



A Model Predictive Controller For Autonomous Ground Vehicles Path Tracking

Otonom Kara Araçları Yol Takibi İçin Model Öngörülü Denetleyici

Nural Mammadov* , İbrahim Alışkan 

Yıldız Technical University, Faculty of Electrical and Electronics, Department of Control and Automation Engineering, İstanbul, Türkiye

Abstract

This paper examines the challenges associated with path tracking in autonomous vehicles, employing Model Predictive Control (MPC) based on a single-track vehicle model. An MPC controller is developed based on the state-space representation of this vehicle model. Longitudinal speed is constant, and the algorithm's performance is tested with different vehicle longitudinal speeds. The generated reference path is sent to the algorithm as a reference lateral position, and the MPC controller predicts the future parameters to correctly track these given reference paths. The Proportional-Integral (PI) control method also tested on model and it is compared with MPC. The model is tested with two different paths, sinusoidal and F1 Barcelona circuit paths. It has also been tested with five different simulation scenarios. These scenarios examine the effects of parameters such as prediction horizon, control horizon, weight matrices, longitudinal velocity, computation time, and control methods. MPC optimization algorithms and simulation results are realized on the MATLAB/Simulink environment using CasADI, an open-source third-party library for nonlinear optimization and solving nonlinear equations.

Keywords: CasADI, F1 Barcelona circuit, nonlinear vehicle model, path tracking, PI control.

Öz

Bu makale, tek izli bir araç modeline dayalı Model Öngörülü Kontrol (MPC) kullanarak otonom araçlarda yol takibiyle ilişkili zorlukları incelemektedir. Bu araç modelinin durum-uzay gösterimine dayalı bir MPC denetleyicisi geliştirilmiştir. Boylamsal hız sabittir ve algoritmanın performansı farklı araç boylamsal hızlarıyla test edilmiştir. Oluşturulan referans yolu, algoritmaya referans yanal konum olarak gönderilir ve MPC denetleyicisi, bu verilen referans yollarını doğru şekilde izlemek için gelecekteki parametreleri tahmin eder. Orantılı-İntegral (PI) kontrol yöntemi de model üzerinde test edilmiş ve MPC ile karşılaştırılmıştır. Model, sinüzoidal ve F1 Barselona pisti olmak üzere iki farklı yolla test edilmiştir. Ayrıca beş farklı simülasyon senaryosu ile test edilmiştir. Bu senaryolar, tahmin ufku, kontrol ufku, ağırlık matrisleri, boylamsal hız, hesaplama süresi ve kontrol yöntemleri gibi parametrelerin etkilerini inceler. MPC optimizasyon algoritmaları ve simülasyon sonuçları, doğrusal olmayan optimizasyon ve doğrusal olmayan denklemlerin çözümü için açık kaynaklı üçüncü taraf bir kütüphane olan CasADI kullanılarak MATLAB/Simulink ortamında gerçekleştirildi.

Anahtar Kelimeler: CasADI, doğrusal olmayan araç modeli, F1 Barselona pisti, PI kontrol, yol takibi.

1. Introduction

A vehicle that can drive itself from one location to another or park itself has always been one of the popular topics that people have always dreamed of and constantly worked on. The early academic studies were mostly theoretical and

mathematical solutions due to the technological conditions of the day. These studies, which started in the last quarter of the 20th century, have gradually become testable and improvable with the development of technology. In parallel, increasing the processing capacity of electronic equipment and computers has accelerated the development of autonomous driving technology. This academic and commercial research aimed to bring this technology to the end user. Because these studies have progressed successfully, many automotive companies have now managed to use the autonomous driving option on the cars they produce and deliver them to the end user.

*Corresponding author: nural.mammadov@std.yildiz.edu.tr

Nural Mammadov  orcid.org/0000-0002-3920-5412

İbrahim Alışkan  orcid.org/0000-0003-3901-4955



This work is licensed under "Creative Commons Attribution-NonCommercial 4.0 International (CC BY NC)".

There are different types of cases within autonomous driving. Some functions can act for different situations such as autonomous parking, autonomous obstacle avoidance, emergency maneuvering, and overtaking. Also, path tracking is one of them. This problem is usually divided into two steps: planning and tracking the path. Path tracking process is turned into a control problem and the aim is for the vehicle to perform the desired path as a result (Ritschel et al. 2019).

Some studies in the literature propose Fuzzy Logic Control (FLC) for solving the path-tracking problem of autonomous vehicles. The most significant difference between this study and our paper is that while no model is needed for FLC, the dynamic or kinematic model of the vehicle must be obtained in the MPC control method (Aliskan 2025).

In the literature, two distinct types of MPC are employed to address this issue: Linear MPC (LMPC) and Nonlinear MPC (NPMC). In the LMPC method, a linearized vehicle model is used. At each operating point, the vehicle model is converted to a linear time-varying model by the linearization method (Falcone et al. 2007).

The main difference between this study from previous studies in the literature is nonlinear system is chosen for applying path-tracking problems using MPC instead of a linear system. One of the other differences is the selected vehicle model type. Generally kinematic vehicle model is used for these studies in literature, but they do not behave well at high speeds. The algorithm is based on a dynamic vehicle model and provides good results even at high speeds.

Different types of solution methods have been presented within Linear MPC in current studies. Linear Time-Varying Model Predictive Control (LTV-MPC) is based on discrete state-space representations, where the system dynamics are linearized to derive the corresponding matrices.

In addition to these, different MPC methods have been used in the literature for the path tracking problem. Robust MPC provides more reliable and robust performance by accounting for system model errors, parameter uncertainties, and the impact of external disturbances (Mata et al. 2019, Peng et al. 2019). It is also used to adjust cost function values (Rokonuzzaman et al. 2020). It has been explained in many review articles why the MPC method gives better results and is more effective than other methods for path tracking (Li et al. 2021).

As a result, MPC, like other classical control methods, is one that approaches solving the problem at hand by utilizing

specific mathematical functions and gives successful results even without using large amounts of data. MPC method has the advantages of having a manageable system for systems with many inputs and outputs (Cao et al. 2021); good orbit tracking performance at medium and high speeds if the predictive model is created successfully (Hajiloo et al. 2021) and is suitable for systems where more than one control method is used like nested (Wang and Liu 2021).

On the other hand, the need for high and fast computational capability during online solutions is one of its main disadvantages of MPC (Yu et al. 2021). Different software and libraries are used to overcome this problem and obtain fast solution time. MPC toolbox in MATLAB/Simulink (Yakub and Mori 2015), qpOASES solver on ACADO (Chowdhri et al. 2021) and CasADI solver are some of them (Laurense and Gerdes 2021). In this study, CasADI solver library was used during solution of the optimal control issue.

In their study, Falcone et al. (2007) implemented MPC using kinematic or linearized vehicle models that could not fully capture the reference, especially at higher speeds. Similarly, Huang et al. (2021) focused on urban speed conditions with fixed horizon values in the MPC application for autonomous vehicles.

This study uses a nonlinear dynamic vehicle model to more accurately represent the real vehicle behavior under changing speed conditions. We evaluate the effects of the control horizon, prediction horizon, and weighted matrices separately. Finally, the preference for CasADI, which is not frequently used in the literature for the solution of nonlinear equations, distinguishes our study from previous studies.

The content of this study proceeds as follows: Second section presents vehicle dynamics fundamentals, modeling methods and equations. Section 3 explains the outline and theoretical details of MPC. Section 4 contains simulation results and detailed explanations of this paper with MPC and PI Control under different path and control parameters. At the end, Section 5 concludes all results of the work.

2. Vehicle Dynamic Model

Vehicle dynamics models are critical for predicting the motion responses of vehicles to various forces and inputs, providing autonomous driving applications. There are different vehicle modelling methods in literature such as longitudinal models, lateral models, dynamic models and kinematic models etc. Dynamic model perspective of vehicle is focused in this study.

The single-track model maintains the vehicle's fundamental dynamic characteristics while condensing the complicated system into a single plane (Hu et al., 2020). The related force and torque equations based on the vehicle's center of gravity are utilized to get these simplified equations. A dynamic vehicle model that accurately captures the motion state of the vehicle is necessary for accurate and responsive path tracking at high speeds. A three-degree-of-freedom dynamics model, based on equations based on Newton's second law, is shown in Figure 1.

According to Newton's second law, it is used for accurate high-speed path following. There are some relations between the forces. These include the vehicle's lateral translational motion, where the equation controlling yaw dynamics and lateral velocity inside the inertial coordinate system is formulated by moment balance about the axis. The vehicle's lateral translational motion, yaw moment relationship, and lateral velocity can be explained as follows (Li et al. 2019):

$$m(\dot{v}_y + v_x \dot{\psi}) = F_{yf} + F_{yr} \quad (1)$$

$$I_z \ddot{\psi} = l_f F_{yf} - l_r F_{yr} \quad (2)$$

$$\dot{Y} = v_x \sin(\psi) + v_y \cos(\psi) \quad (3)$$

In this equations, m is mass of the vehicle, I_z is rotational inertia, l_f is distance of vehicle's center and front tire and l_r is distance of vehicle's center and rear tire respectively (Hu et al. 2020). The ψ is yaw angle, $\dot{\psi}$ is yaw rate is $\ddot{\psi}$ is yaw angle acceleration, v_x is longitudinal velocity, v_y is lateral velocity and $\dot{v}_y$ is lateral acceleration.

F_{yf} , F_{yr} are lateral tire forces of wheels, and F_{xf} , F_{xr} are longitudinal tire forces. Longitudinal tire forces are not

displayed in equations since the speed is taken as a constant in this investigation. The front tire's cornering stiffness is denoted by C_f , whereas the rear tire's is denoted by C_r .

$$F_{yf} = 2C_f \alpha_f \quad (4)$$

$$F_{yr} = 2C_r \alpha_r \quad (5)$$

Using small angle approximation α_f and α_r slip angle of front wheel and rear wheel. δ_f is front wheel steering angle (Rajamani 2012):

$$\alpha_f = \left(\delta_f - \frac{v_y + l_r \dot{\psi}}{v_x} \right) \quad (6)$$

$$\alpha_r = - \left(\frac{v_y - l_f \dot{\psi}}{v_x} \right) \quad (7)$$

At the end, vehicle motion equations are presented as follows:

$$\dot{v}_y = \left(-\frac{2C_f + 2C_r}{mv_x} \right) v_y + \left(-v_x - \frac{2C_f l_f - 2C_r l_r}{mv_x} \right) \dot{\psi} + \left(\frac{2C_f}{m} \right) \delta_f \quad (8)$$

$$\ddot{\psi} = \left(-\frac{2C_f l_f - 2C_r l_r}{I_z v_x} \right) v_y + \left(-\frac{2C_f l_f^2 + 2C_r l_r^2}{I_z v_x} \right) \dot{\psi} + \left(\frac{2C_f l_f}{I_z} \right) \delta_f \quad (9)$$

$$\dot{Y} = v_x \sin(\psi) + v_y \cos(\psi) \quad (10)$$

The given equations are evaluated linearized vehicle model, which describes vehicle's lateral dynamics. This structure allows the MPC to effectively track the reference path. Vehicle parameters used in model are given in Table 1 (Hu et al. 2020).

