



Trajectory Tracking of a Pioneer P3DX Robot Using Model Predictive Control

Azam Zamhuri Fuadi¹, Ahmad Rizal Miftah Awali Sofyan², Angga Rusdinar³

^{1,2,3}Department of Electrical Engineering, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia

*azamzamhurifuadi@telkomuniversity.ac.id

Abstract

Trajectory tracking of nonholonomic mobile robots using Model Predictive Control (MPC) has attracted significant attention due to its capability to explicitly handle system dynamics and actuator constraints within an optimization framework. However, limited studies specifically investigate the computational consistency of constrained Nonlinear Model Predictive Control (NMPC) under fixed-sampling closed-loop simulation environments. This study presents the implementation of a constrained NMPC framework for trajectory tracking of a Pioneer P3DX differential-drive robot using a discrete-time kinematic model and Sequential Least Squares Quadratic Programming (SLSQP) optimization. The controller is integrated with the CoppeliaSim environment through a ZeroMQ based communication interface operating at a fixed sampling time of 0.1 s. Controller performance is evaluated using circular, lemniscate, and square reference trajectories to analyze predictive behavior under varying curvature conditions. The simulation results demonstrate cumulative Root Mean Square Error (RMSE) values of 0.3868 m for the circular trajectory, 0.0942 m for the lemniscate trajectory, and 0.4911 m for the square trajectory. In the square trajectory case, the controller autonomously reduces linear velocity before sharp corners and generates smooth feasible turning maneuvers while satisfying actuator constraints. These behaviors indicate that the implemented NMPC framework is capable of maintaining stable and constraint-aware trajectory tracking performance within a structured closed-loop simulation environment. The study therefore provides a preliminary validation of computational feasibility prior to hardware level implementation.

Keywords: CoppeliaSim; model predictive control; NMPC; pioneer P3DX; SLSQP; trajectory tracking

How to Cite: A. Z. Fuadi, A. R. M. A. Sofyan, and A. Rusdinar, "Trajectory Tracking of a Pioneer P3DX Robot Using Model Predictive Control", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 10, no. 3, pp. 648 - 656, Jun. 2026.
Permalink/DOI: <https://doi.org/10.29207/resti.v10i3.7383>

Received: December 15, 2025

Accepted: May 23, 2026

Available Online: June 14, 2026

*This is an open-access article under the [CC BY 4.0 License](#)
Published by [Ikatan Ahli Informatika Indonesia](#)*

1. Introduction

The development of autonomous mobile robots requires control strategies capable of ensuring accurate trajectory tracking while respecting nonholonomic motion constraints and actuator limitations [1]. Among various approaches, Model Predictive Control (MPC) has become a prominent solution due to its ability to explicitly incorporate system dynamics and input constraints within an optimization-based framework [2]. Compared to conventional geometric or classical controllers, MPC provides a systematic methodology for managing multi-variable systems under physical limitations [3-6].

In mobile robotics, MPC has been extensively applied to path-following and trajectory-tracking problems, where the objective extends beyond minimizing positional deviation to ensuring convergence along time-parameterized reference trajectories [7-11]. The

predictive nature of MPC allows the controller to anticipate upcoming curvature variations and adjust control inputs accordingly [12]. Recent studies have further explored adaptive and learning-based extensions to enhance robustness under varying operating conditions [13-16]. While linearized MPC schemes are frequently adopted to reduce computational burden, nonlinear MPC (NMPC) offers improved fidelity in representing nonholonomic robot dynamics, particularly for differential-drive platforms [4], [17].

Despite the extensive body of literature on MPC-based path tracking, most studies emphasize algorithmic refinement, adaptive weighting strategies, or high-speed vehicle dynamics [7], [14], [18-20]. Comparatively less attention has been directed toward examining the computational behaviour of constrained nonlinear MPC within structured closed-loop simulation environments. For differential-drive robots, solving a nonlinear constrained optimization problem at

every sampling instant introduces computational demands that must remain consistent with the selected sampling interval. Evaluating whether such optimization can be executed reliably within a fixed control period is therefore an important preliminary step prior to hardware-level implementation.

The Pioneer 3DX robot provides a representative nonholonomic differential-drive platform commonly used for control validation. Its planar kinematic structure aligns with widely adopted modelling approaches in robotic and autonomous vehicle research [3]. However, integrating NMPC with realistic actuator bounds and maintaining synchronized communication between controller and plant requires careful architectural design to preserve temporal consistency within a closed-loop framework.

Accordingly, this study implements a nonlinear MPC framework for trajectory tracking of a Pioneer 3DX robot within a structured CoppeliaSim-based simulation environment. The controller employs a discrete-time kinematic model and constrained optimization using SLSQP, integrated through a ZeroMQ communication interface to enable closed-loop operation with a fixed sampling time [21]. The objective of this work is to assess the computational consistency and tracking robustness of NMPC under simulation conditions, rather than to claim hardware-level real-time deployment or superiority over alternative controllers. Performance is evaluated across trajectories with varying curvature characteristics to analyse predictive behaviour and constraint handling within the simulated environment [22].

This study contributes by demonstrating the computational consistency of constrained NMPC within a fixed-sampling closed-loop simulation framework using a Pioneer 3DX differential-drive robot. Furthermore, the proposed framework evaluates predictive and constraint-aware trajectory tracking behaviour across multiple trajectory geometries through synchronized integration between CoppeliaSim and a ZeroMQ based communication architecture.

2. Methods

2.1 Pioneer robot

The kinematic behaviour of the Pioneer 3DX is modelled as a differential drive system as illustrated in Figure 1. Figure 2 illustrates the robot's pose (x, y, θ) is governed by the independent velocities of its left and right wheels. While dynamic models offer precision at high speeds, this study employs a kinematic model due to its computational efficiency and sufficient accuracy for low-speed manoeuvring in urban-like environments. The relationship between the robot's global pose and its control inputs (linear velocity v and angular velocity ω) is described by standard nonholonomic constraints, as indicated by the equations $(\dot{x} = V \cos(\theta))$, $(\dot{y} = V \sin(\theta))$, and $(\dot{\theta} = \omega)$.

