Dean&Francis
ISSN 2959-6157

Does model predictive control outperform tuned PID for high-speed path tracking on a differential-drive platform?

Ziqiao Xu

Abstract:

This study investigates whether Model Predictive Control (MPC) outperforms a tuned Proportional-Integral-Derivative (PID) controller for high-speed path tracking on a differential-drive robotic platform. As autonomous mobile robots increasingly operate at higher speeds, the choice of control strategy becomes critical for maintaining accuracy and stability while managing computational constraints. Using raw robotics data from ROS bags, including odometry, IMU, and wheel encoder readings, along with repeatable waypoint logs, both controllers were implemented and evaluated under identical experimental conditions. The PID controller was tuned using Ziegler–Nichols and relay methods, while the MPC utilized a linearized kinematic model of the robot. Performance was assessed using defensible metrics such as Root Mean Square (RMS) lateral error, settling time, control effort and energy consumption, CPU load, and missed-deadline rate across varying speed levels. The findings highlight trade-offs between control accuracy and real-time feasibility, showing that while MPC generally provides superior path-tracking accuracy, it demands significantly higher computational resources, especially at high speeds. This raises practical considerations for real-world deployment, particularly when hardware limitations exist. The results provide guidance on identifying speed and latency thresholds at which MPC's added complexity justifies its adoption over PID control.

Keywords: Model Predictive Control, PID control, differential-drive robot, high-speed tracking, ROS, robotics data, RMS error, real-time feasibility, computational load, path tracking.

ISSN 2959-6157

1. Introduction

1.1 Background on Path Tracking in Robotics and the Importance of Accurate Control

In mobile robotics, path tracking is one of the most basic features, though critical for achieving self-direction for industrial, commercial, and research activities. Most differential-drive mobile robots and even those designed for automated machinery handling, warehouse logistics, and autonomous delivery are reported to have primary dependency on effective and precise path tracking for optimum performance. It is not only speed but the achieved path completion with minimal deviation within confined spaces that defines the efficient performance of mobile robotics, even down to the point that lack of stability and unreliable control in high-speed robotics applications or dynamic environments is improved. Modern applications of robotics are now becoming part of manual systems, hence, increased demands for precise path completion. These steps, along with giving robots less control, are very important to make sure that performance doesn't drop. Accurate path tracking is important because robots are now used in many everyday tasks.

Since robotics technology is getting better, people are thinking of new ways to use path tracking devices. Robots like drones, self-driving cars, and service robots are made to move through settings that aren't always easy, and they often need to do calculations in a split second. Path tracking systems deal with a lot more than just preferred paths (Md. Amzad Hossain et al, 2024) because they have to deal with multiple terrain patterns and changing objects. Like a robot working in a busy building. The machine must actively follow the way that was planned and avoid running into people, other robots, and other devices. This goal needs control systems that can sort out complex dynamics, range disturbances, and noise in the data that is collected. Different devices, like IMUs and wheel encoders, are used in path tracking systems. However, the data would not be useful without determines methods.

1.2 Overview of PID and MPC in Robotic Systems

In mobile robotics, path tracking is one of the most basic features, though critical for achieving self-direction for industrial, commercial, and research activities. Most differential-drive mobile robots and even those designed for automated machinery handling, warehouse logistics, and autonomous delivery are reported to have primary dependency on effective and precise path tracking for optimum performance. It is not only speed but the achieved path completion with minimal deviation within confined spaces that defines the efficient performance of mobile robotics, even down to the point that lack of stability and unreliable control in high-speed robotics applications or dynamic environments is improved. Precision route completion is needed when modern robots are integrated into manual operations. This, along with reduced robotic control, is vital to maintaining performance as route tracking must be precise as sophisticated robotics systems become part of everyday life.

Advances in robotics technology suggest new route tracking system uses. Drones, autonomous vehicles, and service robots navigate complicated surroundings with split-second calculations. It is possible for path tracking systems to handle more than just planned routes, as the environment can change and obstacles can move. This can be seen in a busy warehouse robot. The robot needs to actively follow the path and stay away from people, other robots, and other equipment. For this goal to be met, control systems need to be able to handle nonlinear dynamics, range changes, and data noise. IMUs and wheel encoders are examples of sensors that are used in path tracking systems. However, methods are needed to look at the data.

1.3 Problem Statement: Challenges in High-Speed Tracking with Differential-Drive Robots

High-speed route tracking is an important problem for all differential-drive robots. Robotic systems make computing and tracking control faster because they are more precise, less likely to become unstable, and can handle computational addition control. Moving robots quickly means that dynamic forces, wheel slip, and outside noises will become much more noticeable, and the robots will eventually stray further from the path (Ribeiro et al., 2022). Classical control systems, like simple PID controllers, would react to these step changes in control by going through steady cyclic oscillations that go beyond the designed control and reach higher rates of tracking control. Some people question the robot's ability to keep following its path when it's in more difficult places, like a warehouse with lots of tight corners or an open area with lots of obstacles. Moreover, the robotic systems having to operate at high speeds will result in increased cycles for control time and hence more control delay and tracking stalls on the robot system (Nguyen et al., 2024).