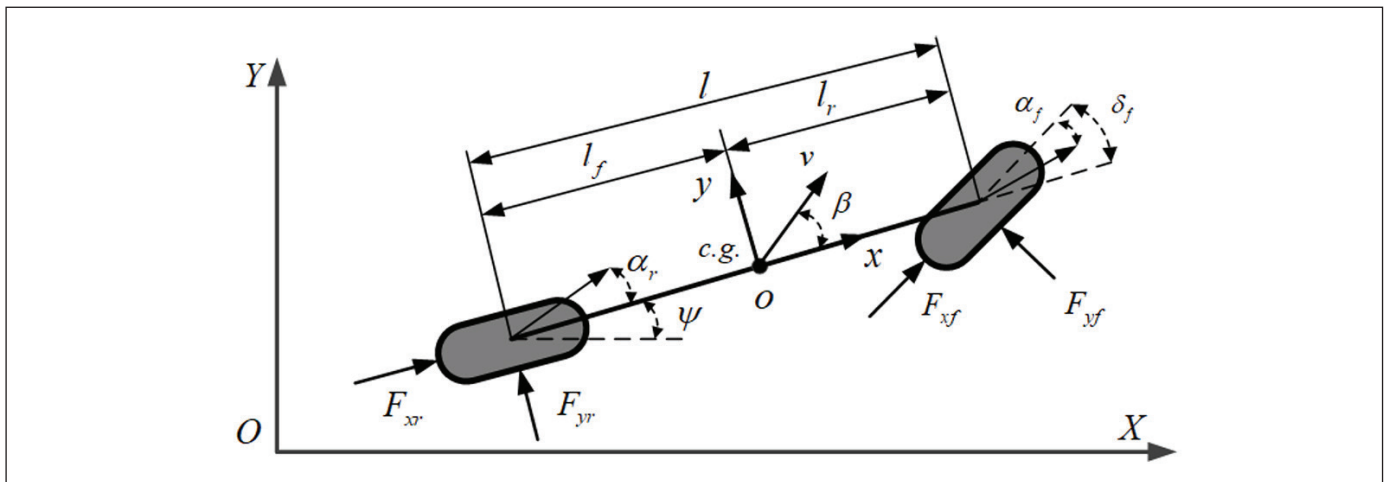


Figure 1. Dynamic single-track vehicle model (Hu et al. 2020)

Table 1: Vehicle parameters.

Symbol	Value/Unit	Description
m	1416 kg	Vehicle mass
I_z	1523 kg.m ²	Inertia moment
l_f	1.016 m	Distance between vehicle center and front tire
l_r	1.562 m	Distance between vehicle center and rear tire
C_f	47000 N/rad	Cornering stiffness of front
C_r	48000 N/rad	Cornering stiffness of rear

3. MPC Controller Formulation

MPC is an advanced technique for optimal control. The ideal control sequence is calculated based on the intended and anticipated system states using the first sequence element as control inputs. The key objectives of this controller are to ensure precise path tracking, maintain stable yaw dynamics, and optimize computational performance, even under varying road adhesion conditions. Safe and stable envelopes are established to constrain the wheel steering angle while reducing deviations from the reference path (Cui et al. 2020).

Stability and robustness analyses of MPC algorithms are analyzed using well-established methodologies in the literature (Qin & Badgwell 2003). Since this is not the focus of this article, a detailed stability analysis is not performed; however, some assumptions are based on to ensure stability. As in Ławryńczuk (2013), Tatjewski and Ławryńczuk (2020), stability is explained by assuming that MPC is symptomatically stable and that the optimization problem can be applied at every sampling time.

Nevertheless, it is known that applying higher weight coefficients in the MPC algorithm gives good results in terms of stability and robustness (Tatjewski and Ławryńczuk 2020). For this reason, precautions such as selecting quadratic cost function with positive Q and R matrices and prefer long prediction and control horizons were taken to avoid instability situations (Alişkan 2019, 2021).

To implement MPC, one must establish a state-space equation for the model. These equations are derived from the vehicle dynamic model relationships discussed in the previous chapter. Next, it is necessary to identify the system states and system inputs. System states are internal variables used for predictions, while system inputs are the decision variables that the controller adjusts to achieve optimal performance. The primary responsibility of the designed MPC controller is to ensure precise path tracking, with vehicle stabilization

along the predefined path being another critical objective (Cui et al., 2020).

Its state vector, which is continuous time, is shown by $X = [v_y, \psi, \dot{\psi}, Y, \delta_f]^T$, and system input vector is $u = [\dot{\delta}_f]$. Equation of the system model is given below:

$$\dot{X} = f(X, U) \quad (11)$$

The created system prediction model provides the vehicle dynamics equations to be seen within a function.

$$\dot{X} = \frac{d}{dt} \begin{bmatrix} v_y \\ \psi \\ \dot{\psi} \\ Y \\ \delta_f \end{bmatrix} = \begin{bmatrix} -\left(\frac{2C_f + 2C_r}{v_x m}\right)v_y + \left(\frac{2C_r l_r - 2C_f l_f}{v_x m} - v_x\right)\dot{\psi} + \left(\frac{2C_f}{m}\right)\delta_f \\ \left(\frac{2C_r l_r - 2C_f l_f}{v_x I_z}\right)v_y - \left(\frac{2C_f l_f^2 + 2C_r l_r^2}{v_x I_z}\right)\dot{\psi} + \left(\frac{2C_f l_f}{I_z}\right)\delta_f \\ v_x \sin(\psi) + v_y \cos(\psi) \\ \dot{\delta}_f \end{bmatrix} \quad (12)$$

Path tracking problem, originally in continuous time, was converted into a discrete-time system to facilitate implementation. Zero Order Hold (ZOH), one of the frequently used methods when converting to a discrete-time system, was used. This method is also known as the Euler method. T_s represents sampling time. Discrete-time system equation is shown below:

$$x_{k+1} = x_k + f(x_k, u_k) T_s \quad (13)$$

Objective function tries to solve the problem by minimizing the error for the parameters it uses. Also, this objective function used for path tracking is shown as an optimal control problem. The optimization problem of MPC has the following objective function below:

$$J(x_k, U_k) = \min_{U_k} \left(\sum_{i=1}^{H_p} \left(\|y_{k+i} - y_{k+i}^{ref}\|_Q^2 \right) + \sum_{i=1}^{H_c} \left(\|\Delta u_{k+i-1}\|_R^2 \right) \right) \quad (14)$$

subjected to,

$$u_k \in \mathcal{U} \quad \forall k \in [t, t + H_c]$$

$$x_k \in \mathcal{X} \quad \forall k \in [t, t + H_p]$$

$$\Delta u_k = u_k - u_{k-1}$$

$$i = 1, \dots, H_p$$

In this equation, H_p represents prediction horizon value and H_c represents control horizon value. Q and R matrices are positive diagonal weight matrices. Tuning matrices are defined $Q \geq 0 \in \mathbb{R}^{5 \times 5}$ and $R > 0 \in \mathbb{R}^{1 \times 1}$. Weight matrices allow the assignment of different weight matrices to the state vectors or input vectors used in the system. At the same time, these matrices are used to normalize the parameters used in the system if necessary. In this way, it prevents deviations.

Also control horizon values impact to system performance in MPC. It is number of future time-steps over which control inputs are explicitly optimized. The control horizon defines the extent to which control actions or adjustments are planned within the current optimization cycle.

In the MPC, some system parameters are constrained for preventing a result that is not consistent with the physical conditions while following the desired path. The steering angle are constrained between ± 30 deg.

$$u_{\min} \leq u_k \leq u_{\max} \quad (15)$$

$$x_{k,\min} \leq x_k \leq x_{k,\max} \quad (16)$$

The correct control sequence is obtained that should be applied to the system by solving this equation. The initial element of the control sequence is found because the controller applies the obtained solution to the model. MPC performs this process in every sample time step until the entire path is completed. The simulation results chapter

explains the details of these values and related test scenarios. MPC parameters are detailed in the simulation results section for each scenario.

4. Simulation Results

All simulations are performed on a computer platform with an Intel Core i5-11300H CPU at 3.1GHz and 16GB memory. CasADI is used for solving nonlinear equations in the model. It is an open-source third-party library for nonlinear optimization, similar to MATLAB Symbolic Math Toolbox. However, CasADI is much faster and has good problem-solving capabilities in complex nonlinear equations. In addition, MPC structure diagram in this study is shown in Figure 2.

Results are split into five scenarios, and the simulations are performed in the MATLAB version 2024a environment. These five different simulation scenarios test to model under different conditions. Each scenario aims to examine a specific aspect of the control system such as prediction horizon, control horizon, longitudinal velocity, weight matrices, computation times and RMSE value.

Root mean square error (RMSE) is one of standardized methods used to find error rate of a system. It is calculated by taking square root of average of squared differences between reference and actual position. It is widely employed to assess regression models' accuracy. The RMSE value equation is shown below.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{ref,i} - y_{actual,i})^2} \quad (17)$$

Also, two different paths have been established to test the model tracking performance. The sinusoidal path is used in first four scenarios and F1 Barcelona path is used in last scenario. A sinusoidal reference signal is a periodic signal

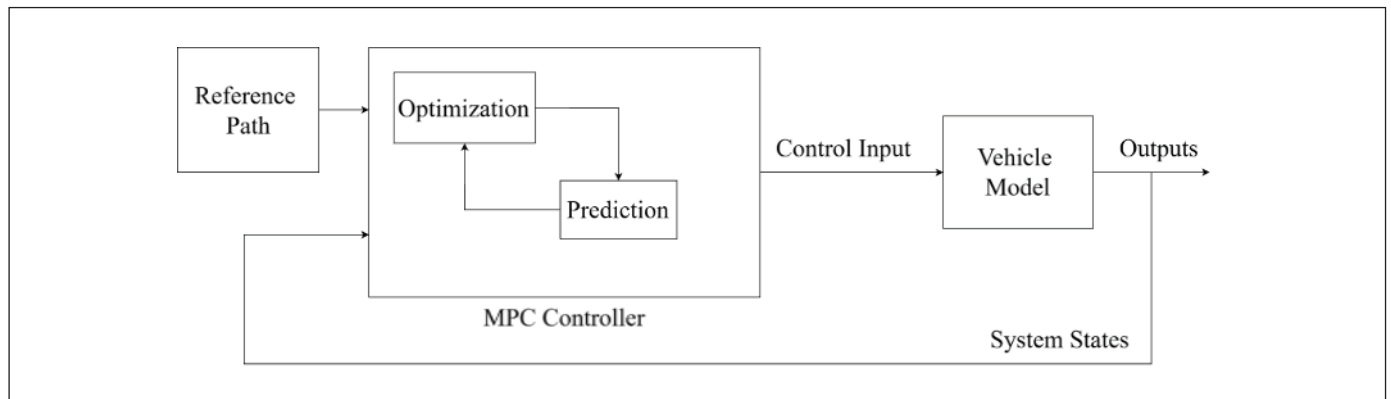


Figure 2: MPC structure diagram for path tracking.

commonly used for sampling or comparison in control systems and analyses. Sine function formula is shown below:

$$A \cdot \sin\left(\frac{2\pi}{T} \cdot k \cdot T_s\right) \quad (18)$$

The amplitude A is selected 5, time period T is selected 30, sampling time T_s is selected 0.01 s and k is the discrete-time index like an integer representing the sample number. Discrete-time index k values are between $[1, 4000]$ in this equation. Details of these five scenarios are explained below. These five scenarios will be discussed in detail under each heading.