Equation 1 describes the continuous-time kinematic model of the Pioneer 3DX differential-drive robot. In this formulation, x and y denote the robot position in the global coordinate frame, θ represents the robot orientation, v corresponds to the linear velocity, and ω denotes the angular velocity. The model captures the nonholonomic motion behavior of the robot under planar movement assumptions.

$$\begin{cases} \dot{x} = v \cos(\theta) \\ \dot{y} = v \sin(\theta) \\ \dot{\theta} = \omega \end{cases} \quad (1)$$

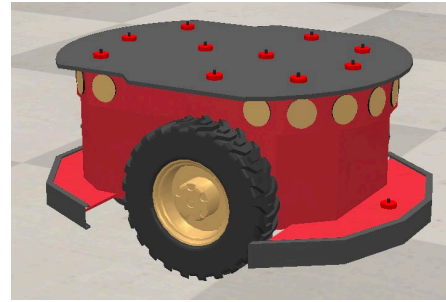


Figure 1. Pioneer 3DX

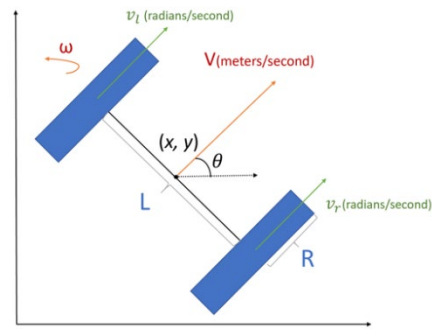


Figure 2. Kinematic Models

Equation 2 presents the discrete-time kinematic model obtained using the Euler discretization method. Planar kinematic models are widely used in the literature due to their simplicity and sufficient performance for low-speed maneuvering in urban environments [3]. While high-speed applications often require dynamic models, kinematic models remain effective for trajectory tracking where tire slip is negligible. As shown in Equation 2, (x_{k+1}) and (y_{k+1}) represent the predicted position coordinates after a sampling interval (Δt) , while (θ_{k+1}) denotes the predicted robot orientation. These future states are calculated using the current robot (x_k, y_k, θ_k) together with the control inputs (V_k) and (ω_k) , which correspond to the linear and angular velocities, respectively. This discretization is crucial for implementing optimization-based controllers operating in a fixed sampling framework, as it enables the robot motion to be simulated and predicted in a step-by-step manner. Furthermore, recent studies suggest that incorporating error compensation mechanisms into the kinematic model can significantly reduce tracking deviations on curved paths [23]. While dynamic models generally provide higher fidelity,

kinematic models remain attractive for low-speed trajectory-tracking applications due to their lower computational complexity and adequate accuracy in start-stop maneuvers [24]. Nevertheless, their performance may deteriorate at high speeds where tire slip effects become significant, requiring appropriate velocity constraints [25].

$$\begin{cases} x_{k+1} = x_k + v_k \cos(\theta_k) \Delta t \\ y_{k+1} = y_k + v_k \sin(\theta_k) \Delta t \\ \theta_{k+1} = \theta_k + \omega_k \Delta t \end{cases} \quad (2)$$

After establishing the kinematic representation of the Pioneer P3DX robot, an integrated control architecture is required to facilitate communication between the optimization-based controller and the simulation environment. Therefore, the overall system architecture employed in this study is presented in the following subsection.

2.2 System Architecture

The architecture of the proposed trajectory tracking system is designed as a closed-loop framework, as illustrated in Figure 3. The system integration is categorized into four fundamental layers, with the specific operational configurations summarized in Table 1. The first layer is the Controller Layer, which serves as the core intelligence by implementing a Nonlinear Model Predictive Control (NMPC) algorithm. This controller utilizes the Sequential Least Squares Quadratic Programming (SLSQP) optimizer from the SciPy library to process the robot's current state (x, y, θ) and predict optimal control inputs (v, ω) . This process is achieved by minimizing a quadratic cost function J over a defined prediction horizon ($N_p = 10$). To ensure low-latency and synchronous data exchange between the high-level Python controller and the simulation environment, a Communication Layer employing the ZeroMQ Remote API is established. This middleware operates via TCP port 23000 using a JSON-encoded Request-Reply (REQ-REP) pattern, enabling a precise sampling time of $dt = 0.1s$.

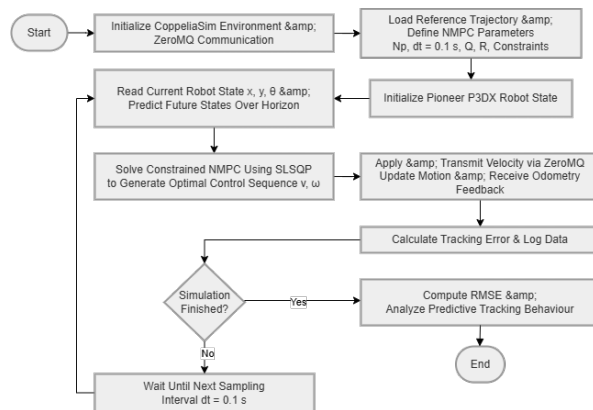


Figure 3. NMPC trajectory tracking framework.

The execution of the control commands occurs within the Plant Layer, hosted in the CoppeliaSim simulator.

This environment features a physics-based model of the Pioneer P3DX differential drive robot, which receives velocity commands and provides continuous feedback of the robot's pose through internal odometry sensors and the simulation clock. Finally, the system incorporates a Visualization and Logging Layer using the Matplotlib FuncAnimation API for real-time monitoring of tracking performance. Beyond real-time visualization, this layer is responsible for comprehensive data logging, which facilitates post-simulation quantitative analysis. This analysis focus on calculating the Root Mean Square Error (RMSE) to objectively evaluate the tracking precision and stability of the NMPC scheme across various reference trajectories.

