Research Square

Preprints are preliminary reports that have not undergone peer review. They should not be considered conclusive, used to inform clinical practice, or referenced by the media as validated information.

Smoothed-PRM-Guided NMPC for Wheeled Mobile Robot Navigation in an Indoor Environment

Ramdane Hedjar

hedjar@ksu.edu.sa

King Saud University

Mohammed Makhtiche

King Saud University

Mohamed Bencherif

King Saud University

Mansour Alsulaiman

King Saud University

Research Article

Keywords: Probabilistic road map (PRM), wheeled mobile robot, nonlinear model predictive control (NMPC), navigation, obstacle avoidance, convex optimization, smoothed path

Posted Date: September 21st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10183116/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.

Smoothed-PRM-Guided NMPC for Wheeled Mobile Robot Navigation in an Indoor Environment

Abstract: This work addresses the challenge of generating dynamically feasible paths for mobile robots navigating cluttered environments using map-based information. Although probabilistic planners can compute collision-free routes, the resulting paths often contain sharp variations and disregard actuator constraints, making direct NMPC tracking potentially unsafe. To ensure smooth and reliable execution, we integrate a smoothed PRM path with a constrained NMPC formulation. The NMPC solves the underlying nonconvex optimization problem to compute control inputs that accurately follow the smoothed trajectory. Stability is established through Lyapunov analysis, and simulation results confirm the effectiveness of the proposed method

Keywords: Probabilistic road map (PRM), wheeled mobile robot, nonlinear model predictive control (NMPC), navigation, obstacle avoidance, convex optimization, smoothed path.

1 INTRODUCTION

In unstructured or cluttered environments, probabilistic approaches are used to sample the configuration space and build a roadmap of a collision-free path. In this architecture, PRM or Rapidly-exploring Random Tree (RRT) or other variants serve as the global path planner to navigate the connectivity of an unstructured environment (Sebastian T. *et al*, 2006). Despite their efficiency, these methods can produce paths with abrupt direction changes that conflict with non-holonomic motion constraints.

To address this issue, many improved techniques have been introduced to smooth the generated paths, yielding continuous and collision-free trajectories (Ambrożkiewicz, M. 2025; Heiden, E., 2018; Li, Q et al 2022; Liu, H., 2025). However, these probabilistic planners overlook the physical constraints imposed by the robot's actuators. Nonlinear model predictive control (James B. Rawling *et al.,* 2017; Lars Grüne *et al*, 2016) emerges as an adequate solution, since it can incorporate input constraints and accurately track the smoothed free-collision path. Consequently, the solution is to combine the smoothed path with a constrained non-linear model predictive controller that includes the constraints on the input vector. To solve the constrained nonlinear model predictive control, a nonconvex optimization problem has to be solved online. This paper presents a combination of a smoothed probabilistic road map (PRM) and a nonlinear model predictive controller (NMPC). Further, stability analysis of the tracking error is provided using the Lyapunov method.

The key contributions lie in introducing a comprehensive navigation framework that improves adaptability and enhances the robustness of path planning. Although currently limited to static environments, the approach establishes a strong basis for future work aimed at more resilient autonomous robotic systems and advanced navigation methods.

The structure of the paper is as follows. Section 2 presents the mathematical model of the differential-drive wheeled mobile robot. Section 3 details the nonlinear model predictive control strategy, followed by a

closed-loop stability analysis. Section 4 discusses the PRM algorithm along with the B-spline smoothing applied to the generated path. Section 5 provides simulation results on an indoor map to validate the effectiveness of the proposed control framework. Finally, Section 6 concludes the paper

2 MATHEMATICAL MODEL OF A NONHOLONOMIC WHEELED MOBILE ROBOT

The robot is a simple unicycle mobile robot type shown in Figure 1. The kinematic model for the WMR is given as follows (G. Klancar et al, 2016), (Nardênio A. M.,2022; Bruno S. et al, 2025). :

$$\begin{aligned} \dot{x}_c &= v\cos(\theta), \\ \dot{y}_c &= v\sin(\theta), \\ \dot{\theta} &= \omega. \end{aligned} \tag{1}$$

Here, (x_c, y_c) denotes the coordinates of the robot's reference point in the Cartesian plane. At the same time, θ represents the robot's orientation with respect to the positive x-axis. The variables v and ω correspond to the linear and angular velocities, respectively.