- **Scenario 1:** $v_x = 10$ m/s with changing $H_p = 10, 20, 30$.
- **Scenario 2:** $H_p = 30$ with changing $v_x = 15, 25, 35$ m/s.
- **Scenario 3:** $H_p = 30$ and $v_x = 35$ m/s, with Standart Configuration, Q-tuned Configuration and H_c -tuned Configuration.
- **Scenario 4:** $H_p = 30$ and $v_x = 11$ m/s, with PI Control and MPC.
- **Scenario 5:** $H_p = 30, H_c = 10$, and $v_x = 10$ m/s with F1 Barcelona path

4.1. Scenario 1: Effect of Prediction Horizon

Scenario 1 analyzes the effect of the prediction horizon on the tracking performance. In this scenario, $v_x = 10$ m/s, $H_c = 1$, and $H_p = 10, 20, 30$, respectively. At the same time, weight matrices are selected to $Q = [1, 1, 1, 1, 1]$ and $R = [0.25]$.

These Q, R matrices are called as “Standart Configuration” in the rest of the paper. Standard configuration does not impact results, however Scenario 3 will discuss the effect of changing weight matrices, which is called Q-tuned. If H_c values are changed in model, it is called H_c -tuned Configuration. Also it is analyzed in Scenario 3.

Figure 3 shows the tracking performance under different prediction horizons. As demonstrated by the simulation results, increasing the prediction horizon value provides better performance. The MPC controller cannot predict the future system behavior well at low H_p values. On the other hand, with a longer H_p , the controller can make decisions by taking the future into account more, which allows the system to follow the reference path more successfully.

Figure 4 shows lateral position error and steering angle response. At short prediction horizon the lateral deviation of the system was high and fluctuations were observed in the steering inputs. At $H_p = 20$, the system showed a more balanced tracking performance; lateral error and steering angles remained limited. At the longest prediction horizon, both the lateral error decreased and the steering commands were softer and lower amplitude. This shows that the controller's ability to predict further points of the reference path significantly improves the stability and control effort of the system.

Figure 5 shows the positive and negative maximum lateral position errors and RMSE values. The lateral errors reach

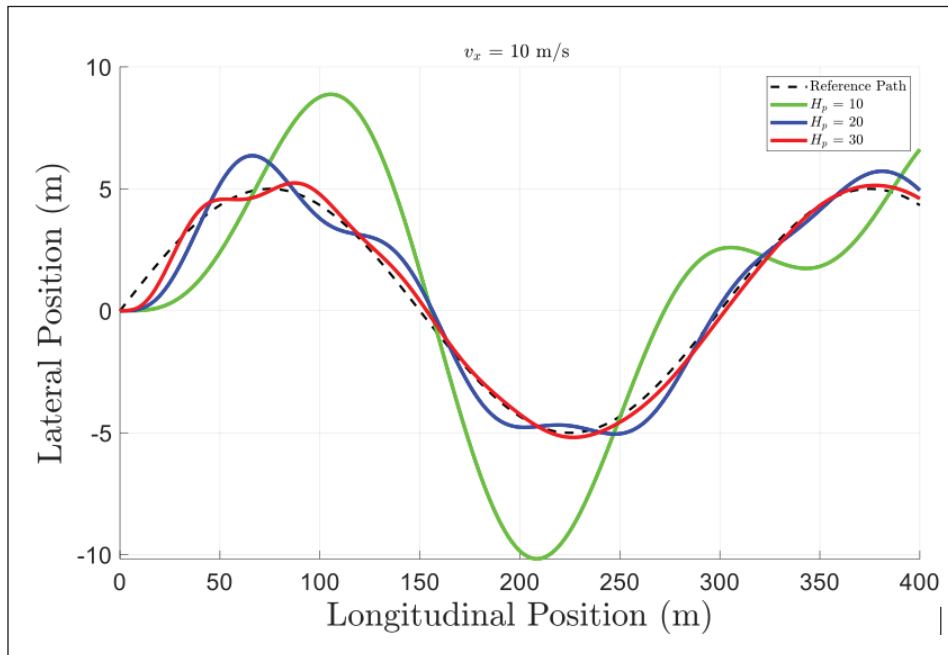


Figure 3. Tracking performance with different prediction horizons at $v_x = 10$ m/s.

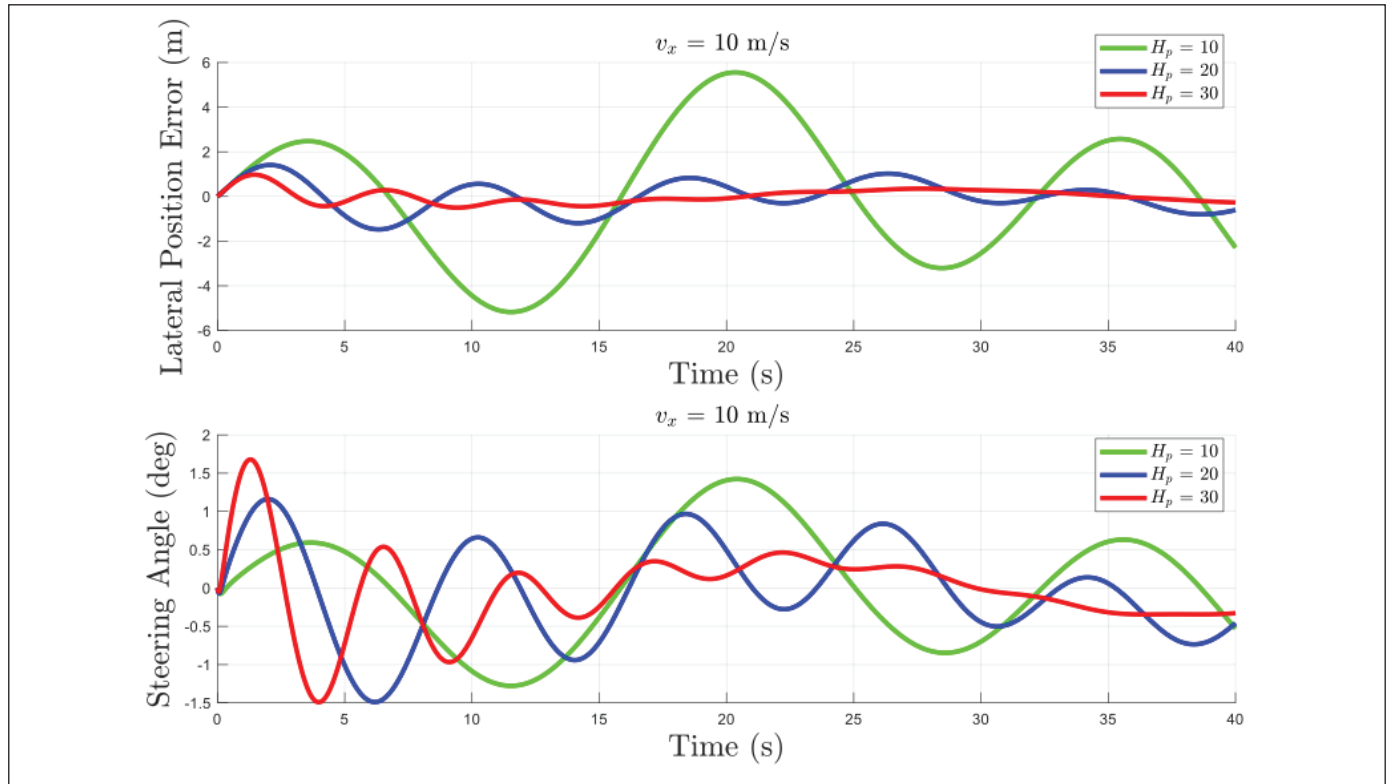


Figure 4. Lateral position error and steering angle at different prediction horizons.

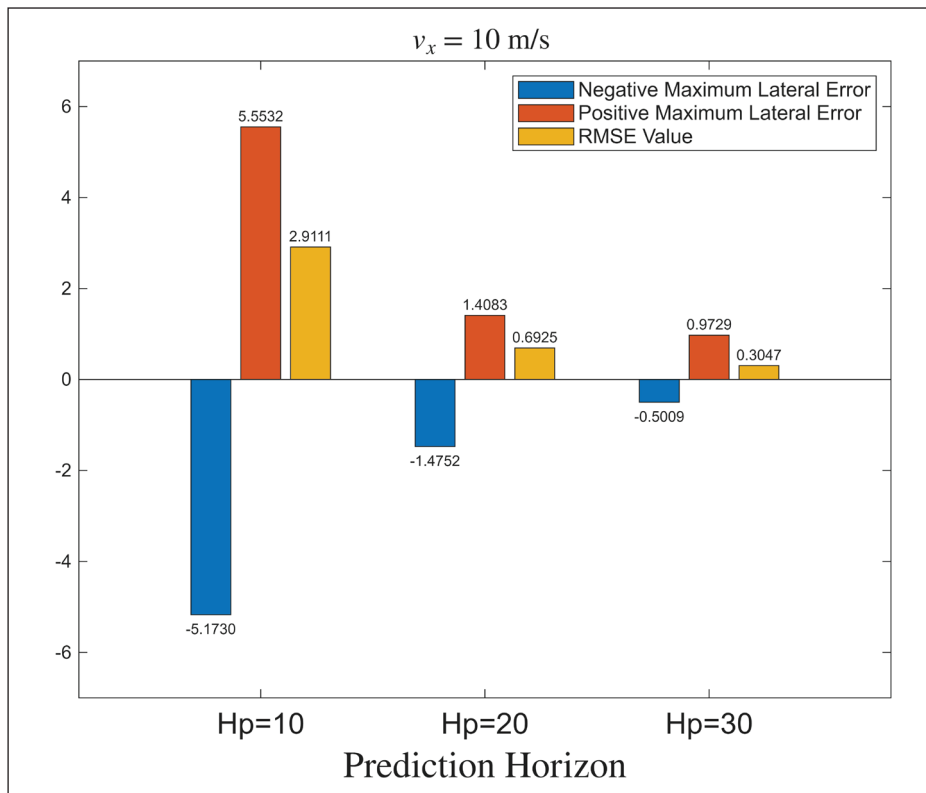


Figure 5. Lateral position error and RMSE value at different prediction horizons.