Table 1. System Parameters

Parameter	Symbol	Value	Unit
Prediction Horizon	N_p	10	steps
Sampling Time	dt	0.1	seconds
State Error Weight	Q	1.0	-
Control Effort Weight	R	0.1	-
Max. Linear Velocity	$v_{\{max\}}$	2.0	m/s
Max. Angular Velocity	$\omega_{\{max\}}$	1.0	rad/s
Wheelbase (Pioneer P3DX)	L	0.33	meters
Communication Port	TCP	23000	-

The system architecture described previously provides the computational framework for controller implementation. Based on this architecture, the nonlinear model predictive control strategy is formulated to generate optimal control actions while satisfying actuator constraints.

2.3 Model Predictive Control

The core mechanism of the proposed controller involves solving a finite-horizon optimization problem at each time step. By utilizing a discrete-time prediction model, the controller anticipates future states and computes an optimal control sequence that minimizes a specific cost function. This receding horizon approach allows for continuous feedback correction, improving tracking stability against model inaccuracies. At each sampling time, MPC solves an optimization problem that considers the predicted states and control actions over a future time window. Specifically, Nonlinear MPC (NMPC) directly discretizes the nonlinear model of the mobile device, resulting in higher path tracking accuracy compared to linearized approaches. The optimal input for the current step is then applied to the system.

MPC operates through iterative optimization of the plant model over a limited prediction horizon. At a specific time t , the current state of the plant is sampled, and a control strategy that reduces a cost function is calculated using numerical minimization techniques for the future period $[t, t + T]$. The optimization process evaluates possible trajectories that come from the current state and finds a control input that minimizes the cost until time $t + T$. This optimization often uses the

Euler–Lagrange equations to find the cost-minimizing solution.

Only the first control input from the computed sequence is applied to the plant. After that, the system state is sampled again, and the optimization problem is solved once more using the new state. This process continues repeatedly, causing the prediction horizon to move forward in time. Because of this shifting time window, MPC is also known as receding horizon control. Although MPC does not always produce a globally optimal solution, it has shown very strong performance in many real applications.

Many academic studies have focused on improving MPC by developing faster numerical solvers for the Euler–Lagrange equations, studying global stability when MPC uses only local optimization, and improving the overall efficiency of the algorithms.

Figure 4 illustrates a discrete Model Predictive Control (MPC) scheme, which is a method used in control systems to optimize the performance of a plant (the system being controlled). The horizontal axis represents time, with k marking the current time step and $k + 1, k + 2$, and so on indicating future time steps within the prediction horizon. The vertical axis shows the output values of the system. The red line with dots represents the reference trajectory, which is the desired path that the system's output should follow. The brown line with circles shows the predicted output, which is the system's expected performance based on the current control strategy. The orange line with diamonds represents the measured output, indicating the actual performance of the system as it has been observed.

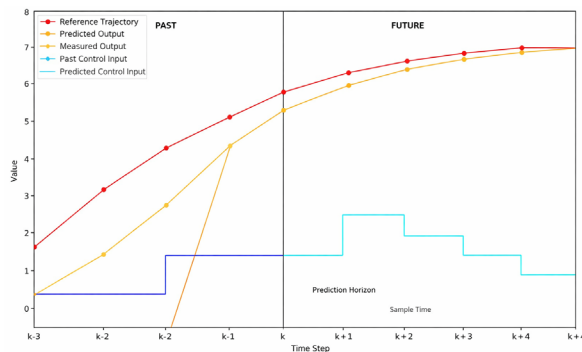


Figure 4. Model Predictive Control Diagram

The blue line with squares depicts the predicted control input, which shows the control actions planned for future time steps to achieve the desired system behaviour. The blue line with circles represents the past control input, indicating the control actions that have been applied up to the current time step. The prediction horizon is the range of future time steps over which the control strategy is optimized. After each sample time, which is the time interval between control updates, the system state is re-evaluated, and the control strategy is recalculated from the new current state, shifting the prediction horizon forward. This process is central to the receding horizon control approach, where the horizon moves as time progresses, allowing for

continuous adaptation to changes in the system or its environment.

The optimization objective of the proposed NMPC controller is formulated through the cost function shown in Equation 3.

$$J = \sum_{k=0}^{N_p} (|(x_k, y_k) - (x_{\text{ref}}, y_{\text{ref}})|_Q^2 + |(v_k, \omega_k)|_R^2) \quad (3)$$

Equation 3 defines the objective function minimized by the NMPC algorithm over a finite prediction horizon. In this formulation, J represents the cumulative optimization cost, while N_p denotes the prediction horizon used to evaluate future system behavior. The first term penalizes the trajectory tracking error between the predicted robot position (x_k, y_k) and the reference position $(x_{\text{ref}}, y_{\text{ref}})$, where the weighting matrix Q determines the importance assigned to tracking accuracy. The second term penalizes the control inputs v_k and ω_k through the weighting matrix R , which regulates actuator utilization and prevents excessively aggressive control actions. The weighting matrices Q and R establish a trade-off between tracking performance and control effort. Larger values of Q encourage the controller to prioritize trajectory convergence, whereas larger values of R result in smoother control commands by penalizing rapid variations in linear and angular velocity. Consequently, the optimization process seeks a feasible control sequence that minimizes tracking error while respecting actuator limitations throughout the prediction horizon. Furthermore, optimizing control inputs contributes to energy efficiency by reducing unnecessary acceleration and braking maneuvers [26].

The MPC framework operates by repeatedly solving a constrained optimization problem over a finite prediction horizon while utilizing the current system state as the initial condition [7], [14]. At each sampling interval, the controller predicts future robot states using the discrete-time nonlinear kinematic model and computes an optimal sequence of control inputs that minimizes a predefined objective function. The optimization process simultaneously considers trajectory tracking accuracy and actuator limitations to ensure feasible control actions throughout the prediction horizon [18].