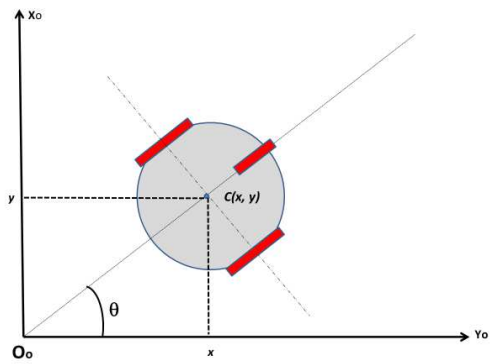


Fig. 1. Wheeled mobile robot

The kinematic model presents the nonholonomic constraint, which is given by:

$$\dot{x}_c \sin\theta - \dot{y}_c \cos\theta = 0,$$

This constraint enforces that the robot's trajectory remains tangent to its principal longitudinal axis. Under the matrix form, the nonlinear kinematic model of WMR is

$$\dot{Z}(t) = f(Z, U, t) = G(\theta)\, u(t) \tag{2}$$

where $G(\theta) = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ 0 & 1 \end{bmatrix}$, $Z(t) = [x(t) \quad y(t) \quad \theta(t)]^T$ and $u(t) = [v(t) \quad \omega(t)]^T$.

The configuration of the robot is given by: $Z = (q^T, \theta)^T \in \Re^2 \times \mathbb{S}^1$ where $q^T = (x, y)$, $\mathbb{S}^1$ is the unit circle in the plane. The control variables of the robot are the linear velocity v and the steering angle ω.

3 NONLINEAR MODEL PREDICTIVE CONTROL

The Nonlinear Model Predictive Control (NMPC) scheme is a receding-horizon optimal control approach that relies on a nonlinear prediction model and explicitly incorporates control constraints by solving, at each sampling instant, a constrained finite-horizon nonconvex optimization problem. At every time step k, given the current error-state measurement $e(k)$, the NMPC computes the control input by minimizing a performance cost function over a prediction horizon N_h (Lazar et al., 2006), thereby determining the optimal control vector U(k):

$$\min_{U_k} J\big(e(k), U(k)\big) = V_f(\, e(k+N_h)) + \sum_{i=0}^{N_h-1} l(\, e(k+i), u(k+i)) \tag{3}$$

Such that:

$e(k+i+1) = f\big(z(k+i), u(k+i)\big) - Z_{ref}(k+i); i = 1, \dots, N_h - 1; e(k)$ is the initail eror − state ; $e(k+i) \in \chi$; $u(k+i) \in \aleph$;$x_{k+N/k} \in \chi_f$.

Where $V_f\big(e(k+N_h)\big)$ is the terminal cost, $\chi_f \in \chi$ is the terminal constraint setand $l\big(e(.), u(.)\big) = \|e(.)\|_Q^2 + \|u(.)\|_R^2$ is the positive definite stage cost. $U(k) = U_k = [u(k) \quad u(k+1) \quad \dots\dots \quad u(k+N_h-1)]$ is the predicted control vector.

In this work, the planned path is assumed to be known and verified as safe a priori (map-based navigation), and an NMPC controller is employed to track this trajectory while ensuring that all control inputs remain within their prescribed bounds. Let the reference trajectory be specified by a sequence of points $P(k) = \begin{bmatrix} x_{ref}(k) \\ y_{ref}(k) \end{bmatrix}; k = 0, \dots, N$, i.e. ($N$+1) points. The corresponding orientation at each point is obtained directly from the path geometry as:

$$\theta_{ref}(k) = tan^{-1}\left(\frac{\Delta y_{ref}(k)}{\Delta x_{ref}(k)}\right),$$

Thus, the complete reference state to be tracked by the NMPC is: $Z_{ref}(k) = \begin{bmatrix} x_{ref}(k) \\ y_{ref}(k) \\ \theta_{ref}(k) \end{bmatrix}; k = 0, \dots, N$. The discrete-time of the mathematical model of the wheeled mobile robot (2) is:

$$z(k+1) = f\big(z(k), u(k)\big), \tag{3}$$

Where $f\big(z(k), u(k)\big) = z(k) + T_s\, G(\theta(k))\, u(i)$; T_s is the sampling time, $G\big(\theta(k)\big) = \begin{bmatrix} \cos\big(\theta(k)\big) & 0 \\ \sin\big(\theta(k)\big) & 0 \\ 0 & 1 \end{bmatrix}; u(k) = \begin{bmatrix} v(k) \\ \omega(k) \end{bmatrix}$.

