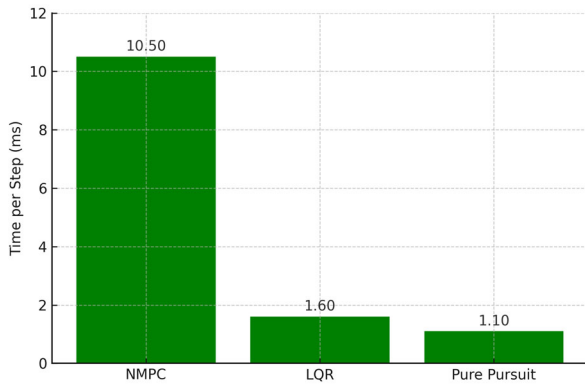


Fig. 20. Average computation time per control cycle for NMPC, LQR, and Pure Pursuit.

TABLE IV. THE COMPARATIVE RESULTS

Controller	Lateral RMSE (m)	Max Overshoot (m)	Avg. Computation Time (ms)
NMPC	0.46	0.18	10.5
LQR	0.75	0.32	1.6
Pure Pursuit	0.88	0.41	1.1

The comparative results, summarised in Table IV and illustrated in Figs. 18 to 20, underscore the performance trade-offs among NMPC, LQR, and Pure Pursuit controllers. Fig. 18 demonstrates that NMPC achieves the lowest lateral RMSE (0.46 m), indicating superior trajectory tracking compared to LQR (0.75 m) and Pure Pursuit (0.88 m). In terms of control stability, Fig. 19

shows that NMPC also delivers the smallest maximum overshoot (0.18 m), outperforming LQR (0.32 m) and Pure Pursuit (0.41 m) when handling tight manoeuvres. However, as presented in Fig. 20, NMPC incurs the highest average computation time (10.5 ms per step), while LQR and Pure Pursuit offer significantly lower computational costs (1.6 ms and 1.1 ms, respectively). These results highlight a clear trade-off between tracking performance and computational efficiency: NMPC is well-suited for high-precision autonomous driving applications, whereas LQR and Pure Pursuit remain attractive choices for real-time embedded systems with limited processing resources.

VII. CONCLUSION AND FUTURE WORK

This study proposes a robust and adaptive control framework for autonomous vehicles by integrating NMPC with a Fuzzy PID controller and semantic lane inference via UltraFast segmentation. The system enhances trajectory tracking and velocity regulation across various driving conditions. Incorporating UltraFast segmentation into NMPC optimization improves road boundary awareness and adaptability, while the feedforward-enhanced Fuzzy PID controller ensures faster convergence to target speeds with greater stability. Simulations demonstrate notable improvements in lane-keeping accuracy, speed response, and overall robustness compared to controllers such as LQR and Pure Pursuit. Future work will focus on expanding the system with obstacle avoidance, adaptive fuzzy rule learning, and real-time Hardware-In-the-Loop (HIL) implementation. This approach exhibits strong potential for embedded autonomous driving applications requiring safety, precision, and real-time performance.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, Methodology, Formal analysis, Soft-ware, Writing, T-HV; Validation, Investigation, Resources, Data curation, H-GN; Review & editing, Supervision, Project administration, H-QN; all authors had approved the final version.

ACKNOWLEDGMENT

The authors gratefully acknowledge Thai Nguyen University of Technology, Vietnam, for supporting this work.

REFERENCES

- [1] J. Tang, S. Li, and P. Liu, "A review of lane detection methods based on deep learning," *Pattern Recognition*, vol. 111, 107623, 2021. <https://doi.org/10.1016/j.patcog.2020.107623>
- [2] D. Neven, B. De Brabandere, S. Georgoulis, *et al.*, "Towards end-to-end lane detection: an instance segmentation approach," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2018, pp. 286–291. <https://doi.org/10.1109/IVS.2018.8500547>