The objective function in Equation 3 is formulated to minimize both trajectory tracking error and control effort over the prediction horizon. The weighting matrices Q and R regulate the trade-off between trajectory convergence and actuator utilization. Larger values in Q prioritize reference tracking performance, whereas larger values in R penalize excessive control variations to maintain smoother system behavior and reduce aggressive actuator commands [22].

In this study, the constrained optimization problem is solved iteratively using the Sequential Least Squares Quadratic Programming (SLSQP) algorithm available in the SciPy optimization library [21]. The optimizer computes feasible linear and angular velocity inputs

while satisfying predefined actuator constraints at every sampling interval. Only the first control input from the optimized control sequence is applied to the robot, while the optimization process is repeated at the next sampling instant using updated feedback states. This receding horizon mechanism enables the controller to continuously adapt to trajectory changes and maintain closed-loop tracking performance under varying path geometries [5], [14].

The prediction horizon length and sampling interval significantly influence controller behavior and computational performance. A longer prediction horizon generally improves predictive capability and trajectory smoothness but increases optimization complexity and computational load. Conversely, shorter horizons reduce computational burden at the expense of reduced anticipatory behavior during rapid trajectory transitions [4]. Therefore, the NMPC parameters are selected to maintain a balance between tracking accuracy, optimization feasibility, and computational consistency within the fixed sampling interval of 0.1 s used throughout the simulation framework. To implement the NMPC algorithm, the robot dynamics are represented in a state-space form consisting of state variables, control inputs, and a discrete-time prediction model.

Equation 4 defines the system state vector used by the NMPC framework. The state vector consists of the robot position coordinates (x) and (y), together with the orientation angle (θ), which collectively describe the robot pose in the global reference frame. Equation 5 defines the control input vector, which contains the linear velocity (v) and angular velocity (ω) applied to the differential-drive robot.

Based on the state and input definitions, the robot motion is predicted using the discrete-time kinematic model presented in Equation 6. This prediction model estimates the future robot state (x_{k+1}) from the current state (x_k) and control input (u_k) over a sampling interval (Δt). The position coordinates are updated according to the commanded linear velocity and robot orientation, while the heading angle is updated using the angular velocity command. Consequently, Equation 6 provides the predictive mechanism required by the NMPC algorithm to evaluate future system behavior over the prediction horizon.

$$\text{State: } x = [x, y, \theta]^T \quad (4)$$

$$\text{Input: } u = [v, \omega]^T \quad (5)$$

Prediction model:

$$x_{k+1} = f(x_k, u_k) = \begin{bmatrix} x_k + v_k \cos(\theta_k) \Delta t \\ y_k + v_k \sin(\theta_k) \Delta t \\ \theta_k + \omega_k \Delta t \end{bmatrix} \quad (6)$$

The prediction model serves as the core component of the optimization process because all future trajectory estimates are generated from this discrete-time representation. Accurate state prediction enables the

controller to anticipate upcoming trajectory deviations and compute feasible control actions before significant tracking errors occur. Furthermore, the discrete formulation maintains computational efficiency and supports real-time optimization within the fixed sampling interval adopted in this study.

$$(v_{\min} \leq v \leq v_{\max}) \quad (7)$$

$$(\omega_{\min} \leq \omega \leq \omega_{\max}) \quad (8)$$

The control inputs are subject to actuator constraints, as shown in Equations 7 and 8. These bounds limit the allowable linear and angular velocities to physically feasible values that can be executed by the Pioneer 3DX platform. Incorporating actuator constraints directly into the optimization problem ensures that the generated control actions remain realistic and prevents excessive commands that could degrade tracking performance or compromise system stability [12].

To objectively assess the effectiveness of the proposed NMPC framework, quantitative performance indicators are required. Therefore, a trajectory-tracking performance metric is introduced in the following subsection.

2.4 Performance Evaluation Metrics

To provide a rigorous quantitative assessment of the proposed NMPC controller, the Root Mean Square Error (RMSE) is employed as the primary performance metric. RMSE quantifies the cumulative deviation between the robot's actual trajectory and the desired reference path throughout the simulation period. Lower RMSE values indicate higher trajectory tracking accuracy and better overall controller performance.

The RMSE is computed from the Euclidean distance between the actual robot position and the corresponding reference position at each sampling instant. The mathematical formulation is given in Equation 9.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n [(x_i - x_{ref,i})^2 + (y_i - y_{ref,i})^2]} \quad (9)$$

Equation 9 calculates the root mean square value of the position tracking error over the entire simulation horizon. In this formulation, n denotes the total number of sampled data points collected during the simulation, while (x_i, y_i) and $(x_{ref,i}, y_{ref,i})$ represent the actual and reference robot positions at sampling instant i , respectively. The Euclidean distance term measures the instantaneous trajectory tracking error between the robot and the desired path.

By aggregating the error over all sampling instants, RMSE provides a single quantitative indicator of overall tracking performance. This metric is particularly suitable for trajectory-tracking applications because it penalizes large deviations more heavily than small deviations, thereby providing a meaningful assessment of controller accuracy and robustness.

In addition to RMSE, controller performance is qualitatively evaluated through the behaviour of the

linear velocity (v) and angular velocity (ω) profiles. Stable and bounded control signals indicate that the NMPC algorithm successfully maintains trajectory tracking performance while satisfying the predefined actuator constraints throughout the simulation.

To objectively evaluate the proposed controller, three reference trajectories with different geometric characteristics are considered. The circular trajectory is used to assess steady-state tracking performance, the lemniscate trajectory evaluates the controller under continuously varying curvature, and the square trajectory examines the predictive capability of the controller when handling abrupt direction changes. The corresponding simulation results and analyses are presented in the following section.

3. Results and Discussions

To validate the performance of the implemented Model Predictive Controller (MPC), a series of simulations were conducted. The controller's ability to track a circle, lemniscate, and square trajectory was evaluated.