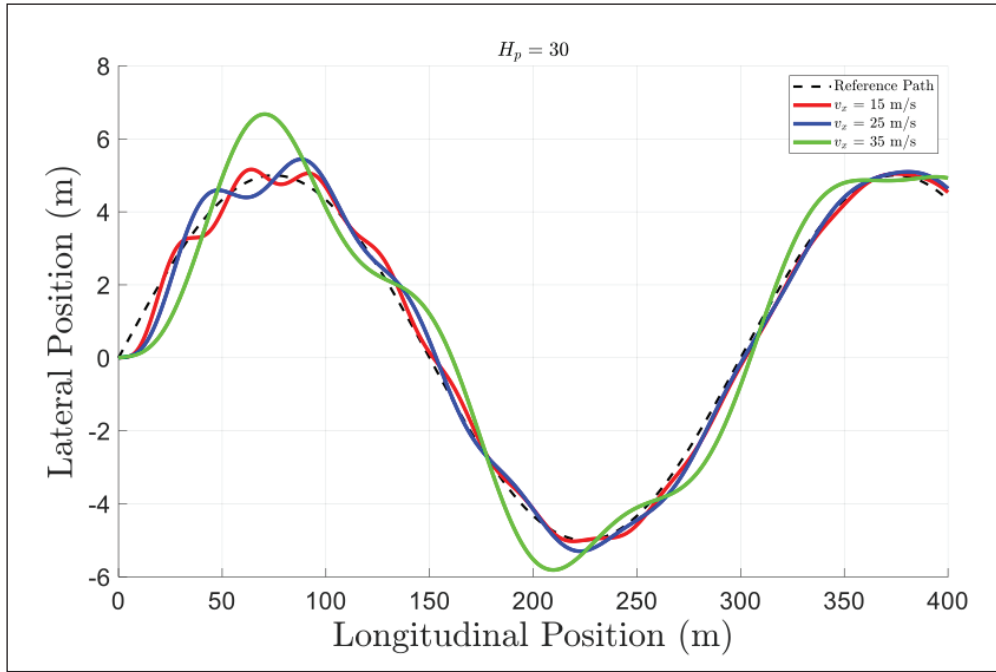


Figure 6. Tracking performance with different longitudinal speeds.

very high values at $H_p = 10$. The RMSE value is also 2.91, indicating that the overall tracking accuracy is relatively poor. At $H_p = 20$, RMSE value is 0.69. These results show that the errors are significantly reduced due to the system's ability to make more forward-looking predictions. At $H_p = 30$, RMSE value is approximately 0.3. These results show that the system converges more stably and accurately to the reference path with longer prediction horizons. However, the impact of this improvement on computational cost should also be discussed. The total simulation times in Scenario 1 was observed as $H_{p,10} = 24.06$, $H_{p,20} = 33.47$, and $H_{p,30} = 43.63$, seconds, respectively.

4.2. Scenario 2: Effect of Longitudinal Velocity

Scenario 2 analyzes the effect of the longitudinal velocity on the tracking performance. In this scenario, $H_p = 30$, $H_c = 1$ and $v_x = 15, 25, 35$ m/s respectively. Weight matrices are selected as Standard configuration.

Figure 6 shows the tracking performance under different longitudinal velocities. The system follows the reference path quite successfully at low speeds, and the errors remain low. The harmony with the reference path continues at medium speed, but some differences are observed in some areas, especially in turns. At the highest speed, deviations from the reference path become apparent, and the system shows sharper oscillations. It shows that the system cannot respond quickly enough to reference path changes at high speed.

Figure 7 shows the lateral position error and steering angle under different longitudinal velocities. The error amplitude is small at low speed, but the frequency is high, indicating faster but controlled corrections. At medium speed, the error amplitude increases, but the system is still stable. At high speed, the lateral error becomes significantly larger, and the damping of oscillations is delayed. When examined in terms of steering angles, control actions at low speeds are more aggressive, while at high speeds, the system produces a more cautious and damped steering response. It shows that the effect of control actions on the system response decreases with speed, and MPC should be supported with stronger or different configurations at high speeds.

Figure 8 shows the lateral error and RMSE values under different longitudinal velocities. While the RMSE value was the lowest at low speed, these values increased significantly at medium speed, and a severe performance decrease was observed at high speed. These findings clearly show that the increase in longitudinal speed negatively affects the tracking performance, and the control system needs stronger response strategies in rapidly changing dynamic conditions.

The total simulation times in Scenario 2 was observed as $v_{x,15} = 43.91$, $v_{x,25} = 43.75$ and $v_{x,35} = 43.59$ seconds, respectively. As these results show, changing longitudinal velocity does not impact to computation time.

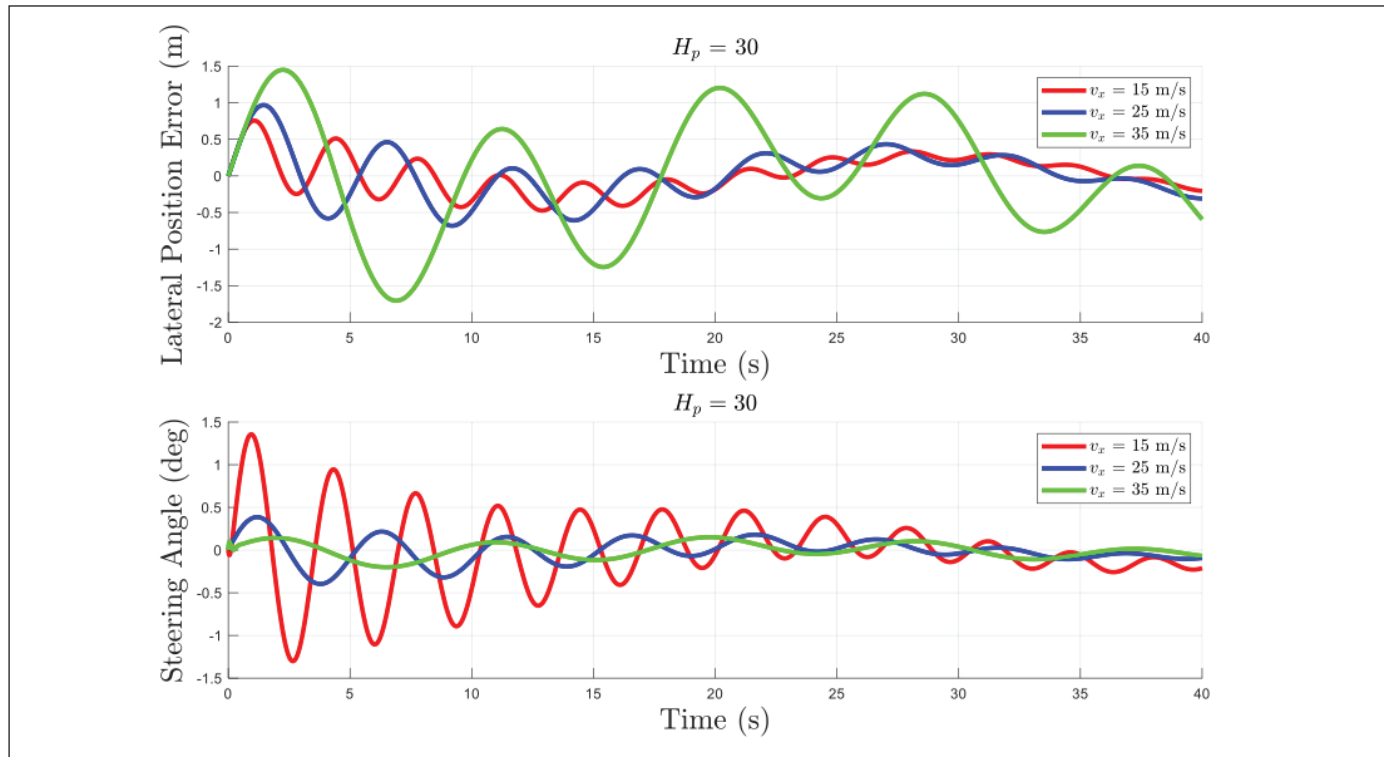


Figure 7. Lateral position error and steering angle at different longitudinal speeds.

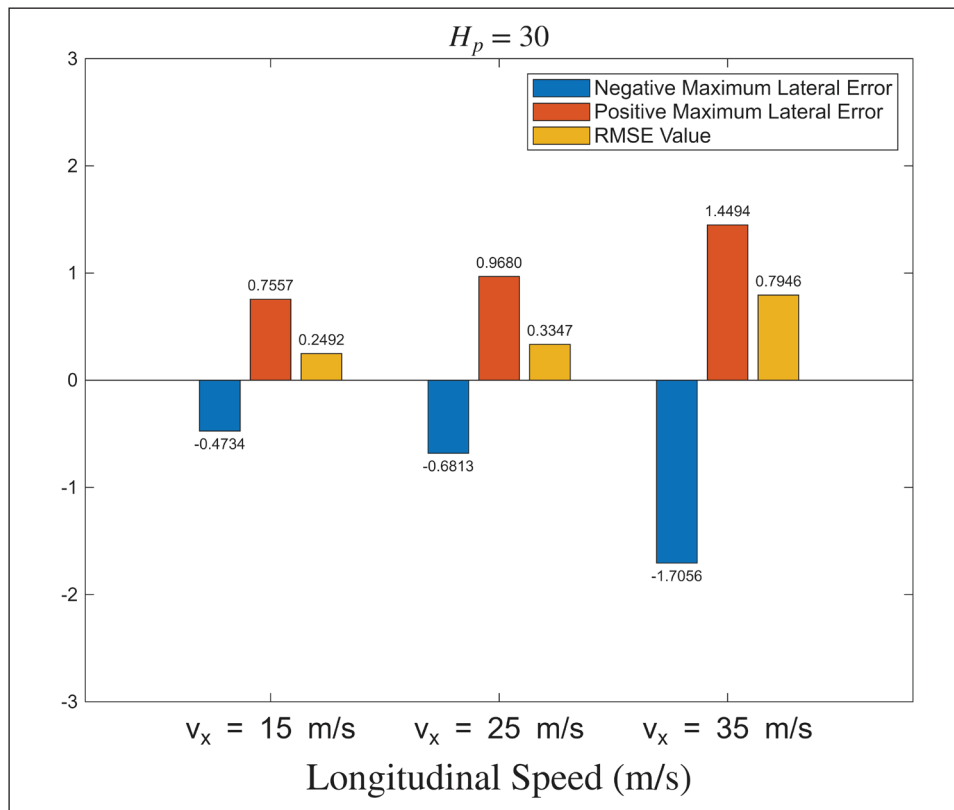


Figure 8. Lateral position error and steering angle at different prediction horizon values.