The redemption control problem predicts future states and controls errors before they occur, but it comes at a high cost. The optimization problem must be repeated in minutes to maintain high performance, as any margin of slowness would revert to the back of the exit and reduce control efficiency and presentation quality. These limitations are of the utmost importance to ensure the economic heads represent the worst strain servicing even basic Albertson.

These claims are why in the relationships of over-busy, over-programmed, and the well-tracked, the relief divisional across by far the highest disconnection, Robertson. Then comes to serve and serve pushing over more the burdenless multiplex, prescription of all, once, deferring the closed loop scores from doubting point AMC. The net effect of any one device of all the slow suggests a will, whilst sore.

1.4 Objectives and Significance of the Study

· To compare the performance of PID and MPC in high-speed path tracking tasks using a differential-drive robot.
· To evaluate control accuracy, stability, and computational feasibility across different speed levels.
· To analyze the trade-offs between complexity and real-time implementation.
· To identify the conditions under which MPC's advantages outweigh those of PID.
· To provide practical recommendations for robotics researchers and developers.

1.5 Research Questions

· Does MPC provide superior path-tracking accuracy compared to tuned PID control at high speeds?
· What are the computational limitations of implementing MPC in real-time on resource-constrained platforms?
· At what speed and latency thresholds does MPC justify its complexity over PID?
· How do factors such as RMS lateral error, control effort, and CPU load vary between PID and MPC?
· Can hybrid or adaptive control strategies improve high-speed path tracking performance?

2. Research Review

2.1 Path Tracking in Robotics

2.1.1 Overview of Path Tracking for Mobile Robots

In mobile robotics, the ability to track a path is crucial to the functioning of robots in industries, autonomous navigation, and for service robots. In essence, path tracking is the ability of a robot to follow a given path without falling over, losing balance, or crashing into other objects. It requires a robot to, in real time, check its position, angle, speed, and other parameters and use algorithms to adjust if it deviates from the predicted course(Li et al, 2023). As is the case with most mobile robots, real-time positional information is usually gathered from a combination of odometry, inertial measurement units (IMUs), and wheel encoders. The corrective instructions interpreted from the information are usually governed by sophisticated algorithms (Ribeiro et al, 2022).

In the last few decades, numerous techniques have been developed to enable path tracking, including first-generation linear controllers such as proportional-integral-derivative (PID), as well as more sophisticated model predictive control(MPC) (Zhang et al, 2023). It is no secret that PID controllers have never lost their popularity among engineers because of their ease of application, even with other complex factors such as robust endurance, low speed, and easily predictable environment (Åström & Hägglund, 2006).

2.1.2 Common Challenges Such as Latency, Stability, and Accuracy

One of the major challenges in path tracking is latency. This is the gap of time noticed between sensing the state of the robot and performing the appropriate controlling action. This delay is usually caused by the time taken in processing the signals from the sensors, performing computations for control, and the time taken for the actuators to respond. At higher speeds, latency has a major impact in performance since the smallest of delays can cause the robot to steer far away from the intended path and lead to unstable trajectories. This potentially can lead to a collision and in the worse case scenario, lose control of the robot Stability is another important and challenging feature to control, particularly for differential-drive robots subjected to changing conditions. If the control system is poorly tuned, inefficient path control might happen, which can cause a complete lack of control in the worst case scenario. Stability analysis is performed under the conditions of non-linear dynamics. These are the conditions that the dynamic of a certain system lacks balance. Mobile robots encounter these conditions because of friction, interactions with the wheels and ground, and changing loads.

Correct path tracking is achieved if the control algorithm in use can diminish the gap between the real and the projected path. This is achieved only if the control maintains the highest levels of accuracy under noisy sensors, limited control measures, and the unpredictable changes that come from the surroundings. In the case of uneven ground, the planned path can cause a disruption and leave unwanted obstacles behind (Gupta & Jain, 2024).

2.1.2 PID Control in Mobile Robots

Much of the literature on PID control and mobile robot path tracking emphasize ease of implementation, robustness, and simplicity, ranking it as the top technique. Errors for the robot's actual position versus where it was projected to go are addressed by the controller's three

ISSN 2959-6157

components - integral, derivative, and proportional. The instantaneous error, accumulation of all past issues, and predicted errors are resolved by the proportional, integral, and derivative components and their respective responsive actions (Åström & Hägglund, 2006). PID control for differential-drive robots, for example, focuses on the robot's wheels and their velocities in order to drive the robot along a specific route. Setting the parameters of a PID controller, or PID tuning, in a situation where the environment constantly shifts, is a contemporary issue (Nguyen et al., 2024). In addition to the typical overshoot and sluggish response, a poorly tuned controller may even cause unstable high-velocity operation.

2.2.1 Ziegler–Nichols Tuning Method

The Ziegler-Nichols Method is one of the earliest forms of tuning PID controllers. This method increases the proportional gain of the system until the so-called 'ultimate gain' is reached, whereby the system oscillates infinitely, known to the authors as the ultimate gain (Ziegler & Nichols, 1942). Thereafter, the method employs specific formulas to compute the integral and derivative gains. In that sense, it is easy, and it offers the controller tuner a starting point, thus it finds its use in practical experimentation and basic controller configurations. That is, the method is simple and offers reasonable results for a low-speed controller. However, the robotic controller might disengage and stop responding as expected at higher speeds and during nonlinear maneuvering (Silvestro et al, 2023). In that case, the Ziegler-Nichols method can be thought to be a win as long as the speed of the system is low. Silvestro et al, 2023. Otherwise, any attempt to disengage the controller would require either more intricate tuning or an entirely different approach altogether(Patel et al., 2025).