3.1 Case 1: Circular Trajectory

The circular trajectory serves as a benchmark for evaluating the controller's steady-state performance. The "Live Trajectory Tracking" plot (Figure 5, left) shows the robot exhibiting a brief transient phase, visible as an initial spiral, as it converges from its starting position to the reference path. Following this, the robot achieves stable steady-state tracking. The performance plot (Figure 5, right) corroborates this, showing a large initial tracking error and control saturation (velocity at 2.0 m/s) corresponding to the "catch-up" phase. After approximately 78 seconds, the system settles into a steady state where the Tracking Error (red) is minimized to a small, constant value, and both the Linear Velocity (green) and Angular Velocity (blue) settle at constant, non-zero values.

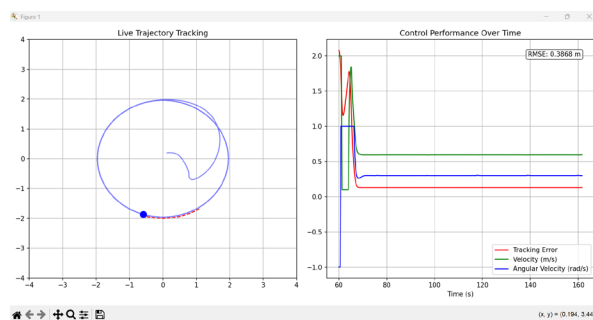


Figure 5. Simulation results for the circular trajectory. (Left) The robot's path (blue) successfully converges to and tracks the circular reference (red dashed). (Right) The corresponding control performance, showing tracking error (red), linear velocity (green), and angular velocity (blue) settling into a steady state.

Quantitatively, the system achieves a cumulative Root Mean Square Error (RMSE) of 0.3868 m over the 160-second simulation period. While the initial transient phase contributes significantly to this value, the real-time tracking error consistently stays near zero once the steady state is reached. This demonstrates the

controller's ability to correctly identify and maintain the optimal constant-velocity and constant-turn-rate commands required to follow a circle with high precision.

3.2 Case 2: Lemniscate Trajectory

The lemniscate trajectory tests the controller's ability to handle dynamic manoeuvres with continuously changing curvature. As shown in the "Live Trajectory Tracking" plot (Figure 6, left), the robot successfully tracks the figure-eight path, demonstrating its capability to manage smooth transitions in turning directions. Accurate prediction models are crucial for maintaining lateral stability during such continuous directional changes, preventing unwanted oscillation [23]. Minor tracking deviations remain visible primarily at the points of highest curvature, where the geometric complexity of the path is most demanding.

The performance plot (Figure 6, right) illustrates a highly dynamic response. The Angular Velocity (blue) oscillates smoothly between positive and negative values, corresponding to the robot navigating the left and right loops of the figure-eight. Concurrently, the Linear Velocity (green) modulates as the controller emergently slows the robot to execute tighter turns and speeds up on straighter sections to maintain stability while minimizing tracking errors [26].

Quantitatively, the system maintains high precision with a cumulative Root Mean Square Error (RMSE) of 0.0942 m over the 160-second simulation. The Tracking Error (red) oscillates accordingly, peaking predictably at high-curvature sections where the controller must balance tracking fidelity against control effort (R) and angular velocity bounds. This relatively low RMSE value, despite the dynamic nature of the path, validates the stability and tracking consistency of the NMPC scheme in handling trajectories with non-constant curvature.

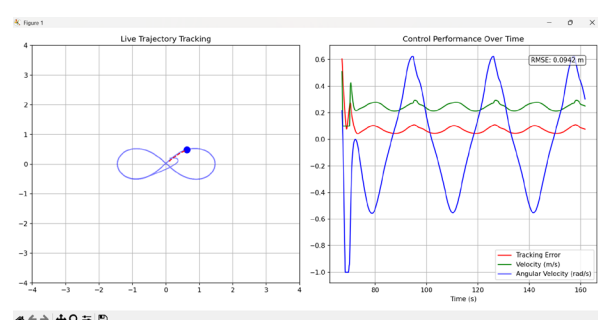


Figure 6. Simulation results for the lemniscate trajectory. (Left) The robot's path (blue) tracks the dynamic figure-eight reference. (Right) The control inputs and tracking error show periodic behaviour as the controller navigates the changing curvature.

3.3 Case 3: Square Trajectory

This trajectory serves as an aggressive test of the controller's predictive nature and its handling of reference discontinuities. The tracking plot in (Figure 7, left) clearly shows the robot *rounding* the 90-degree corners. This behaviour avoids the issue where blind

pursuit of tracking accuracy results in violent steering changes, thereby improving motion smoothness. This is not a failure, but a key demonstration of the MPC's predictive optimization. The controller's predictive horizon ($N_p=10$) allows it to "see" the upcoming sharp corner and compute the optimal, *feasible* rounded manoeuvre that respects its dynamic limits. Strict adherence to kinematic constraints is crucial to prevent overshoot during such aggressive manoeuvres, ensuring feasible motion generation.

This aligns with recent findings where integrating curvature prediction enables the controller to adjust strategies based on real-time conditions, enhancing tracking accuracy [14]. The performance plot (Figure 7, right) reveals a distinct, periodic behaviour. The Angular Velocity (blue) is near zero on the straight edges, followed by a sharp spike to execute the turn. Concurrently, the controller adaptively reduces the linear velocity (the dips in the Linear Velocity (green)) for the turn and accelerate on the straight sections.

Quantitatively, the experiment yielded a cumulative Root Mean Square Error (RMSE) of 0.4911 m over the 160-second duration. The Tracking Error (red) spikes precisely at the corners, which is the moment the reference path is furthest from the robot's optimal rounded path. This emergent behaviour aligns with speed-assisted constraint strategies, where velocity is dynamically limited based on path curvature to ensure safe and accurate cornering [27]. Despite these transient spikes, the rapid recovery of the tracking error to a low steady-state value demonstrates the stable and constraint-aware behaviour of the NMPC-SLSQP scheme in handling non-differentiable trajectory profiles.