At time $t = k\, T_s$, the tracking error at prediction step i is defined as:

$$e(k+i) = Z(k+i) - Z_{ref}(k+i); for\ i = 1, \dots . N.$$

Therefore, the nonlinear model predictive controller computes the control sequence:

$$U(k) = [u(k) \quad u(k+1) \quad \dots\dots \quad u(k+N_h-1)]$$

which minimizes the finite-horizon cost function (Eduardo F. C. et al, 2026, James B. Rawling, et al, 2022)):

$$J(U(k), k) = \frac{1}{2} \min_U \Big(\|e(k+N_h)\|_P^2 + \sum_{i=0}^{N_h-1} (e^T(k+i) Q e(k+i) + u^T(k+i) R\, u(k+i))\Big) \tag{4}$$

Where P and Q are (3×3) positive-definite matrices, R is a (2×2) positive semi-definite matrix, and Nh is the prediction horizon. The optimization problem

Stability Analysis:

The closed-loop stability of the NMPC scheme can be established under full-state availability by interpreting the optimal value function of the control problem as a Lyapunov function, provided that certain standard assumptions are satisfied.

Theorem (Goodwin et al, 2000): Consider the system (3) regulated by the NMPC law (4) and subject to the terminal constraint $e(k+N_h)=0$. Under these conditions, the resulting closed-loop system is asymptotically stable.

Proof:

The cost function $J(e(k),U(k))$is positive definite and proper, and can therefore be employed as a Lyapunov function. Let

$$U^*(k)=[u^*(k) \quad u^*(k+1) \quad \dots\dots \quad u^*(k+N_h-1)]$$

denote the optimal control sequence obtained by minimizing the value function in (4). The corresponding optimal value of the cost is denoted by

$$J(e(k),U^*(k))=V^*_{opt}(e(k)),$$

According to the receding-horizon principle, only the first element of the control vector is applied to the process:

$$u^*(k)=h\big(e(k)\big)= \quad \Rightarrow \quad e^*(k+1)=f\big(z(k),u^*(k)\big)-Z_{ref}(k+1)\ ,$$

which represents the nominal closed loop at time *k*.

Consider the resultant cost function, the terminal constraint $e(k+N_h)=0$:

$$J(e(k),U^*_k)=\frac{1}{2}\sum_{i=0}^{N_h-1}\big(\|e(k+i)\|^2_Q+\|u^*(k+i)\|^2_R\big)=V^*_{opt}(e(k)). \tag{5}$$

The optimal cost can be written as:

$$V^*_{opt}\big(e(k)\big)=\frac{1}{2}\|e(k)\|^2_Q+\|u^*(k)\|^2_R+\frac{1}{2}\sum_{i=1}^{N_h-1}\big(\|e(k+i)\|^2_Q+\|u^*(k+i)\|^2_R\big) \tag{6}$$

At the subsequent time step (k+1), we construct a candidate control sequence (not necessarily optimal) as follows:$\widehat{U}(k+1)=\widehat{U}_{k+1}=[u^*(k+1) \quad u^*(k+2) \quad \dots\dots \quad u^*(k+N_h-1) \quad 0]$ With $h(e(k+N_h))=0$.

The cost function for this candidate sequence is:

$$J_{Nopt}\left(e(k+1),\widehat{U}(k+1)\right)=\frac{1}{2}\sum_{i=0}^{N-1}(\|e(k+i+1)\|^2_Q+\|u^*(k+i+1)\|^2_R)$$
$$=\frac{1}{2}\sum_{i=1}^{N-1}\|e(k+i)\|^2_Q+\|u^*(k+i)\|^2_R+\frac{1}{2}\big(\|e(k+N_h)\|^2_Q+\|u^*(k+N_h)\|^2_R\big)=\frac{1}{2}\sum_{i=1}^{N-1}\|e(k+i)\|^2_Q+$$
$$\|u^*(k+i)\|^2_R=V^*_{opt}\big(e(k)\big)-\frac{1}{2}\big(\|e(k)\|^2_Q+\|u^*(k)\|^2_R\big) \tag{7}$$