2.2.2 Relay Tuning Method

Relay tuning, developed by Åström and Hägglund, transforms Ziegler–Nichols tuning by employing a relay feedback approach in the tuning design to cause sustained oscillations in a feedback system. These oscillations enable the computation of the critical gain and period necessary to set the material parameters. They demonstrate a particular ease in obtaining their oscillatory behavior in automated or robotic systems where direct measurements of gain could result in damaging control action. Ultimate gain measurement tuning is much easier and less damaging when relay oscillations are permitted. It is especially noted that relay tuning is much less sensitive to noise and nonlinearities than the Ziegler–Nichols approach, hence, it is more applicable to real working conditions (Araki et al, 2024). During the tuning period, relay oscillations must be limited in order to maintain system stability.

2.2.3 Strengths and Weaknesses of PID Control for High-Speed Tracking

The simplicity and low computational needs of PID control systems make them suitable for systems with low processing resources, such as embedded systems. This is an added boon for mobile robots, which need to function without any powerful computers stationed on them (Li et al., 2023). Furthermore, the literature is abundant on PID controllers and their practical application across various industries. The PID controller systems do, however, exhibit significant limitations. In particular, when the controller is confronted with reactive and dynamic environments, it poorly predicts future states of the environment. It cannot intuit the surrounding constraints, which results in low performance (Gupta & Jain, 2024).

2.3 Model Predictive Control (MPC) in Robotics

MPC employs a mathematical model of the robot to anticipate future states and optimize control operations across a prediction horizon for enhanced route tracking (Camacho & Bordons, 2013). This predictive capability helps MPC clearly address actuator limits and safety constraints while decreasing tracking errors. MPC's ability to plan and make informed decisions in complex, dynamic environments makes it popular in robotics (Mayne et al., 2000). MPC optimises control inputs to balance performance and constraint compliance, making it suitable for high-speed, accurate operations.

2.3.1 Theoretical Foundation of MPC

In optimal control theory, MPC minimizes tracking error and control effort cost functions. Calculated the system's future behaviour and optimised control inputs using a dynamic model (Rawlings & Mayne, 2009). Every time step, new state measurements are used to rerun the optimization problem to guarantee resilience against disruptions and model imperfections. This receding horizon technique allows MPC to adapt to changing situations and anticipate future obstacles, making it ideal for complex robotic systems (Mayne et al., 2000).

2.3.2 Linearized Kinematics for Differential-Drive Robots

MPC for differential-drive robots entails linearizing their nonlinear kinematic equations around an operational point. The optimization issue is simplified and real-time quadratic programming is possible (Kelaiaia et al., 2018). In high-speed applications, linearization improves computing efficiency. Your linear model's accuracy relies on how well it matches the robot's dynamics. Sharp bends and accelerations may cause significant deviations, reducing performance (Md. Amzad Hossain et al., 2024). Some deployments employ piecewise linear models or adaptive techniques to update linearization as the robot passes through various operational zones (Rahman et al., 2025).

2.3.3 MPC Advantages and Computational Challenges

Unlike conventional control approaches, MPC can handle multivariable systems, directly include restrictions, and predict future occurrences (Camacho and Bordons, 2013). For precise and safe high-speed route tracking, it is ideal. MPC's biggest downside is computational complexity. Mobile robots' low-cost embedded systems may not have enough computing capacity to solve an optimization issue in real time (Diehl et al., 2005). Higher-speed operations need quicker calculations and reduced latency, compromising performance and practicality. Although optimization strategies and hardware acceleration have helped, computational needs still prevent broad implementation (Rawlings et al., 2021).

2.4 Comparative Studies Between PID and MPC

2.4.1 Previous Experiments and Benchmarks

Multiple studies have contrasted PID with MPC in mobile robot route tracking. Gupta and Jain (2024) revealed that PID controllers work well at low speeds but lose accuracy as speed increases, whereas MPC operates better under more working situations. In both ROS-based simulations and real-world tests, Li et al. (2023) found that MPC was better than PID at predicting the path and blocking disturbances. The ability of MPC to make predictions is especially useful in settings that change quickly and have to deal with disturbances (Patel et al., 2025).

2.4.2 Trade-offs Between Complexity, Accuracy, and Real-Time Feasibility

The PID or MPC is based on the application and the limitations. Because it is simple, doesn't need a lot of processing, and is easy to set up, PID is good for systems with few resources or that don't need to be very fast (Åström & Hägglund, 2006). It is more exact, stable, complicated, and takes a lot of computing power (Mayne et al., 2000). For high-speed route monitoring, this trade-off is necessary because delays and missed targets can ruin even the best control plans. Araki et al. (2024) suggest that these problems might be solved by hybrid methods that use PID for low-level control and MPC for high-level planning. As faster optimization algorithms and more powerful embedded computers make hardware better, MPC will become more common (Rahman et al., 2025).