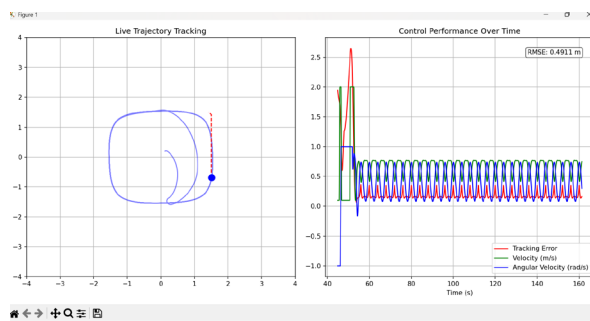


Figure 7. Simulation results for the square trajectory. (Left) The robot's path (blue) shows the characteristic "rounding" of corners, a result of the predictive optimization. (Right) Control inputs and error spike periodically, corresponding to the "straight-turn" manoeuvres.

3.4 Discussion

The simulation results demonstrate that the proposed NMPC framework is capable of maintaining stable trajectory tracking performance under different path geometries while respecting actuator constraints imposed on the Pioneer P3DX platform. Each trajectory provides distinct insights into the predictive characteristics of the controller and the interaction between trajectory curvature, optimization behaviour, and control effort.

For the circular trajectory, the controller exhibits stable convergence behaviour after an initial transient phase. The relatively larger RMSE value of 0.3868 m is primarily influenced by the robot's initial convergence from its starting pose toward the reference trajectory. Once the robot reaches the desired path, the tracking error remains consistently small and the control inputs converge to nearly constant values. This indicates that the NMPC formulation successfully computes steady-state linear and angular velocity combinations required for continuous circular motion while maintaining closed-loop stability.

The lemniscate trajectory represents a more challenging scenario due to its continuously varying curvature and alternating turning directions. Despite the increased geometric complexity, the controller achieves the lowest RMSE value of 0.0942 m. This observation may initially appear counterintuitive because the lemniscate trajectory contains more frequent directional changes than the circular trajectory. However, unlike the square trajectory, the curvature changes occur smoothly and continuously, allowing the NMPC optimizer to anticipate future motion requirements more effectively. As a result, the controller can generate gradual velocity adjustments that reduce tracking deviations while maintaining constraint satisfaction. This result suggests that the prediction horizon and optimization process are sufficiently responsive to handle smooth curvature transitions. The oscillatory angular velocity profile observed during the simulation corresponds directly to the alternating directional changes of the figure-eight trajectory. Simultaneously, the linear velocity adapts dynamically to maintain tracking precision without violating actuator constraints. The relatively low tracking error throughout the simulation indicates that the controller effectively balances tracking accuracy and control effort within the quadratic optimization framework.

The square trajectory provides the most aggressive evaluation of the predictive capability of the NMPC controller because it contains discontinuous curvature transitions at each 90-degree corner. In this scenario, the robot does not attempt to follow the corners exactly; instead, it generates smooth rounded turns. This behaviour is not caused by tracking failure but emerges naturally from the finite prediction horizon and constrained optimization process. Such corner-smoothing behaviour is commonly observed in constrained MPC implementations and reflects the predictive nature of optimization-based control. Rather than forcing the robot to exactly follow a geometrically discontinuous path, the controller generates dynamically feasible manoeuvres that simultaneously satisfy tracking objectives and actuator limitations. Consequently, small deviations near the corners should be interpreted as a trade-off between trajectory accuracy and motion feasibility rather than as a control deficiency. The controller anticipates future trajectory discontinuities and computes feasible control actions that satisfy the predefined velocity constraints. As the

robot approaches each corner, the optimizer autonomously reduces the linear velocity while increasing angular velocity to achieve a stable turning manoeuvre. Consequently, transient spikes in tracking error occur near the corners, resulting in a higher cumulative RMSE value of 0.4911 m. Nevertheless, the rapid recovery of the tracking error after each turn demonstrates the stable and constraint-aware behavior of the NMPC scheme under non-smooth trajectory conditions.

Overall, the observed behaviours confirm that the implemented NMPC framework successfully captures the predictive and constraint-aware characteristics expected from nonlinear optimization-based control strategies. The controller consistently produces feasible control inputs while maintaining stable trajectory convergence across trajectories with different curvature profiles. Furthermore, the integration between the Python-based optimization framework and the CoppeliaSim environment through the ZeroMQ communication interface remains computationally consistent throughout the fixed sampling interval of 0.1 s.

A comparative evaluation of the three trajectory scenarios reveals several important characteristics of the proposed NMPC framework. Although the circular trajectory represents a geometrically simple path, its RMSE value is influenced by the initial convergence phase required for the robot to reach the reference trajectory. In contrast, the lemniscate trajectory achieves the lowest RMSE despite its continuously varying curvature. This result suggests that the prediction horizon and optimization strategy are particularly effective when tracking smooth and continuously differentiable paths.

The square trajectory generates the highest RMSE because of the abrupt curvature discontinuities at each corner. Nevertheless, the controller maintains stable closed-loop behaviour by producing smooth and feasible turning manoeuvres rather than attempting to follow the discontinuous path exactly. This behaviour reflects a fundamental characteristic of constrained NMPC, where trajectory feasibility and actuator limitations are incorporated directly into the optimization process. Consequently, the controller prioritizes stable and physically realizable motion generation over exact geometric path matching when sharp trajectory transitions are encountered.

Overall, the results indicate that trajectory smoothness has a significant influence on tracking performance. Paths with continuous curvature transitions are more compatible with optimization-based predictive control, whereas discontinuous trajectories naturally introduce larger transient tracking errors due to the physical limitations of the robot and the finite prediction horizon.

It is important to note that the present study focuses specifically on simulation-level validation and does not include comparisons with alternative controllers such as

PID, Pure Pursuit, Stanley, or linear MPC approaches. In addition, the experiments are conducted under idealized simulation conditions without external disturbances, sensor noise, wheel slip, or actuator uncertainty. Therefore, the reported results should be interpreted as an evaluation of computational feasibility and predictive tracking behaviour within a structured closed-loop simulation environment rather than as a comprehensive benchmark for real-world deployment.