Let the optimal sequence $U^*(k+1)$, which minimizes the cost at step *k+1*, yield the corresponding optimal value

$$V^*_{opt}\big(e(k+1)\big)=J_{opt}(e(k+1),U^*(k+1))$$

is necessarily no greater than the cost produced by the candidate control sequence. Therefore, the following inequality holds:

$$V^*_{opt}(e(k+1),U^*(k+1))\le V_{Nopt}\left(e(k+1),\widehat{U}(k+1)\right)=V^*_{opt}\big(e(k)\big)-\frac{1}{2}\big(\|e(k)\|^2_Q+\|u^*(k)\|^2_R\big)$$

Or

$$V_{opt}^{*}\big(e(k+1)\big) - V_{opt}^{*}\big(e(k)\big) \leq -\frac{1}{2}\big(\|e(k)\|_{Q}^{2} + \|u^{*}(k)\|_{R}^{2}\big) \leq -\sigma(\|e(k)\|^{2}) \tag{8}$$

Since the stage cost $(e(k), u^{*}(k))$is strictly positive and$\sigma(.)$ is an class-$\mathcal{K}$ function, the obtained inequality satisfies the discrete-time Lyapunov decrease condition. Consequently, the origin of the closed-loop system is guaranteed to be asymptotically stable.

Constraints on the control vector:

Because actuator saturation is unavoidable in practical systems, the robot's linear velocity v(t) and steering velocity ω(t) are restricted to lie within admissible ranges. We therefore enforce the following constraints on the control vector:

$$v_{min} \leq v(t) \leq v_{max} \quad \text{and} \quad \omega_{min} \leq \omega(t) \leq \omega_{max} \ ,$$

These constraints can be expressed as linear constraints as follows: $A\,U(t) \leq B,$

where $\mathrm{A} = \begin{vmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{vmatrix}$, and $B = \begin{vmatrix} v_{max} \\ \omega_{max} \\ -v_{min} \\ -\omega_{min} \end{vmatrix}$. The optimization problem (9) with obstacle avoidance can be formulated as:

$$U^{*}(t) = Arg \min_{U(t)} J(e(k)) \tag{9}$$

such that $AU(t) \leq B$.

4 PROBABILISTIC ROAD MAP

Probabilistic methods are an effective approach for mobile robots that use map-based navigation to find the optimal path to the target in crowded environments.

1. Probabilistic Road Map algorithm

The Probabilistic Roadmap (PRM) is designed to find a safe path between a starting point and a goal in an environment cluttered with obstacles. The steps of the algorithm are given below (Howie C, K. *et al* ,2005; Kavraki L.E. *et al*, 1996):

1.	**Start (Initialization)**
	Input: Robot configuration space with obstacles (Map), start and goal configurations, *N* (number of samples), *k* (Number of nearest neighbors), *d* (distance between neighbors)
2.	**Learning Phase:** a. Sample *N* random configurations in free space (reject if in collision) b. For each sampled node, find *k* nearest neighbors (using distance metric) c. Attempt to connect each neighbor with a local planner (e.g., straight line), checking for collision d. If collision-free, add edge to roadmap graph
3.	**Query Phase:** a. Add start and goal configurations to the roadmap (connect to nearby nodes) b. Perform graph search (for instance: Dijkstra, A*) to find a path from the start to the goal. c. If path exists, return path (set of points from the start to the goal); else report failure
4.	**Output**: Path or failure
5.	**End**

The function that will be used is: Path=PRM (Map, $\boldsymbol{Z_{init}}$ and $\boldsymbol{Z_{Goal}}$, N,k,d)

2. Path Smoothing:

In some circumstances, the generated path is unfeasible due to its hard variations and unshaped trajectory. This causes a discontinuous movement of the robot due to nonholonomic and physical control signal constraints. Many algorithms have been developed to smooth these paths, enabling the mobile robot to navigate safely and smoothly to the goal.