4.3. Scenario 3: Effect of H_c and Q Parameters at High Speed

Scenario 3 analyzes the effect of H_c and Q matrices on tracking performance. In this scenario, $H_p=30$ and $v_x = 35$ m/s respectively. This scenario aims to demonstrate the effect of control parameters on increasing tracking accuracy at high speeds. Therefore, the system performance was tested with three different control parameter configurations. These configurations are:

- Standart configuration: $H_c = 1$, $Q=[1,1,1,1,1]$, $R=[0.25]$
- Q -tuned configuration: $H_c = 1$, $Q=[10,5,1,100,0.5]$, $R=[0.25]$
- H_c -tuned configuration: $H_c = 5$, $Q=[1,1,1,1,1]$, $R=[0.25]$

Figure 9 shows the tracking performance under different control horizons and weight matrices. In the standard configuration, the system deviates significantly from the reference, especially in the first inclined segments. Since the controller cannot predict the system's dynamics very well, the tracking performance is low. In contrast, both the Q -tuned and H_c -tuned configurations followed the reference path very closely and significantly reduced the deviations.

Figure 10 shows the lateral position error and steering angle. The standard configuration exhibited amplitude oscillations in the lateral error time series, and the error could not be damped for a long time. In contrast, the Q -tuned and H_c -tuned structures produced error responses with smaller amplitudes and faster damping. Regarding steering angle,

the Q -tuned structure used the control input more aggressively, while the H_c -tuned structure produced softer and lower amplitude steering commands. It shows that the system provides more stable and comfortable control with H_c .

Figure 11 shows the lateral position error and RMSE values. The lateral errors are the highest in the standard, and this structure has the lowest performance. These values have decreased significantly in the system with the optimized Q matrix, and the error amplitudes have been minimized. The most successful result was obtained in the H_c -tuned structure. This situation shows that the predictive spreading of the control input over time significantly contributes to the system's stability.

The total simulation times in Scenario 3 was observed as $T_{Standard} = 43.365$, and $T_{Q-tuned} = 44.535$ and $T_{Hc-tuned} = 44.633$ seconds, respectively. As these results show, without increasing the prediction horizon, indirectly computing time, changing weight matrices and control horizon values provides better tracking performance results.

4.4. Scenario 4: PI Control and MPC

Automated vehicle control systems rely on precise lateral dynamics regulation to ensure stability and path-following accuracy. Among various control strategies, Proportional-Integral (PI) Control is commonly used because it is simple and effective in tracking reference trajectories with minimal steady-state error. PI Control and MPC performances are examined for sine wave reference path in this scenario. The general PI Control equation is given below.

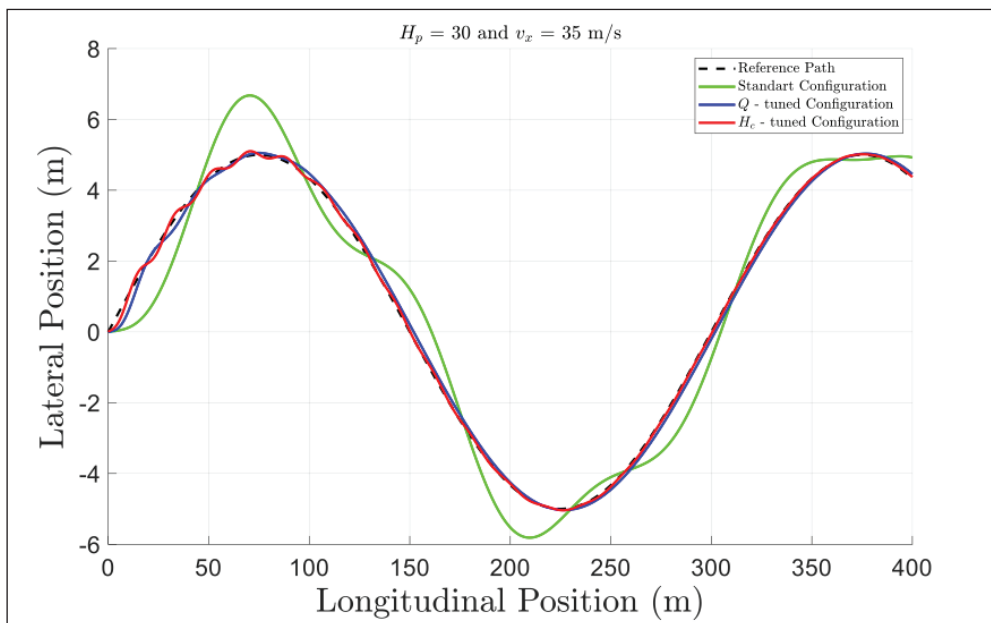


Figure 9. Tracking performance with different control horizon and weight matrix.

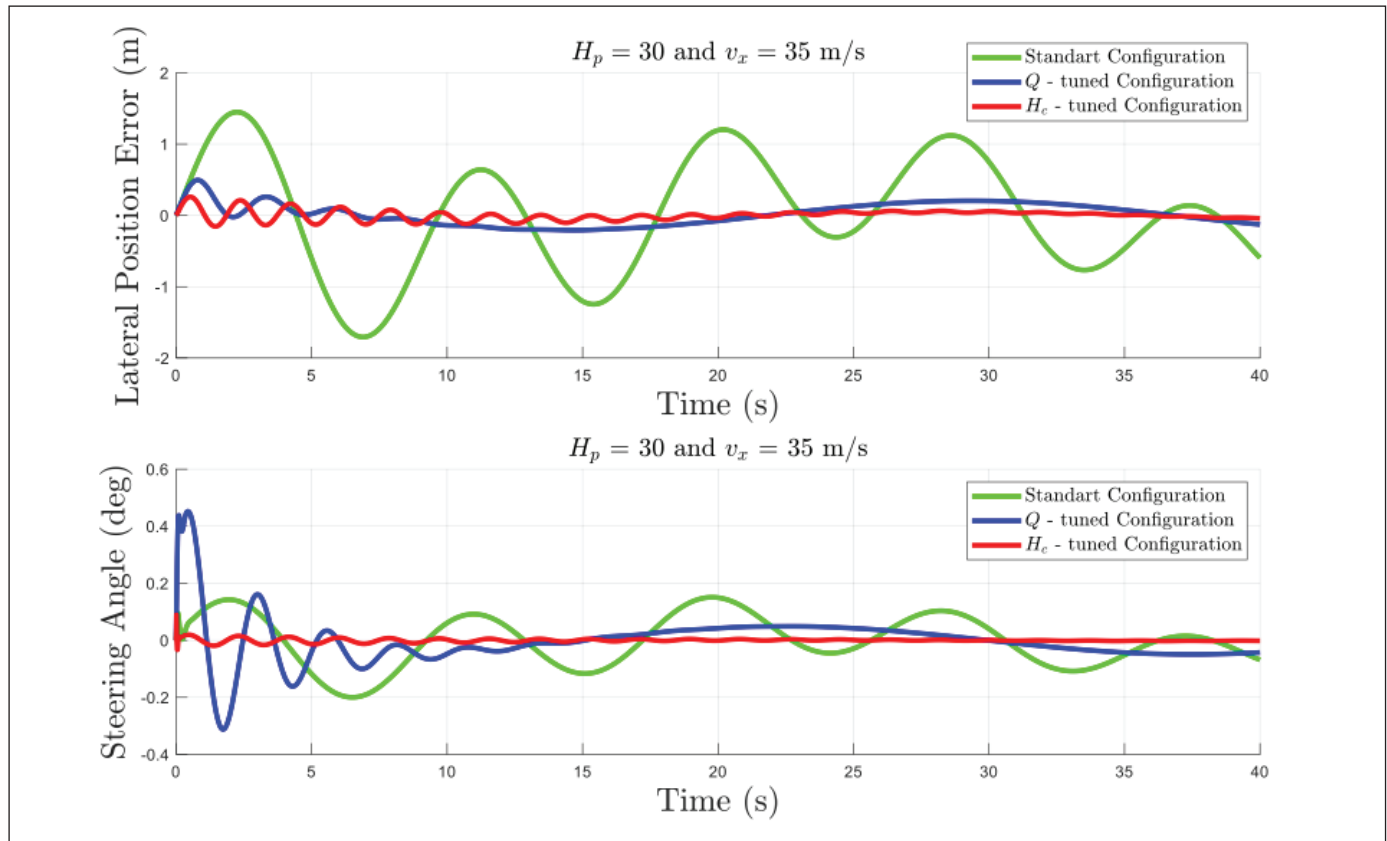


Figure 10. Lateral position error and steering angle with different control horizon and weight matrice.