4. Conclusions

This study presented the implementation and evaluation of a constrained Nonlinear Model Predictive Control (NMPC) framework for trajectory tracking of a Pioneer 3-DX differential-drive robot within a structured closed-loop CoppeliaSim environment. The controller employs a discrete-time kinematic model and solves a constrained optimization problem using the Sequential Least Squares Quadratic Programming (SLSQP) method at each sampling interval. Through this formulation, the controller is capable of explicitly considering actuator constraints while generating predictive control actions for trajectory tracking.

The simulation results demonstrate that the proposed NMPC framework achieves stable tracking performance across trajectories with different geometric characteristics. For the circular trajectory, the controller achieves a cumulative RMSE of 0.3868 m after converging to a steady-state tracking condition. In the lemniscate trajectory, the controller obtains the lowest RMSE value of 0.0942 m, indicating effective handling of continuously varying curvature and alternating turning directions. Meanwhile, the square trajectory produces an RMSE of 0.4911 m due to the discontinuous curvature at sharp corners. In this case, the controller exhibits predictive behaviour by autonomously reducing linear velocity prior to turning and generating smooth feasible cornering manoeuvres while satisfying actuator constraints.

The results further indicate that the integration between the NMPC algorithm and the CoppeliaSim environment through the ZeroMQ communication framework remains computationally consistent under a fixed sampling interval of 0.1 s. This confirms the feasibility of executing constrained NMPC in a structured closed-loop simulation environment for differential-drive mobile robot trajectory tracking applications.

Nevertheless, several limitations remain in the present study. All experiments were conducted under idealized simulation conditions without disturbance modelling, sensor noise, wheel slip, communication delay, or actuator uncertainty. In addition, the study does not provide comparative evaluation against other trajectory tracking methods. Therefore, the conclusions of this work are limited to validating computational feasibility and predictive constraint-aware behaviour under simulation conditions.

Future work will focus on extending the proposed framework toward real-time hardware implementation using a physical Pioneer 3-DX platform. Further investigations will include disturbance-response analysis under sensor uncertainty, computational latency evaluation, and comparative studies with alternative trajectory tracking controllers to provide a more comprehensive assessment of the proposed NMPC approach for autonomous mobile robotic systems.

Acknowledgements

The author declares no conflict interest. This research is independent and not received specific grant from any funding agency.