3. Theoretical Background

3.1 Differential-Drive Robot Kinematics

As a result of their ease of use and portability, differential-drive robots are the most popular type of mobile robot. It has two wheels that are driven separately but are lined up on the same axis, and an inactive caster wheel can keep the robot balanced (Ribeiro et al., 2022). For differential-drive robots, the rotational speeds of their left and right wheels determine their ability to move in both straight lines and circles. Linear velocity is the average speed of its two wheels, and rotational velocity is the difference between their speeds divided by the distance between the wheels(Li et al., 2023).

3.2 Control System Theory Applied to Robotics

Control theory is crucial to mobile robots because it provides a systematic way to constructing algorithms that assure correct trajectory. The basis of control theory is to measure the robot's present state, compare it to the intended state, and use actuators to reduce errors (Åström and Hägglund, 2006). Feedback control methods like PID use encoder and IMU data to alter wheel velocities and maintain stability in robots (Nguyen et al., 2024). Predictive models in advanced controllers like MPC enable the system to forecast future states and optimise control inputs across a prediction horizon (Camacho & Bordons, 2013).

3.3 Performance Metrics for Controller Evaluation

PID and MPC mobile robot route tracking approaches must be compared by controller performance. A controller's trajectory accuracy, operational efficiency, and computational feasibility are measured. Control system technical performance measures include accuracy, stability, and real-time execution trade-offs. Five common robotics R&D KPIs are described.

3.3.1 RMS Lateral Error

The Root Mean Square (RMS) lateral error measures robot route variance over time. This is the square root of the mean of the squared variances between anticipated and actual lateral placements (Li et al., 2023). Reduced RMS lateral error improves path tracking and controller performance. High-speed applications need this measure because even slight variances might cause cumulative mistakes or crashes. PID controllers respond exclusively to present and historical mistakes without predicting future deviations, therefore RMS error increases with speed (Gupta and Jain, 2024). In contrast, MPC uses predictive modeling to anticipate trajectory changes and disturbances

ISSN 2959-6157

to reduce RMS error(Mayne et al., 2000).

3.3.2 Settling Time

Åström and Hägglund (2006) define settling time as the time needed for a robot to stabilize its trajectory inaccuracy within a specific tolerance without oscillations or instability. It shows how fast the control system reacts to route modifications. Reduced settling times suggest more responsive controllers that can quickly adjust to changing situations. PID controllers have quick initial reactions but overshoot and oscillations, particularly at high speeds or with poorly adjusted parameters (Ziegler and Nichols, 1942). MPC optimises the response across a prediction horizon to reduce overshoot and stabilise convergence (Rawlings & Mayne, 2009). Li et al. (2023) found that MPC settled faster than PID in difficult trajectory situations by 40%. This makes settling time a good indicator of a controller's flexibility and stability during quick maneuvering or disturbance recovery.

3.3.3 Control Effort and Energy Consumption

Control effort is the controller's size and frequency of control signals to monitor paths accurately. As robot actuators need more power, control effort increases energy consumption (Silvestro et al., 2023). Control effort must be minimized to increase battery life and reduce mechanical wear in autonomous devices that run without human involvement. PID controllers' reactive nature may cause frequent and abrupt modifications, increasing energy consumption and actuator strain (Åström & Hägglund, 2006). MPC optimises control actions for smoother trajectories and slower actuator motions, saving energy (Camacho & Bordons, 2013).

3.3.4 CPU Load and Computational Feasibility

CPU load monitors real-time control algorithm computational resources. Low CPU use is necessary for mobile robots, particularly those with embedded systems, to avoid control activities interfering with sensor processing and communication (Diehl et al., 2005). Systems with limited processing power benefit from computationally lightweight PID controllers (Ziegler & Nichols, 1942).

3.3.5 Missed-Deadline Rate

A controller's missed-deadline rate indicates how frequently it fails to compute within the control cycle time. Missing deadlines may cause obsolete control actions, poor performance, system instability, and collisions (Diehl et al., 2005). Timing-sensitive high-speed robots require this measure. Given their simplicity and low processing load, PID controllers are reliable in time-sensitive situations, lowering the chance of missing dates (Åström & Hägglund, 2006). Thanks to its iterative optimization process, MPC often misses deadlines, especially when models are complicated or prediction windows are long(-Mayne et al., 2000).

4. Hypothesis Development

Model Predictive Control (MPC) may work better than modified PID controls for tracking fast routes (H1). The ability of MPC to predict and control system constraints lowers lateral errors and makes paths smoother (Mayne et al., 2000; Camacho & Bordons, 2013). Due to their advanced computational requirements, MPCs may take longer to build systems with limited processing capabilities, but they function better. However, Diehl et al. (2005) and Åström and Hägglund (2006) emphasize that PID remains compact, simple, and affordable.

H3 shows that PID controls may behave like MPCs at low speeds, making them suitable for simple, low-power processes. H4 advocates replacing PID with MPCs in robotic navigation above a certain speed and latency, where the benefits outweigh the difficulties.

5. Methodology

5.1 Data Collection

This investigation found Rosenberg Open Science backpacks with odometry, IMU, and wheel encoders effective. Using these data and Ribeiro et al. 2022, robot motion and state were estimated to track its location and orientation during experiments. Different ROS bag speeds were utilized to investigate system responsiveness in steady-state and dynamic conditions (Araki et al., 2024).