To address the geometric irregularities inherent in sampling-based planners, the initially generated PRM path is refined using a smoothing procedure that adds waypoints and reduces abrupt directional changes or curvature (kappa). The probabilistic road map generates a sequence of (N+1) safe waypoints $P = |P_0 \quad \ldots\ldots \quad P_N|$ where $P_i = \begin{bmatrix} x_{prmi} \\ y_{prmi} \end{bmatrix}; i = 0, \ldots, N, \dim(P) = 2 \times (N+1))$. In this work, the B-spline polynomial is used to connect the initial position to the target of the planned path. It is given by [Ezhov, N *et al.*]:

$$S(t) = \sum_{i=0}^{N} S_i \, B_{i,d}(t) = \sum_{i=0}^{N} \begin{bmatrix} S_{x,i} \\ S_{y,i} \end{bmatrix} B_{i,d}(t),$$

where $B_{i,d}(t)$ is the basis functions (weights) of degree d given by the Cox-de Boor recursion formula for degree d:

- For $d = 0$; we have $B_{i,0}(t) = \begin{cases} 1 & if\ t_i \le t \le t_{i+1} \\ 0 & \text{otherwise} \end{cases}$
- For $d > 0 \quad \Rightarrow \quad B_{i,d}(t) = \frac{t - t_i}{t_{i+d} - t_i} B_{i,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(t)$ where $i = 0, \ldots, n$.

Cubic B-splines (*d=3*) are the most common splines used for the smoothing process because they are C^2 continuous. This means the curve is very smooth. For n waypoints, the knot vector t_i should be augmented to have $N + d + 1$ elements.

The smoothed curve is given by the B-Spline of order:

$$S(t) = \sum_{i=0}^{N} S_i \, B_{i,d}(t) = \begin{bmatrix} S_x(t) \\ S_y(t) \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{N} S_{x,i} \, B_{i,d}(t) \\ \sum_{i=0}^{N} S_{y,i} \, B_{i,d}(t) \end{bmatrix} \text{ for } t \in [0,1].$$

The optimization problem is to find the $2 \times (N+1)(S_{x,i} \text{and } S_{y,i})$ control parameters that minimizes the following cost function:

$$J = \eta_s J_{Smo} + \eta_d J_{Dev} + \eta_o J_{Obse}$$

where $\eta_{[.]}$ denotes the weighting coefficient; J_{Smo} represents the cost term associated with the smoothing process; J_{Dev} penalizes deviations from the reference path generated by the PRM; and J_{Obse} corresponds to the cost term enforcing obstacle avoidance.

To evaluate the cost function of the smoothed waypoints, the spline function is discretized over $(N_s + 1)$ points using the sampling time $h = \frac{1}{N_s}$:

$$t = \{0, h, \ldots\ldots, (N_s - 1)h, 1\}$$

and the discrete smoothed points $S(kh) = \begin{bmatrix} S_x(kh) \\ S_y(kh) \end{bmatrix}, k = 0,1, \ldots\ldots, N_s$. Therefore, the sub-cost functions are:

- **Smoothness cost**: The smoothness is related to the change of the acceleration of the trajectory $S(t)$, and it is given by:

$$J_{Smo} = \int_0^1 \left\| \frac{d^2 S(t)}{dt^2} \right\| dt = \sum_{k=0}^{N_s - 2} \left\| S((k+2)h) - 2\, S\big((k+1)h\big) + S(kh) \right\|$$

This term penalizes the changes in path direction. The second derivative of $S(t)$ is approximated by a finite-difference scheme over $(N_s - 1)$ points.

- **Deviation cost:**

$$J_{Dev} = \sum_{i=0}^{N-1} \|S(t_i) - P_i\|$$

Because the waypoints P_i may not be uniformly spaced, a direct assignment of discrete times t_i leads to a misalignment between $S(t_i)$ and the corresponding waypoint P_i. To ensure that $S(t_i)$ remains close to P_i, the time discretization must be redistributed by explicitly accounting for the distance between successive waypoints P_i, and P_{i-1}.