References

- [1] Y. Liang, Y. Li, A. Khajepour, and L. Zheng, "Holistic Adaptive Multi-Model Predictive Control for the Path Following of 4WID Autonomous Vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 69–81, 2021, doi: [10.1109/TVT.2020.3046052](https://doi.org/10.1109/TVT.2020.3046052).
- [2] A. Domina and V. Tihanyi, "Model Predictive Controller Approach for Automated Vehicle's Path Tracking," *Sensors*, vol. 23, no. 15, p. 6862, 2023, doi: [10.3390/s23156862](https://doi.org/10.3390/s23156862).
- [3] M. Rokonzaman, N. Mohajer, S. Nahavandi, and S. Mohamed, "Review and performance evaluation of path tracking controllers of autonomous vehicles," *IET Intell. Transp. Syst.*, vol. 15, no. 5, pp. 646–670, May 2021, doi: <https://doi.org/10.1049/itr2.12051>.
- [4] P. Stano et al., "Model predictive path tracking control for automated road vehicles: A review," *Annu. Rev. Control*, vol. 55, pp. 194–236, 2023, doi: <https://doi.org/10.1016/j.arcontrol.2022.11.001>.
- [5] S. Wang, J. Guo, Y. Mao, H. Wang, and J. Fan, "Research on the Model Predictive Trajectory Tracking Control of Unmanned Ground Tracked Vehicles," *Drones*, vol. 7, no. 8, p. 496, 2023, doi: [10.3390/drones7080496](https://doi.org/10.3390/drones7080496).
- [6] Q. Yao, Y. Tian, Q. Wang, and S. Wang, "Control Strategies on Path Tracking for Autonomous Vehicle: State of the Art and Future Challenges," *IEEE Access*, vol. 8, pp. 161211–161222, 2020, doi: [10.1109/ACCESS.2020.3020075](https://doi.org/10.1109/ACCESS.2020.3020075).
- [7] Q. Ye, C. Gao, Y. Zhang, Z. Sun, R. Wang, and L. Chen, "Intelligent Vehicle Path Tracking Control Method Based on Curvature Optimisation," *Sensors*, vol. 23, no. 10, p. 4719, 2023, doi: [10.3390/s23104719](https://doi.org/10.3390/s23104719).
- [8] F. Micheli, M. Bersani, S. Arrigoni, F. Braghin, and F. Cheli, "NMPC trajectory planner for urban autonomous driving," *Veh. Syst. Dyn.*, vol. 61, no. 5, pp. 1387–1409, May 2023, doi: [10.1080/00423114.2022.2081220](https://doi.org/10.1080/00423114.2022.2081220).
- [9] M. Kim, D. Lee, J. Ahn, M. Kim, and J. Park, "Model Predictive Control Method for Autonomous Vehicles Using Time-Varying and Non-Uniformly Spaced Horizon," *IEEE Access*, vol. 9, pp. 86475–86487, 2021, doi: [10.1109/ACCESS.2021.3088937](https://doi.org/10.1109/ACCESS.2021.3088937).
- [10] S. Yu, M. Hirche, Y. Huang, H. Chen, and F. Allgöwer, "Model predictive control for autonomous ground vehicles: a review," *Auton. Intell. Syst.*, vol. 1, no. 1, p. 4, 2021, doi: [10.1007/s43684-021-00005-z](https://doi.org/10.1007/s43684-021-00005-z).
- [11] W. Tan, M. Wang, and K. Ma, "Research on Intelligent Vehicle Trajectory Tracking Control Based on Improved Adaptive MPC," *Sensors*, vol. 24, no. 7, p. 2316, 2024, doi: [10.3390/s24072316](https://doi.org/10.3390/s24072316).
- [12] M. Aoki, K. Honda, H. Okuda, and T. Suzuki, "Comparative Study of Prediction Models for Model Predictive Path-Tracking Control in Wide Driving Speed Range," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021, pp. 1261–1267, doi: [10.1109/IV48863.2021.9575868](https://doi.org/10.1109/IV48863.2021.9575868).
- [13] B. Liu, Z. Wang, H. Guo, and G. Zhang, "Improved Model Predictive Control Path Tracking Approach Based on Online Updated Algorithm with Fuzzy Control and Variable Prediction Time Domain for Autonomous Vehicles," *World Electric Vehicle Journal*, vol. 15, no. 6, p. 257, 2024, doi: [10.3390/wevj15060257](https://doi.org/10.3390/wevj15060257).
- [14] C. Li, H. Jiang, X. Yang, and Q. Wei, "Path Tracking Control Strategy Based on Adaptive MPC for Intelligent Vehicles," *Applied Sciences*, vol. 15, no. 10, p. 5464, 2025, doi: [10.3390/app15105464](https://doi.org/10.3390/app15105464).
- [15] K. Hu and K. Cheng, "Robust Tube-Based Model Predictive Control for Autonomous Vehicle Path Tracking," *IEEE Access*, vol. 10, pp. 134389–134403, 2022, doi: [10.1109/ACCESS.2022.3231443](https://doi.org/10.1109/ACCESS.2022.3231443).
- [16] L. Hewing and M. N. Z. Kim P. Wabersich, Marcel Menner, "Learning-Based Model Predictive Control: Toward Safe Learning in Control," *Annu. Rev. Control. Robot. Auton. Syst.*, vol. 3, pp. 269–296, 2020, doi: <https://doi.org/10.1146/annurev-control-090419-075625>.
- [17] N. Sun, W. Zhang, and J. Yang, "Integrated Path Tracking Controller of Underground Articulated Vehicle Based on Nonlinear Model Predictive Control," *Applied Sciences*, vol. 13, no. 9, p. 5340, 2023, doi: [10.3390/app13095340](https://doi.org/10.3390/app13095340).
- [18] H. Liu, J. Sun, and K. W. E. Cheng, "A Two-Layer Model Predictive Path-Tracking Control With Curvature Adaptive Method for High-Speed Autonomous Driving," *IEEE Access*, vol. 11, pp. 89228–89239, 2023, doi: [10.1109/ACCESS.2023.3306239](https://doi.org/10.1109/ACCESS.2023.3306239).
- [19] M. Rokonzaman, N. Mohajer, S. Nahavandi, and S. Mohamed, "Model Predictive Control With Learned Vehicle Dynamics for Autonomous Vehicle Path Tracking," *IEEE Access*, vol. 9, pp. 128233–128249, 2021, doi: [10.1109/ACCESS.2021.3112560](https://doi.org/10.1109/ACCESS.2021.3112560).
- [20] K. Geng, N. A. Chulin, and Z. Wang, "Fault-Tolerant Model Predictive Control Algorithm for Path Tracking of Autonomous Vehicle," *Sensors*, vol. 20, no. 15, p. 4245, 2020, doi: [10.3390/s20154245](https://doi.org/10.3390/s20154245).
- [21] M. Wang, J. Chen, H. Yang, X. Wu, and L. Ye, "Path Tracking Method Based on Model Predictive Control and Genetic Algorithm for Autonomous Vehicle," *Math. Probl. Eng.*, vol. 2022, no. 1, p. 4661401, Jan. 2022, doi: <https://doi.org/10.1155/2022/4661401>.
- [22] G. Park and S. B. Choi, "A Model Predictive Control for Path Tracking of Electronic-Four-Wheel Drive Vehicles," *IEEE Trans. Veh. Technol.*, vol. 70, no. 11, pp. 11352–11364, 2021, doi: [10.1109/TVT.2021.3114729](https://doi.org/10.1109/TVT.2021.3114729).
- [23] S. Peicheng, L. Li, X. Ni, and A. Yang, "Intelligent Vehicle Path Tracking Control Based on Improved MPC and Hybrid PID," *IEEE Access*, vol. 10, pp. 94133–94144, 2022, doi: [10.1109/ACCESS.2022.3203451](https://doi.org/10.1109/ACCESS.2022.3203451).
- [24] J. Nan, Z. Ge, X. Ye, A. F. Burke, and J. Zhao, "Model predictive control for autonomous vehicle path tracking through optimized kinematics," *Results Eng.*, vol. 24, p. 103123, 2024, doi: <https://doi.org/10.1016/j.rineng.2024.103123>.
- [25] L. Tang, F. Yan, B. Zou, K. Wang, and C. Lv, "An Improved Kinematic Model Predictive Control for High-Speed Path Tracking of Autonomous Vehicles," *IEEE Access*, vol. 8, pp. 51400–51413, 2020, doi: [10.1109/ACCESS.2020.2980188](https://doi.org/10.1109/ACCESS.2020.2980188).
- [26] M. Schwenzer, M. Ay, T. Bergs, and D. Abel, "Review on model predictive control: an engineering perspective," *Int. J. Adv. Manuf. Technol.*, vol. 117, no. 5, pp. 1327–1349, 2021, doi: [10.1007/s00170-021-07682-3](https://doi.org/10.1007/s00170-021-07682-3).
- [27] C. Dai, C. Zong, and G. Chen, "Path Tracking Control Based on Model Predictive Control With Adaptive Preview Characteristics and Speed-Assisted Constraint," *IEEE Access*, vol. 8, pp. 184697–184709, 2020, doi: [10.1109/ACCESS.2020.3029635](https://doi.org/10.1109/ACCESS.2020.3029635).