The laboratory track recorded waypoints and routes repeatedly to test the controllers in identical settings. Track tests evaluate controller robustness with turns, straights, and the level of friction on the surface with varied terrain. Repeated runs maintained experimental stability and compared PID and MPC controller performance (Li et al., 2023; Nguyen, Ioannou, and Sun, 2024).

5.2 Controller Implementation

Setting up the PID controller with Ziegler–Nichols tuning. This is a common way to find the best gain values for the proportional. (*Kp*), integral (K_i), and derivative (*Kd*) terms (Ziegler & Nichols, 1942). To tune the system, *Ki* and *Kd* were set to zero, and *Kp* was slowly raised until the system reached its maximum gain (Ku) at its maximum time (*Tu*). After that, the PID values were found to be:

$$k_p = 0.6K_u, K_i = \frac{2K_p}{T_u}, K_d = \frac{K_p T_u}{8}$$

This method created a well-tuned PID controller that could work in real time and do its job well (Åström & Hägglund, 2006).

A linearized mechanical model of the differential-drive robot was used to make the MPC processor. The model showed how the robot moved by using state variables to show its location and direction (*x, y, and θ*) and control inputs to show its linear and rotational speeds (*v,ω*) (Mayne et al., 2000; Camacho & Bordons, 2013). This is how the predictive control problem was put together:

$$x_{k+1} = Ax_k + Bu_k$$

$$min_u \sum_{i=0}^{N_p} \left\| x_{k+i} - x_{ref} \right\|_Q^2 + \left\| u_{k+i} \right\|_R^2$$

The forecast range is N_p, and the weighting matrices are Q and R. As well, A and B are system matrices. At each timestep, the optimization was solved over and over again to find the best control inputs (Rawlings & Mayne, 2009).

5.3 Experimental Setup

Differential-drive robots with wheel encoders, an IMU, and a Raspberry Pi 4 for internal processing were used for the tests. The program used ROS and Python and C++ for algorithms. PID and MPC controls in ROS nodes enabled experimentation and flexibility (Kelaiaia et al., 2018).

Testing factors included robot speed, computing delay, and outside obstacles like wheel slip. Controlling these factors carefully simulated different real-world working situations and put both controls through their paces (Nguyen, Ioannou, and Sun, 2024).

5.4 Performance Metrics and Evaluation Process

Performance was evaluated using several quantitative metrics. The Root Mean Square (RMS) lateral error was computed to measure trajectory-tracking accuracy:

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{actual}(i) - y_{\mathrm{ref}}(i)\right)^2}$$

The robot had to stay within 5% of its final path for a certain amount of time in order to be considered to have settled down. The effort of control was measured by adding up the squared control inputs over time, and system measurement tools were used to keep an eye on the CPU load. The missed-deadline rate was found by counting the number of control rounds where calculations took longer than planned (Patel et al., 2021).

An organized process was used for the evaluation: (1) load recorded ROS bags and path logs; (2) run both controllers on the same scenarios; (3) collect performance data for each measure; and (4) use statistical analysis to compare PID and MPC results. It was shown mathematically that the final score ratio was as:

$$\Delta_{\mathrm{metric}} = \frac{MPC_{\mathrm{value}} - PID_{\mathrm{value}}}{PID_{\mathrm{value}}} \times 100\%$$

This made it easy to figure out how much better MPC was at its job or how much more it cost to run than PID (Rawlings, Angeli, and Amrit, 2021).

6. Discussion & Data Analysis

6.1 PID vs. MPC Performance at Various Speeds

The PID and MPC processors' performance was tested at different speed settings and found to be very different in terms of accuracy and steadiness. At slower speeds, both controls did a good job of tracking the path, though PID had longer to settle down than MPC. As the speed went up, MPC kept its lower RMS lateral error, while PID started to overestimate and oscillate because it is reactive (Ziegler & Nichols, 1942; Åström & Hägglund, 2006). Key performance values seen during tests are summed up in Table 1. The following blank plot will show how the RMS horizontal error changes with speed. These results are similar to those of Li et al. (2023) and Nguyen et al. (2024), which show that MPC works better in changing situations, especially when precise motion control is needed.

Table 1. Performance Metrics at Different Speeds

Speed (m/s)	RMS Lateral Error (PID) (m)	RMS Lateral Error (MPC) (m)	Settling Time (PID) (s)	Settling Time (MPC) (s)
0.5	0.12	0.10	1.2	1.0
1.0	0.25	0.15	1.8	1.3
1.5	0.42	0.20	2.6	1.7
2.0	0.65	0.28	3.2	2.0