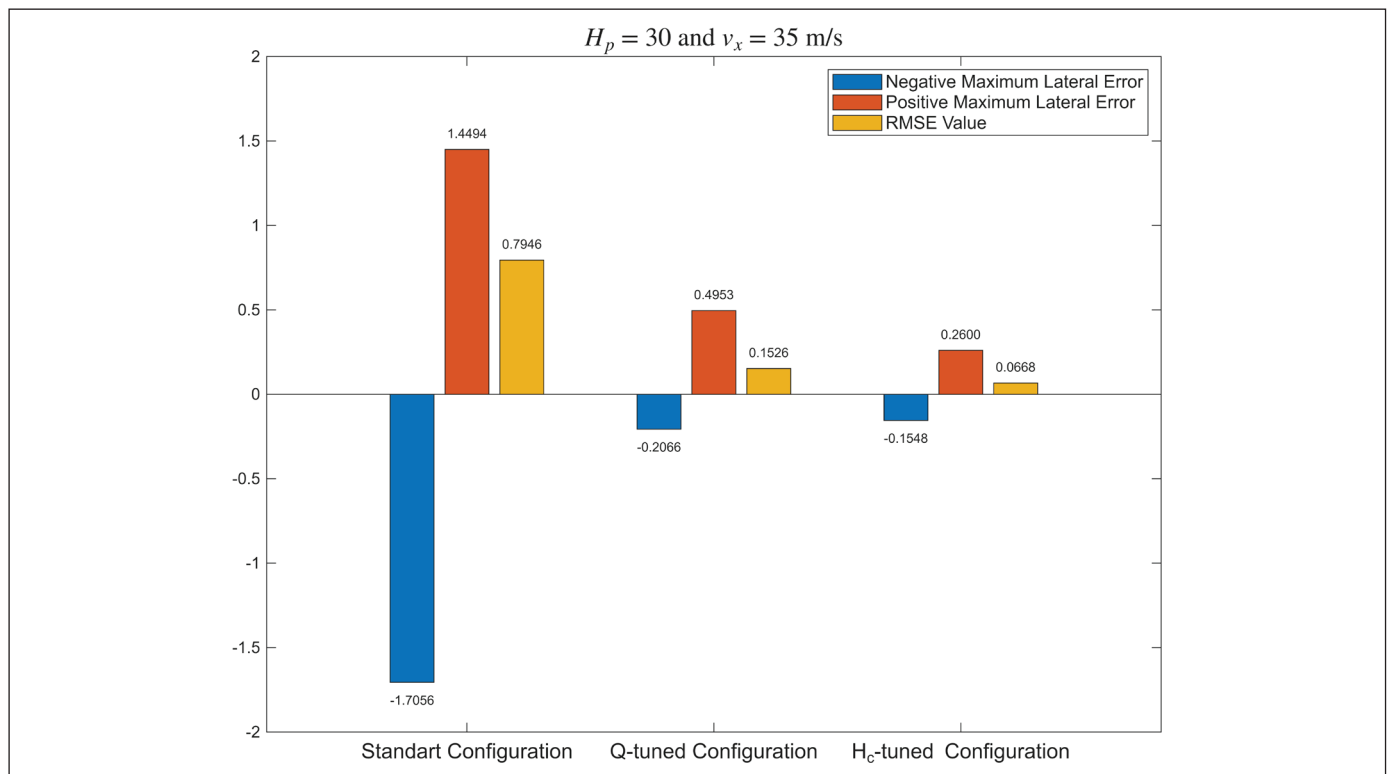


Figure 11. Lateral position error and steering angle with different prediction horizon values.

$$\delta_f = K_p e + K_i \int e dt \quad (25)$$

K_p is proportional gain coefficient and K_i is integral gain coefficient in this formula. Suitable values is obtained using the Ziegler-Nichols method. The Ziegler-Nichols method is a widely used heuristic tuning approach to determine proportional and integral gains for a controller.

At the beginning of Ziegler-Nichols, K_i is set to 0 in the simulation. Then gradually increase K_p until sustained oscillations appear in the lateral error response. Oscillation period of it is obtained as T_u . Then ultimate gain K_u is obtained by observing the point where oscillations first appear. Ultimate gain $K_u = 1.2$ is found and oscillation period $T_u = 4$ sec respectively.

Then, applying Ziegler-Nichols and we used these K_p and K_i values in PI controller:

$$K_p = 0.45 \cdot 1.2 = 0.54 \quad (26)$$

$$K_i = \frac{K_p}{1.2 T_u} = \frac{0.54}{1.2 \cdot 4} = 0.1125 \quad (27)$$

Scenario 4 analyzes the effect of control methods on tracking performance. In this scenario, two different control methods are tested. Longitudinal velocity is constant 11 m/s because this speed is the maximum limit of PI Control to work in a stable region. The PI controller becomes unstable at constant speeds greater than 11 m/s. PI Control parameters

are selected as $K_p = 0.54$, $K_i = 0.1125$. MPC parameters are selected as $H_p = 30$ and $H_c = 1$ and standart configuration.

Figure 12 shows the tracking performance under different control methods. Both control methods track the reference path, but when examined carefully, significant differences are observed. The PI controller's response on the reference path contains more oscillations, especially in regions with high slopes or rapid changes of direction, and slight fluctuations are seen in the tracking. The MPC controller's response is smoother and converges more smoothly to the reference path. The main reason for this difference is that the MPC optimizes by considering future system behaviors and has a limits sudden changes in control inputs. This situation reveals that the MPC is more advantageous, especially regarding comfort and system stability.

Figure 13 shows the lateral position error and steering angle. It shows that the PI controller responds quickly and accurately to the reference, but these responses are high-frequency and continuous oscillations. The error signal contains high-frequency oscillations and is not damped. Conversely, MPC produced a larger error at the beginning but damped this error over time, providing a more stable and smooth tracking performance. The steering angle results explains this difference: the PI controller applies continuous high-frequency steering corrections, while the MPC controller acts with lower-frequency and lower-amplitude control actions.

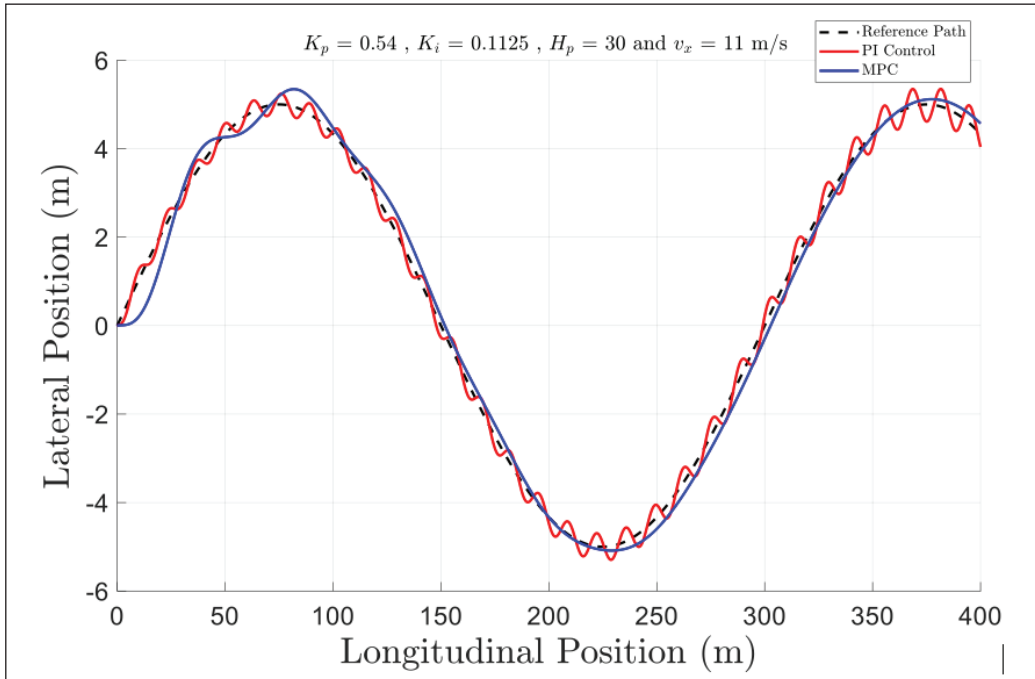


Figure 12. Tracking performance with PI Control and MPC.

Figure 14 shows the lateral position errors and RMSE values. The PI controller performed better and its RMSE value was lower. Although the RMSE value of MPC was higher, this difference is mainly due to the initial delay. However, when the high-frequency oscillations in the PI controller are considered, it is understood that low error

values are achieved at the expense of control effort. The total simulation times in Scenario 4 was observed as $T_{PI} = 0.012$ and $T_{MPC} = 43.463$ seconds, respectively. As can be seen from these results, even though PI control computation time is much faster than MPC but MPC is better than PI Control in terms of accuracy and comfort.

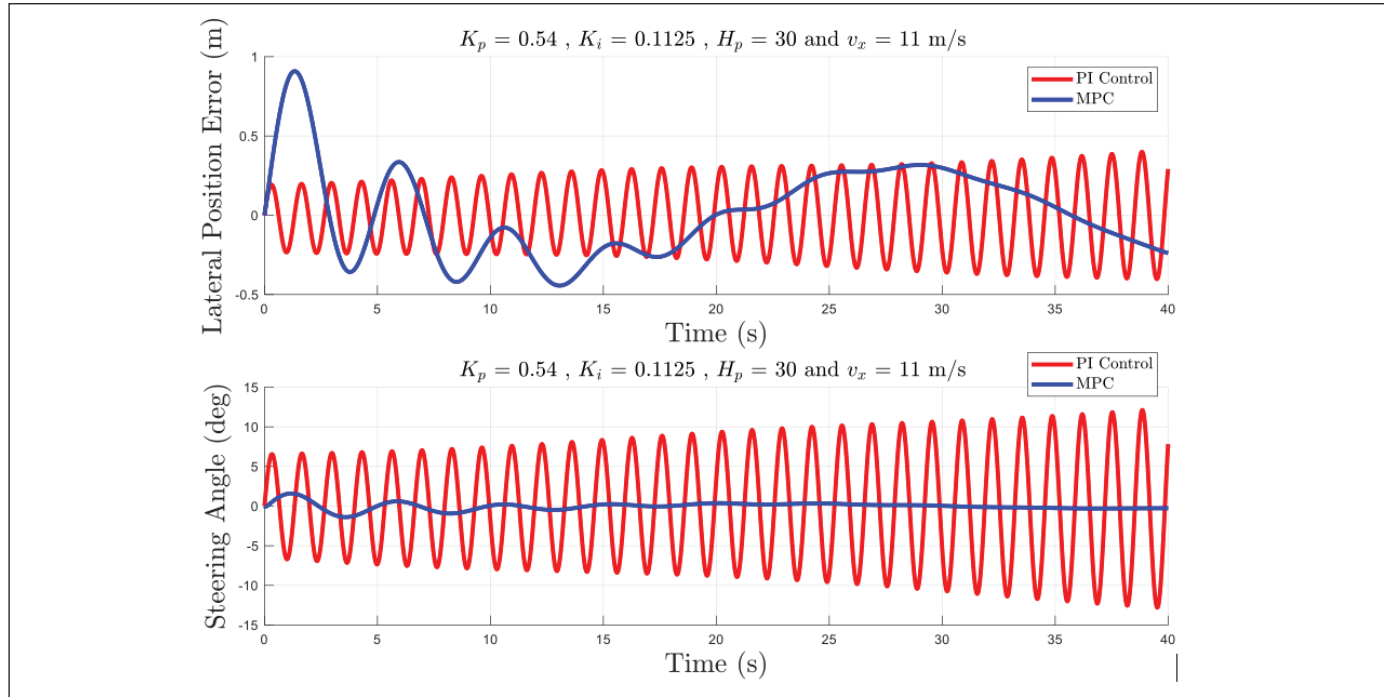


Figure 13: Lateral position error and steering angle with PI Control and MPC.