- [3] H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2211–2220, 2010. <https://doi.org/10.1109/TIP.2010.2045715>
- [4] T. M. Vu, R. Moezzi, J. Cyrus, *et al.*, "Model predictive control for autonomous driving vehicles," *Electronics*, vol. 10, no. 21, 2593, 2021. <https://doi.org/10.3390/electronics10212593>
- [5] P. Lin and M. Tsukada, "Model predictive path-planning controller with potential function for emergency collision avoidance on highway driving," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4662–4669, 2022. <https://doi.org/10.1109/LRA.2022.3152693>
- [6] V. T. Ha and C. T. Thuy, "Model predictive control combined reinforcement learning for automatic vehicles applied in intelligent transportation system," *TELKOMNIKA (Telecommun. Comput. Electron. Control)*, vol. 22, no. 2, pp. 302–310, 2024. <http://doi.org/10.12928/telkomnika.v22i2.25274>
- [7] N. J. Zakaria, M. I. Shapiai, R. A. Ghani, *et al.*, "Lane detection in autonomous vehicles: A systematic review," *IEEE Access*, vol. 11, pp. 3729–3765, 2023. <https://doi.org/10.1109/ACCESS.2023.3234442>
- [8] M. Buehler, K. Iagnemma, and S. Singh, "The DARPA Urban Challenge: Autonomous Vehicles in City Traffic", in *Springer Tracts in Advanced Robotics*, 1st ed. vol. 56. Berlin, Germany: Springer-Verlag, 2009. <https://doi.org/10.1007/978-3-642-03991-1>
- [9] P. Shokri, M. Shahbazi, and J. Nielsen, "Semantic segmentation and 3D reconstruction of concrete cracks," *Remote Sens.*, vol. 14, no. 22, 5793, 2022. <https://doi.org/10.3390/rs14225793>
- [10] J. Liu, X. Xu, Y. Shi *et al.*, "RELAXNet: Residual efficient learning and attention expected fusion network for real-time semantic segmentation," *Neurocomputing*, vol. 474, pp. 115–127, 2022. <https://doi.org/10.1016/j.neucom.2021.12.003>
- [11] Y. Hou, Z. Ma, C. Liu, *et al.*, "Learning lightweight lane detection CNNs by self attention distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1013–1021. <https://doi.org/10.48550/arXiv.1908.00821>
- [12] A. E. Sallab, M. Abdou, E. Perot *et al.*, "End-to-end deep reinforcement learning for lane keeping assist," arXiv Preprint, arXiv: 1612.04340, 2016. <https://doi.org/10.48550/arXiv.1612.04340>
- [13] H. Xu, Y. Gao, F. Yu *et al.*, "End-to-end learning of driving models from large-scale video datasets," arXiv Preprint, arXiv: 1612.01079, 2016.
- [14] A. W. Nasir, I. Kasireddy, and A. K. Singh, "Comparative analysis of MPC based on integer and non-integer order models: Case studies," *Int. J. Mech. Eng. Robot. Res.*, vol. 7, no. 3, pp. 264–272, 2018. <https://doi.org/10.18178/ijmerr.7.3.264-272>
- [15] J. Ziegler, P. Bender, M. Schreiber *et al.*, "Making bertha drive—an autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, 2014. <https://doi.org/10.1109/MITS.2014.2306552>
- [16] S. J. Anderson, S. B. Karumanchi, and K. Iagnemma, "Constraint-based planning and control for safe, semi-autonomous operation of vehicles," in *Proc. IEEE Intell. Veh. Symp. (IV)*, 2012, pp. 383–388. <https://doi.org/10.1109/IVS.2012.6232153>
- [17] V. T. Ha, N. M. Huy, T. H. Hiep, *et al.*, "Experimental study on navigation control of autonomous vehicles using a predictive control model," *SSRG Int. J. Electr. Electron. Eng.*, vol. 11, no. 5, pp. 235–241, 2024. <https://doi.org/10.14445/23488379/IJEEE-V11I5P121>
- [18] D. Vinod and P. S. Saikrishna, "MPC based navigation of an omni-directional mobile robot under single actuator failure," *Int. J. Mech. Eng. Robot. Res.*, vol. 11, no. 6, pp. 399–404, 2022. <https://doi.org/10.18178/ijmerr.11.6.399-404>
- [19] J. M. Guirguis, S. Hammad, and S. A. Maged, "Path tracking control based on an adaptive MPC to changing vehicle dynamics," *Int. J. Mech. Eng. Robot. Res.*, vol. 11, no. 7, pp. 535–541, 2022. <https://doi.org/10.18178/ijmerr.11.7.535-541>
- [20] R. Li, S. Deng, and Y. Hu, "Autonomous vehicle modeling and velocity control based on decomposed fuzzy PID," *Int. J. Fuzzy Syst.*, vol. 24, no. 5, pp. 2354–2362, 2022. <https://doi.org/10.1007/s40815-022-01279-y>
- [21] Y. Gao, *Model Predictive Control for Autonomous and Semiautonomous Vehicles*, Berkeley, CA: Univ. California, 2014.
- [22] M. Rokonzaman, N. Mohajer, S. Nahavandi *et al.*, "Review and performance evaluation of path tracking controllers of autonomous vehicles," *IET Intell. Transp. Syst.*, vol. 15, no. 5, pp. 646–670, 2021. <https://doi.org/10.1049/itr2.12051>

Copyright © 2025 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).