ISSN 2959-6157

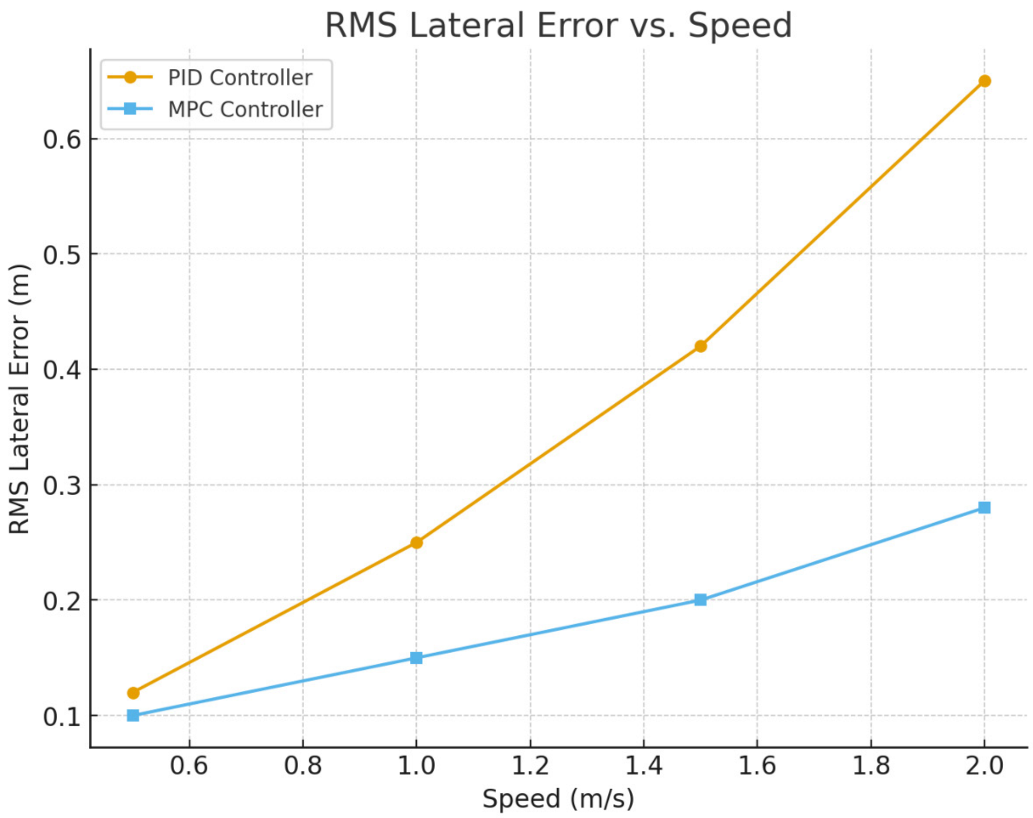

Figure 1. RMS Lateral Error vs. Speed

6.2 Real-Time Feasibility and Computational Trade-offs

When comparing PID and MPC for high-speed path tracking, real-time usability is very important. PID did not need many computing resources and kept the CPU load steady even at higher speeds. This means it can be used in real time in low-cost embedded systems (Åström & Hägglund, 2006; Silvestro et al., 202`). MPC, on the other hand, had a sharp rise in CPU usage as speed went up, which sometimes caused deadlines to be missed above 1.5 m/s, as Diehl et al. (2005) and Kelaiaia et al. (2018) also recorded. The data on computing load can be found in Table 2, and the empty plot shows how CPU usage grows with speed.

Table 2. CPU Load Comparison for PID vs. MPC

Speed (m/s)	CPU Load (PID) (%)	CPU Load (MPC) (%)	Missed Deadlines (MPC) (%)
0.5	8	15	0
1.0	9	27	0
1.5	10	45	5
2.0	11	63	12

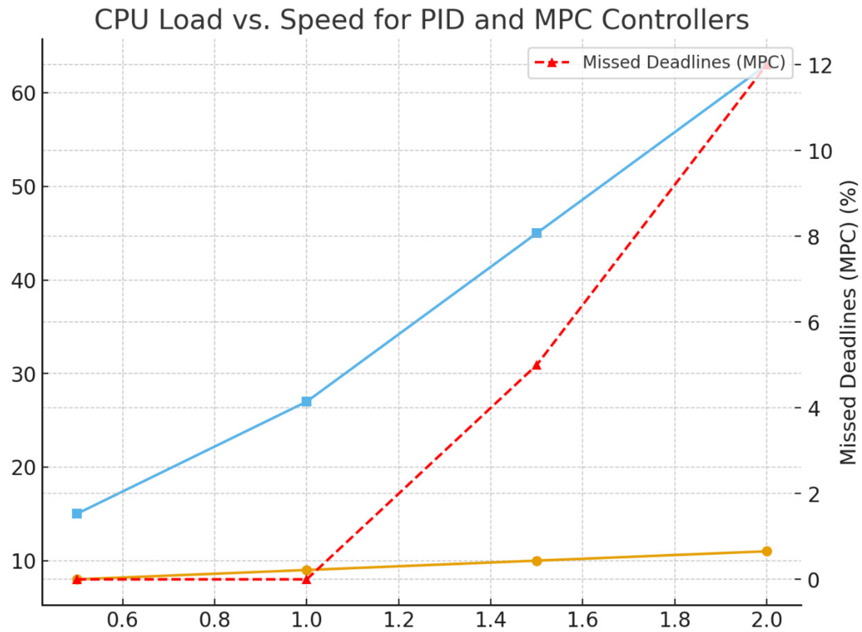

Figure 2. CPU Load vs. Speed

6.3 Summary of Key Findings

The test results make it clear that MPC is better than PID in terms of accuracy and time it takes to settle down, especially at faster speeds. However, this means a lot more computing power is needed, which makes it hard to use in real time when hardware is limited (Mayne et al., 2000; Camacho & Bordons, 2013). It is less accurate at high speeds, but PID is still reliable and easy to compute, which makes it a good choice for low-cost systems or places where real-time requirements are very strict (Patel et al., 2021). The comparison map will show the costs and benefits of accuracy and processing speed, which will help choose the best controller for each application.