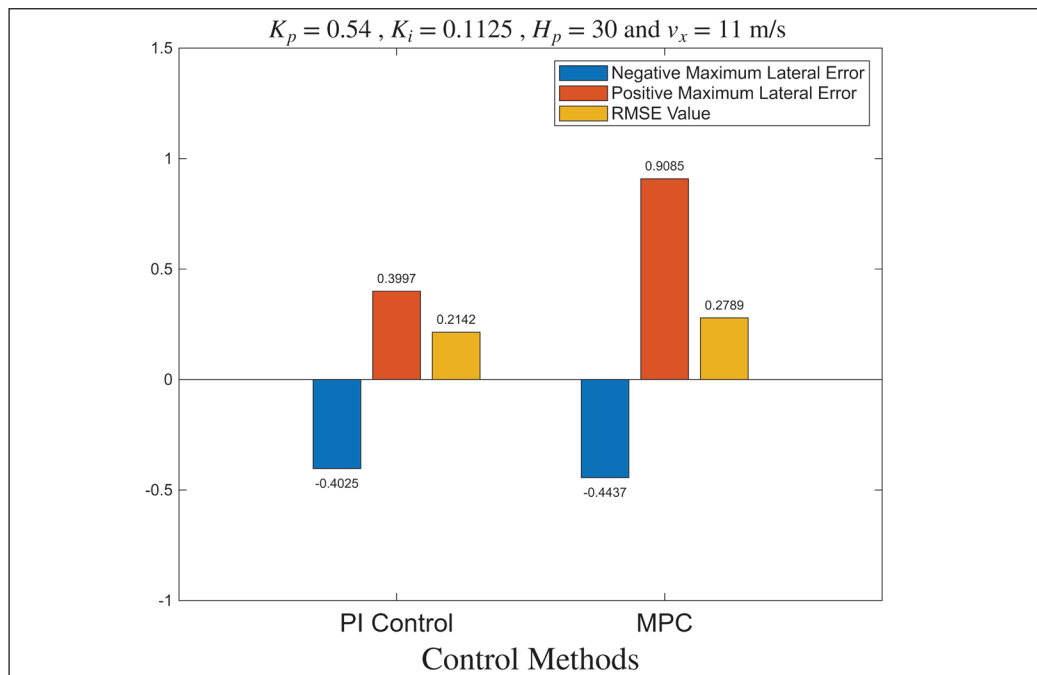


Figure 14. Lateral position error and steering angle with PI Control and MPC.

4.5. Scenario 5: Real-world Path Test

In this scenario, reference path is F1 Barcelona circuit. Located in Barcelona, Spain, Circuit de Barcelona is a world-famous racing track spanning 4.675 km. Known for its blend of high-speed straights and challenging technical corners, it frequently hosts prominent motorsport events. The circuit also plays a crucial role as a testing ground for advancing racing vehicle performance and development (Jiménez Elbal et al. 2024). F1 Barcelona circuit is shown in Figure 15.

Barcelona circuit path is created in the MATLAB Simulink environment using data available as open source on the web and the latest papers in the literature. The coordinate (0,0) is the starting point. Figure 16 shows the tracking performance of the model. The path's road boundaries are set with ± 5 m intervals. The vehicle's longitudinal speed was selected as $v_x = 20$ m/s, which corresponds to 72 km/h. $H_p = 30$ and $H_c = 10$. In this scenario, Q-tuned and H_c -tuned are applied to model together.



Figure 15: Formula 1 Barcelona circuit (Circuit de Catalunya 2025).

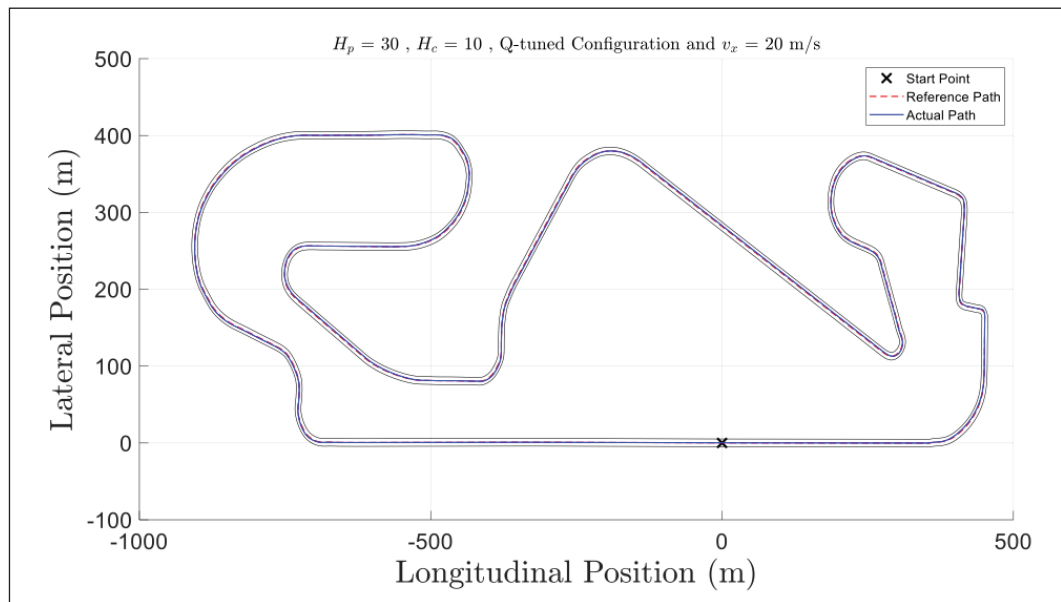


Figure 16. Tracking performance of the model.

Since the path is too long, it prevents us from seeing the accuracy of result. Therefore, details are in Figure 17 from points where the lateral position error is higher on the path. In addition, lateral position error values will also allow us to understand the results better.

Figure 18 shows lateral position error and steering angle results. It is observed that the system exhibits significant deviations in the reference path tracking at certain time intervals. When the steering angle responses are examined, it is seen that the system exhibits a very aggressive structure toward the control inputs.

The system initially gives low-amplitude responses, producing higher amplitude and sudden steering commands. When evaluated in general, the lateral error suppression performance of the current MPC configuration is not very good, making the control system increasingly aggressive. As a solution, the Q matrix's weights can be optimized, or the H_c value can be increased. These development possibilities will be considered in future studies. RMSE value of this scenario is 1.259 and the total simulation times in Scenario 5 was observed as $T_{F1} = 51.38$ seconds.

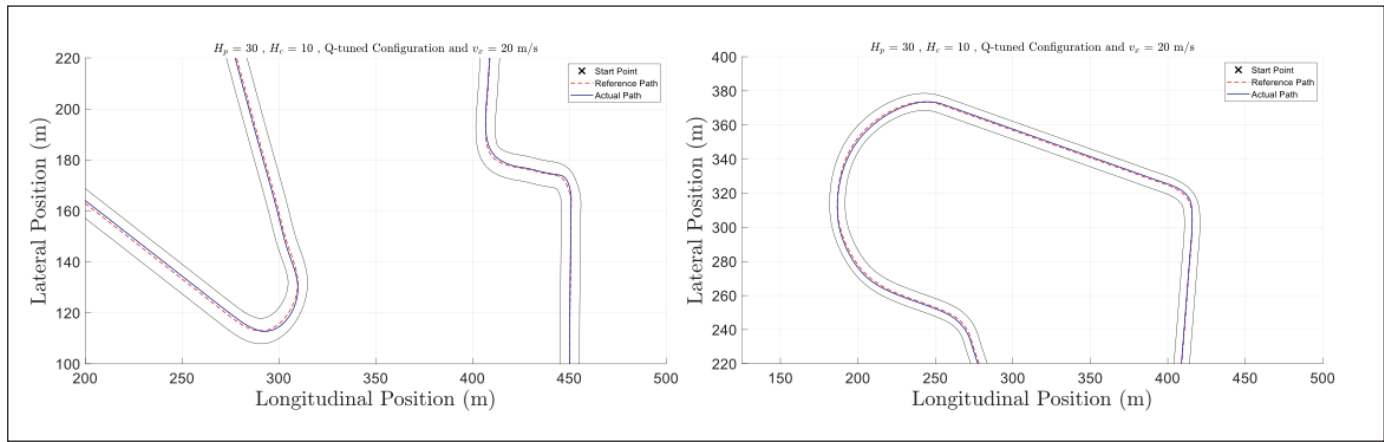


Figure 17. Details of actual path.

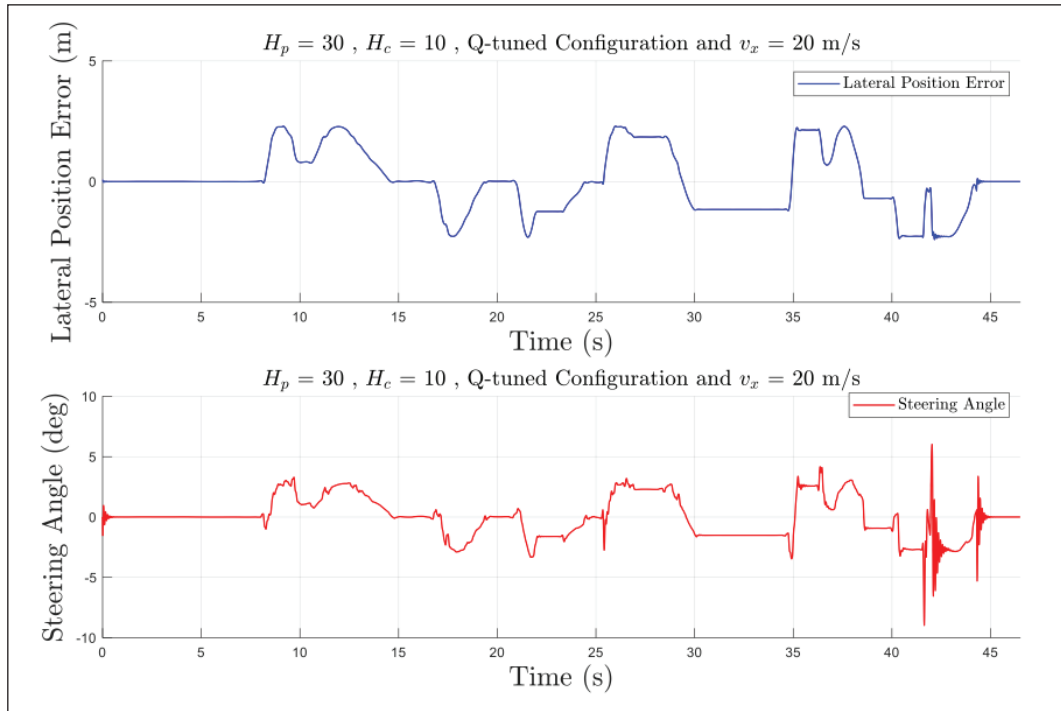


Figure 18. Lateral position error and steering angle of actual path.