Define the segment length: $S_{dist}(i) = \|P_{i+1} - P_i\|, i = 0, \dots N-1$, and the cumulative distance:

$$Cum_{dist}(j) = \sum_{i=0}^{j} S_{dist}(i), j = 0, \dots, N-1.$$

The normalized discrete time instants are then computed as: $t_i = \frac{Cum_{dist}(i)}{Cum_{dist}(N-1)}; i = 0, \dots, N-1.$

- **Collision avoidance cost**: To add the collision cost function to be minimized. The map has to be converted to a set of polygon obstacles. The distance of any points of the spline $S(kh)$ to the nearest edge obstacle is calculated, and is defined as:

$$d_{obs}(k) = dist(S(kh), poly(i));$$

where $poly(i)$ is the nearest edge of the ith polygon.

The violation distance can be calculated as:

$$viol(k) = (r + marg) - d_{obs}(k),$$

Here, *r* denotes the robot's radius, while *marg* represents the additional safety distance. Accordingly, the collision-avoidance cost function is reformulated as follows:"

$$J_{Obse} \sum_{k=0}^{N_s} max(0, viol(k))^2.$$

The smoothed path defined by the matrix $S \begin{bmatrix} S_x \\ S_y \end{bmatrix}$ is the solution of the following cost function:

$$\min_{S_x, S_y} J = \eta_s J_{Smo} + \eta_d J_{Dev} + \eta_o J_{Obse}$$

Subject to $S(0) = P_0$, and $S(1) = P_N$.

The smoothed path, to be tracked by the mobile robot, is the sampled curve at (Ns+1) points uniformly spaced parameter values $t \in [0,1]$, and is expressed by the following matrix:

$$S = \begin{bmatrix} S_x(0) & S_y(0) \\ S_x(h) & S_y(h) \\ \dots\dots & \dots\dots \\ S_x(1) & S_y(1) \end{bmatrix} \text{ with } (N_s + 1) \times 2 \text{ elements.}$$

5 SIMULATION RESULTS

To evaluate the effectiveness of the proposed Smoothed-PRM-Guided NMPC framework for wheeled mobile robot navigation in unstructured environments, a series of simulations were conducted under various challenging scenarios. Numerical simulations were conducted in MATLAB (2026a), and the Processor is an Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz. The numerical simulations rely on several MATLAB functions, namely *fmincon* for solving the nonlinear optimization problems arising in both the NMPC formulation and the smoothing stage, *findpath f*or generating the PRM-based reference path, and *optimizeSplineSmooth* for refining and smoothing the resulting PRM trajectory.

The simulations consider two environments populated with irregularly shaped obstacles, in which the robot must navigate safely from the initial position to the goal. The Probabilistic Roadmap (PRM) planner is first employed to generate a collision-free path, which is subsequently refined using a smoothing technique, based on the optimization framework, to ensure curvature continuity and feasibility for wheeled motion. The smoothed path is then used as a reference for the Nonlinear Model Predictive Control (NMPC) scheme, which generates control inputs in real time while accounting for system dynamics. It should be emphasized that the numerical parameters associated with probabilistic road map generation, the smoothing process, and the nonlinear model predictive controller were carefully tuned to their optimal values through extensive simulations.

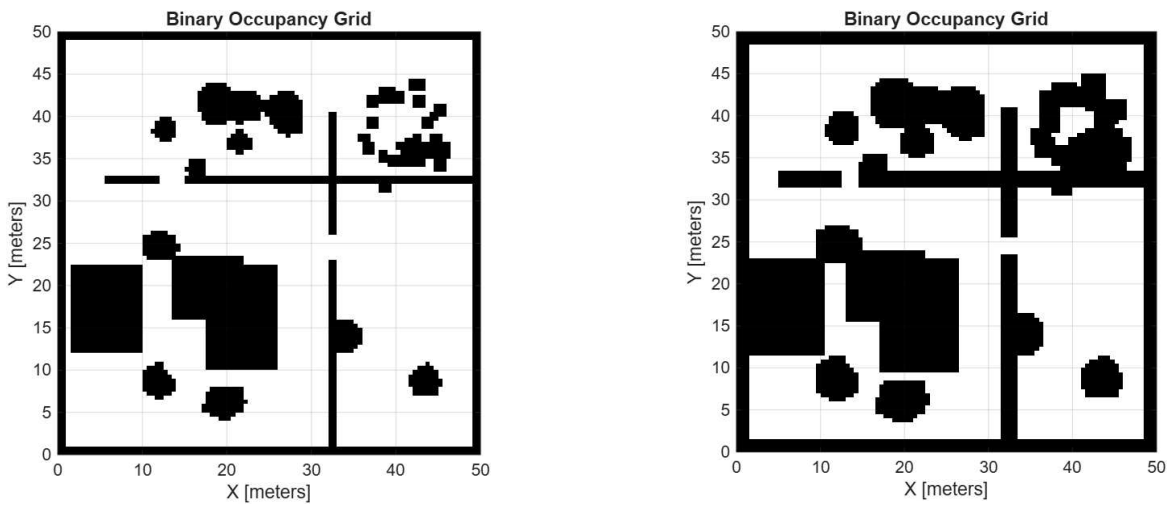
Scenario 1: The first map is generated randomly with irregular polygon-shaped obstacles (12), narrow vertical and horizontal corridors, and a cluttered region with small obstacles at the corners of the map. The size of the map is 50 by 50 m (5 cells per meter).