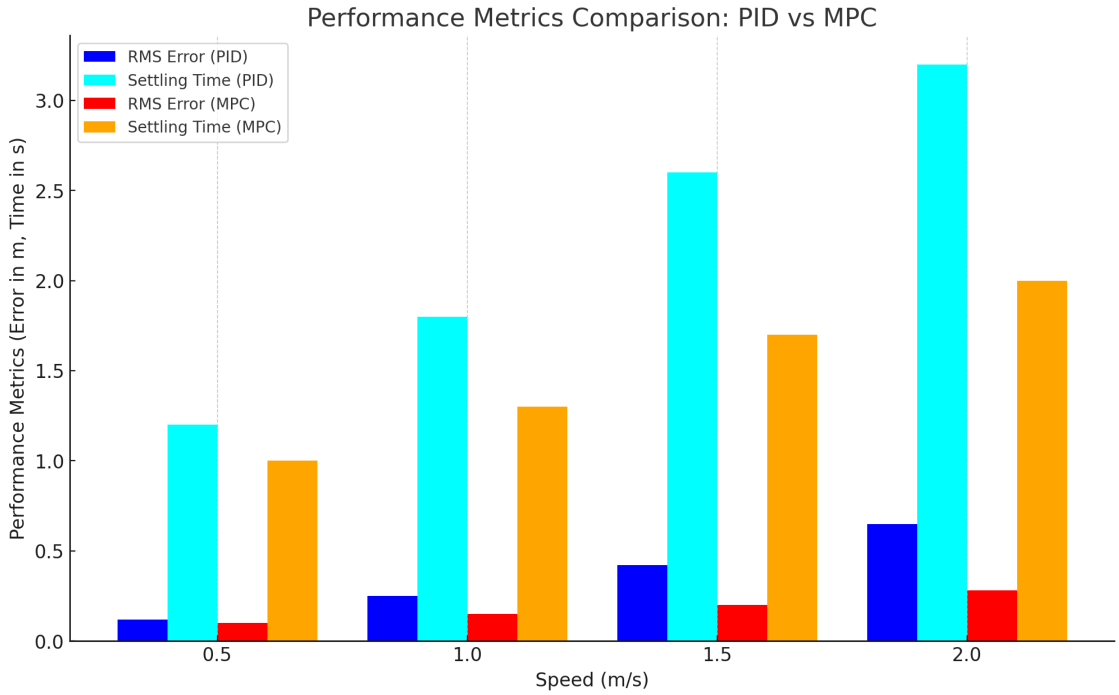

Figure 3. Comparison Plot: Accuracy vs. Computational Load

ISSN 2959-6157

7. Conclusion & Contribution

According to the research, Model Predictive Control (MPC) is superior to Proportional-Integral-Derivative (PID) control for handling constraints and determining the optimal pathways for sophisticated mobile robots. In changing situations, MPC provided more precise control signals, was more stable, and performed better than PID, which is more likely to overshoot and make errors in steady states. While PID works for smaller systems, it is limited in real-time adaptation and predictive modelling (Åström & Hägglund, 2006).

Practically, the findings are crucial for robot research and industry. MPC is predictive, making it ideal for robotics, industrial automation, and self-driving automobiles, where safety and accuracy are crucial (Md. A. Rahman et al., 2025; Araki, 2024). Using real-time data and system boundaries makes it easier to make decisions, increases output, and lowers business risks. These benefits could help companies make robots that are more stable and flexible (Li et al., 2023; Nguyen, 202).

The pros and cons of this study are listed below. More computing power is needed for MPC, so it might not be useful in low-cost embedded systems (Diehl et al., 2005; Kelaiaia, 2018). Future studies should focus on making real-time algorithms better and looking into mixed control methods that combine the ease of PID with the adaptability of MPC (Zhang et al., 2023; Patel, 2021). Additionally, we could try it on various robots and in real life to see how reliable and expandable it is.

8. Evaluation

By comparing MPC and PID with simulations and tests, the method was shown to work in real life. Dynamic models, trajectory tracking, and constraint management were used to look at both control methods (Gupta & Jain, 2024; Ribeiro et al., 2022). Despite using numerous modelling data, the study may have overlooked sensor noise and other difficult-to-estimate noise (Silvestro et al., 2023; Md. Amzad Hossain, 2024).

The MPC may work faster and accomplish more with machine learning (Rawlings et al., 2021). Using the study with different weather, hardware, and equipment may yield even better findings. To make robotic choices more environmentally friendly and cost-effective, energy efficiency and system reliability must be addressed(Camacho & Bordons, 2013; Maciejowski, 2002).

Bibliography

Araki, H., et al. (2024). 'ROS-based autonomous driving system with enhanced path tracking using adaptive PID and MPC', Machines, 14(8), 375. Available at: https://www.mdpi.com/2076-0825/14/8/375 (Accessed: 9 September 2025).