Table 2 includes the simulation times for all scenarios.

Table 2. Simulation times of all scenarios

Test Scenario	Constant Parameters	Variable Parameters	Total Simulation
Scenario 1	Standart Configuration $v_x = 10$ m/s $H_c = 1$	$H_p = 10, 20, 30$	$T_{H_p,10} = 24.06$ sec $T_{H_p,20} = 33.47$ sec $T_{H_p,30} = 43.63$ sec
Scenario 2	Standart Configuration $H_p = 30$ $H_c = 1$	$v_x = 15, 25, 35$ m/s	$T_{v_x,15} = 43.91$ sec $T_{v_x,25} = 43.75$ sec $T_{v_x,35} = 43.59$ sec
Scenario 3	$v_x = 35$ m/s $H_p = 30$	Standart Configuration Q-tuned Configuration H_c -tuned Configuration	$T_{Standart} = 43.36$ sec $T_{Q-tuned} = 43.53$ sec $T_{H_c-tuned} = 44.63$ sec
Scenario 4	Standart Configuration $v_x = 11$ m/s $K_p = 0.54, K_i = 0.1125$ $H_p = 30, H_c = 1$	PI Control MPC	$T_{PI} = 0.012$ sec $T_{MPC} = 43.46$ sec
Scenario 5	Q-tuned Configuration $v_x = 20$ m/s $H_p = 30$ $H_c = 5$	-	$T_{FI} = 51.38$ sec

5. Conclusion

This study introduces path tracking for autonomous vehicles with MPC. A dynamic single-track vehicle model is used while creating the MPC algorithm, and it relies on this vehicle model. The path tracking controller is designed with these assumptions. Longitudinal speed is used as a constant. Lateral position is, therefore, necessary to follow the intended course. The model is tested with different constant speeds, and parameters are updated to obtain good performance for a wide range of speeds. Also, the model is tested with different prediction horizon and control horizon values. Another challenging topic is to create different scenarios for path tracking. Sinusoidal and F1 Barcelona circuits are the used paths. The algorithm also has good lateral position control performance under different constant longitudinal speeds and at different paths. Despite these positive results, the study does have some limitations. Furthermore, the dynamic structure of the vehicle imposes nonholonomic constraints, which prevent it from performing certain instantaneous motions, such as direct lateral displacement. Instead, such motions can only be achieved over time through feasible combinations of steering and longitudinal movement.

First, road–tire friction variations are not accounted for in the model, and these variations can have considerable impacts on the stability of the vehicle—especially when traveling at higher speeds or in adverse weather conditions. Second, increased prediction horizons lengthen the computational complexity of the MPC and potentially create challenges for real-time implementation on low-resource embedded systems.

In future studies, a dual-loop MPC architecture could modify the current assumption of constant longitudinal velocity. Under such a configuration, the inner loop would handle lateral path tracking under time-varying velocity references. In contrast, the outer loop would produce good tracking performance.

Another possible development in the future is the incorporation of real-time obstacle detection, avoidance, or emergency braking ability. LIDAR sensor data combined with the predictive model would allow one to update the feasible trajectory set inside the MPC horizon dynamically. The controller could then re-optimize the path responding to sensed hazards, ensuring collision-free path tracking. Lastly, adding tube-based or stochastic MPC techniques to the current MPC structure would increase its durability to

real-world uncertainties such as model mismatch, actuation delays, and sensor noise. These techniques would especially take into account limited disturbances and guarantee constraint satisfaction under uncertainty, enhancing safety and reliability for autonomous uses.

Yazar Katkıları: All authors contributed equally to all stages of this work.

6. References

- Aliskan, I. 2025.** The optimization-based fuzzy logic controllers for autonomous ground vehicle path tracking. *Engineering Applications of Artificial Intelligence*, 151, 110642. <https://doi.org/10.1016/j.engappai.2025.110642>
- Aliskan, I. 2019.** Adaptive model predictive control for Wiener nonlinear systems. *Iran Journal of Science and Technology, Transactions of Electrical Engineering*, 43(S1), 361–377. <https://doi.org/10.1007/s40998-018-0159-0>
- Aliskan, I. 2021.** Optimized inverse nonlinear function-based Wiener model predictive control for nonlinear systems. *Arabian Journal for Science and Engineering*, 46, 12403–12417. <https://doi.org/10.1007/s13369-021-05681-w>
- Cao, M., Hu, C., Wang, R., Wang, J., Chen, N. 2021.** Compensatory model predictive control for post-impact trajectory tracking via active front steering and differential torque vectoring. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 235(4), pp. 903–919. Doi: 10.1177/0954407020979087.
- Chowdhri, N., Ferranti, L., Iribarren, FS., Shyrokau, B. 2021.** Integrated nonlinear model predictive control for automated driving. *Control Engineering Practice*, 106. Doi: 10.1016/j.conengprac.2020.104654.
- “Circuit de Catalunya,”** CircuitCat. Accessed: Feb. 6, 2025. [Online]. Available: <https://www.circuitcat.com/en/>.
- Cui, Q., Ding, R., Wei, C., Zhou, B. 2020.** Path-tracking and lateral stabilisation for autonomous vehicles by using the steering angle envelope. *Vehicle System Dynamics*, 59(11), pp. 1672–1696. Doi: 10.1080/00423114.2020.1776344.
- Falcone, P., Borrelli, F., Asgari, J., Tseng, HE., Hrovat, D. 2007.** Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, 15(3), pp. 566–580.
- Gamez Serna, C., Lombard, A., Ruichek, Y., Abbas-Turki, A. 2017.** GPS-Based Curve Estimation for an Adaptive Pure Pursuit Algorithm. *Advances in Computational Intelligence*, pp. 497–511. Doi: 10.1007/978-3-319-62434-1_40.
- Hajiloo, R., Abroshan, M., Khajepour, A., Kasaiezadeh, A., Chen, SK. 2021.** Integrated steering and differential braking for emergency collision avoidance in autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(5), pp. 3167–3178. Doi: 10.1109/TITS.2020.2984210.
- Hu, J., Zhang, Y., Rakheja, S. 2020.** Path planning and tracking for autonomous vehicle collision avoidance with consideration of tire-road friction coefficient. *IFAC-PapersOnLine*, 53(4), pp. 13–18. Doi: 10.1016/j.ifacol.2020.11.003.
- Jiménez Elbal, A., Zarzuelo Conde, A., Siampis, E. 2024.** Simultaneous Optimisation of Vehicle Design and Control for Improving Vehicle Performance and Energy Efficiency Using an Open-Source Minimum Lap Time Simulation Framework. *World Electric Vehicle Journal*, 15(8), p. 366. Doi: 10.3390/wevj15080366.
- Laurense, VA., Gerdes, JC. 2021.** Long-Horizon Vehicle Motion Planning and Control Through Serially Cascaded Model Complexity. *IEEE Transactions on Control Systems Technology*. Doi: 10.1109/TCST.2021.3056315.
- Ławryńczuk, M. (2013).** Practical nonlinear predictive control algorithms for neural Wiener models. *Journal of Process Control*, 23(5), 696–714. <https://doi.org/10.1016/j.jprocont.2013.02.004>
- Li, L., Li, J., Zhang, S. 2021.** Review article: State-of-the-Art trajectory tracking of autonomous vehicles. *Mechanical Sciences*, 12(1), pp. 419–432. Doi: 10.5194/ms-12-419-2021.
- Li, S., Li, Z., Yu, Z., Zhang, B., and Zhang, N. 2019.** Dynamic trajectory planning and tracking for autonomous vehicle with obstacle avoidance based on model predictive control. *IEEE Access*, 7: 132074–132086. <https://doi.org/10.1109/ACCESS.2019.29407587>
- Mata, S., Zubizarreta, A., Pinto, C. 2019.** Robust Tube-Based Model Predictive Control for Lateral Path Tracking. *IEEE Transactions on Intelligent Vehicles*, 4(4), pp. 569–577. Doi: 10.1109/TIV.2019.2938102.
- Peng, H., Wang, W., An, Q., Xiang, C., Li, L. 2019.** Path Tracking and Direct Yaw Moment Coordinated Control Based on Robust MPC with the Finite Time Horizon for Autonomous Independent-Drive Vehicles. *IEEE Transactions on Vehicular Technology*, 69(6), pp. 6053–6066. Doi: 10.1109/TVT.2020.2981619.
- Qin, SJ., Badgwell, TA. (2003).** A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764. [https://doi.org/10.1016/S0967-0661\(02\)00186-7](https://doi.org/10.1016/S0967-0661(02)00186-7)
- Rajamani, R. 2012.** *Vehicle Dynamics and Control*, 2nd ed., Mechanical Engineering Series. Springer. Doi: 10.1007/978-1-4614-1433-9.

- Ritschel, R., Schrödel, F., Hädrich, J., Jäkel, J. 2019.** Nonlinear Model Predictive Path Following Control for Highly Automated Driving. IFAC-PapersOnLine, 52(8), pp. 350-355. Doi: 10.1016/j.ifacol.2019.08.112.
- Rokonuzzaman, M., Mohajer, N., Nahavandi, S., Mohamed, S. 2020.** Learning-based Model Predictive Control for Path Tracking Control of Autonomous Vehicle. IEEE Transactions on Systems, Man, and Cybernetics: Systems, pp. 2913-2918. Doi: 10.1109/SMC42975.2020.9283293.
- Tatjewski, P., Ławryńczuk, M. (2020).** Algorithms with state estimation in linear and nonlinear model predictive control. Computers & Chemical Engineering, 143, 107065. <https://doi.org/10.1016/j.compchemeng.2020.107065>
- Wang, H., Liu, B. 2021.** Path Planning and Path Tracking for Collision Avoidance of Autonomous Ground Vehicles. IEEE Systems Journal. Doi: 10.1109/JSYST.2021.3085479.
- Yakub, F., Mori, Y. 2015.** Comparative study of autonomous path-following vehicle control via model predictive control and linear quadratic control. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 229(12), pp. 1695-1714. Doi: 10.1177/0954407014566031.
- Yu, Y., Li, Y., Liang, Y., Zheng, L., Ren, Y. 2021.** Decoupling motion tracking control for 4WD autonomous vehicles based on the path correction. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 236(1), pp. 99-108. Doi: 10.1177/09544070211015928.
- Zou, K., Cai, Y., Chen, L., Sun, X. 2021.** Event-triggered nonlinear model predictive control for trajectory tracking of unmanned vehicles. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. Doi: 10.1177/0954407021992165.