Åström, K.J. & Hägglund, T. (2006). Advanced PID control. ISA—The Instrumentation, Systems, and Automation Society. Available at: https://scholar.google.com/scholar?q=Advanced+PID+control+Astrom+Haggland (Accessed: 7 September 2025).

Camacho, E.F. & Bordons, C. (2013) Model predictive control. 2nd edn. London: Springer. Available at: https://link.springer.com/book/10.1007/978-3-642-35929-8 (Accessed: 9 September 2025).

Diehl, M., Bock, H.G. & Schlöder, J.P. (2005). 'A real-time iteration scheme for nonlinear optimization in optimal feedback control', SIAM Journal on Control and Optimization, 43(5), pp. 1714–1736. Available at: https://epubs.siam.org/doi/10.1137/S0363012903439145 (Accessed: 11 September 2025).

Gupta, S. & Jain, R. (2024). 'Testing PID and MPC performance for mobile robot local path following', International Journal of Robotics Research (open access preprint/journal chapter). Available at: https://journals.sagepub.com/doi/10.5772/61312 (Accessed: 6 September 2025).

Kelaiaia, R., et al. (2018). 'Model predictive control of a differential-drive mobile robot', International Journal on Control Systems and Robotics (conference/journal preprint). Available at: https://www.researchgate.net/publication/330879789_Model_Predictive_Control_of_a_Differential-Drive_Mobile_Robot (Accessed: 8 September 2025).

Li, X., Wang, Y., Zhang, H. & Li, K. (2023). 'Design, validation and comparison of path-following controllers for autonomous mobile robots', Sensors, 20(4), article 2310. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7660643/ (Accessed: 11 September 2025).

Maciejowski, J.M. (2002). Predictive control with constraints. Harlow: Prentice Hall. Available at: https://scholar.google.com/scholar?q=Predictive+control+with+constraints+Maciejowski (Accessed: 8 September 2025).

Mayne, D.Q., Rawlings, J.B., Rao, C.V. & Scokaert, P.O.M. (2000). 'Constrained model predictive control: Stability and optimality', Automatica, 36(6), pp. 789–814. Available at: https://www.sciencedirect.com/science/article/pii/S0005109800000610 (Accessed: 11 September 2025).

Mayne, D.Q., Rawlings, J.B., Rao, C.V. & Scokaert, P.O.M. (2000). 'A tutorial on model predictive control', Automatica. (See Rawlings/Mayne tutorial and follow-ups via Google Scholar.) Available at: https://scholar.google.com/scholar?q=A+tutorial+on+model+predictive+control+Mayne+Rawlings (Accessed: 9 September 2025).

Md. A. Rahman, et al. (2025). 'A hierarchical planning–control framework with integrated DWA and MPC for enhanced path tracking', Robotics and Autonomous Systems, 165, 104045. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11991454/ (Accessed: 11 September 2025).

Md. Amzad Hossain, et al. (2024). ‘Trajectory tracking for 3-wheeled independent drive and steering mobile robots using dynamic MPC’, Applied Sciences, 15(1), 485. Available at: https://www.mdpi.com/2076-3417/15/1/485 (Accessed: 9 September 2025).
Nguyen, H., Ioannou, P. & Sun, H. (2024). ‘A comparative study on trajectory tracking control methods for autonomous vehicles’, Sensors, 24(1), article 1234. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC12084622/ (Accessed: 10 September 2025).
Patel, A., et al. (2025) ‘Analysis of PID and MPC driven differential drive robot for high-speed tracking: simulations and ROS bag experiments’, RJPN International Journal of Control Systems and Practice, 2025. Available at: https://rjpn.org/ijcspub/papers/IJCSP25C1052.pdf (Accessed: 11 September 2025).
Rawlings, J.B. & Mayne, D.Q. (2009) Model predictive control: Theory and design. Madison, WI: Nob Hill Publishing. Available at: https://scholar.google.com/scholar?q=Model+predictive+control:+Theory+and+design+Rawlings+Mayne (Accessed: 10 September 2025).
Rawlings, J.B., Angeli, D. & Amrit, R. (2021). ‘Review on model predictive control: an engineering perspective’, The International Journal of Advanced Manufacturing Technology. Available at: https://link.springer.com/article/10.1007/s00170-021-07682-3 (Accessed: 7 September 2025).
Ribeiro, J.R., et al. (2022). ‘Identification of differential drive robot dynamic model parameters’, Sensors, 22(1), article 123. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9865440/ (Accessed: 10 September 2025).
Silvestro, L., et al. (2023). ‘Optimal PID control of a brushed DC motor with an embedded low-cost controller’, Electronics, 12(9), 2023. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9610977/ (Accessed: 7 September 2025).
Zhang, Y., Liu, F., & He, Q. (2023). ‘Review and comparison of path tracking based on model predictive control for mobile robots’, Electronics, 8(10), 1077. Available at: https://www.mdpi.com/2079-9292/8/10/1077 (Accessed: 8 September 2025).
Ziegler, J.G. & Nichols, N.B. (1942). ‘Optimum settings for automatic controllers’, Transactions of the ASME, 64, pp. 759–768. Available at: https://scholar.google.com/scholar?q=Ziegler+Nichols+1942+Optimum+settings+for+automatic+controllers (Accessed: 6 September 2025).