Parameters for PRM: Robot radius is 0.4m, N_{nodes} = 800 nodes; $D_{link} = 5m$ (Max connection distance between nodes); $dim(P) = N \times 2 = 40 \times 2$; $P(0) = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$ initial location with $\theta(0) = 0\, rad$, $P(40) = \begin{bmatrix} 48 \\ 35 \end{bmatrix}$ end location with $\theta_{ref}(35) = \frac{\pi}{2}$ as the desired final orientation. The robot's radius has dilated the resulting map to incorporate a safety buffer.

Parameters for the smoothing process: $N_s = 400; \eta_s = 4; \eta_d = 20; \eta_o = 1000; d = 3.$

Parameters for NMPC: $Q = \begin{bmatrix} 2000 & 0 & 0 \\ 0 & 2000 & 0 \\ 0 & 0 & 800 \end{bmatrix}; R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}; N_h = 10; T_s = 0.01$ (sampling time); $v_{max} = 35\, m/s\, ; v_{min} = -10\, m/s\, ; \omega_{max} = 10\ rad/s\, ; \omega_{min} = -10 rad/s.$

Figure 2 (a) shows the generated binary occupancy map, and (b) the inflated map with the threshold as the radius of the robot. Figure 3 shows the initial PRM path alongside the resulting smoothed trajectory.



(a) Generated occupancy grid map. (b) Inflated map

Figure 2. Generated map for scenario 1.

Figure 4 illustrates the NMPC tracking performance along the smoothed path over the interval $0 < t < 0.7s$, as well as the complete navigation of the mobile robot from its initial to its desired position. This figure displays only the PRM-generated waypoints and the robot's position during the maneuver. Note that the reference trajectory Z_{ref} (smoothed path) is extended by N_h elements (target states) so that its dimension becomes $dim(Zref) = ((N_s + N_h) \times 3)$.

Figure 5 presents the clearance profile associated with the obstacle-avoidance process during the tracking of the smoothed path. Figure 6 depicts the profile of the constrained control signals. Figure 7 shows the tracking performance of the robot's smoothed path under NMPC. It was noticed that in this work, obstacle avoidance has not been included in the NMPC formulation. This leads to the fact that for tight control signals, the behaviour of the robot can lead to a collision with obstacles. Figure 7 shows the robot tracking behaviour of the smoothed path using NMPC.

Note that when the weighting matrix R is increased to further penalize the control effort, the resulting reduction in control amplitude causes the robot to miss some waypoints along the planned trajectory, preventing it from passing through them

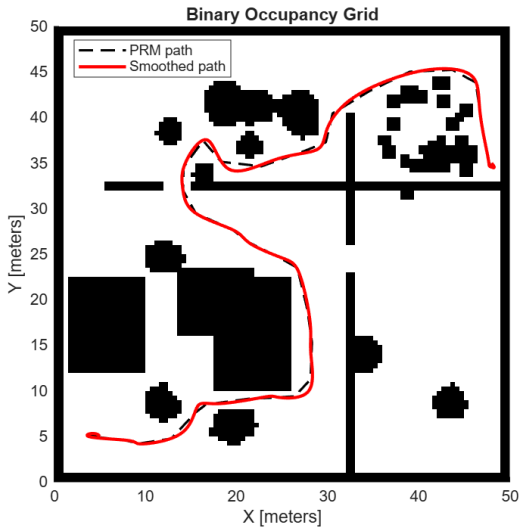
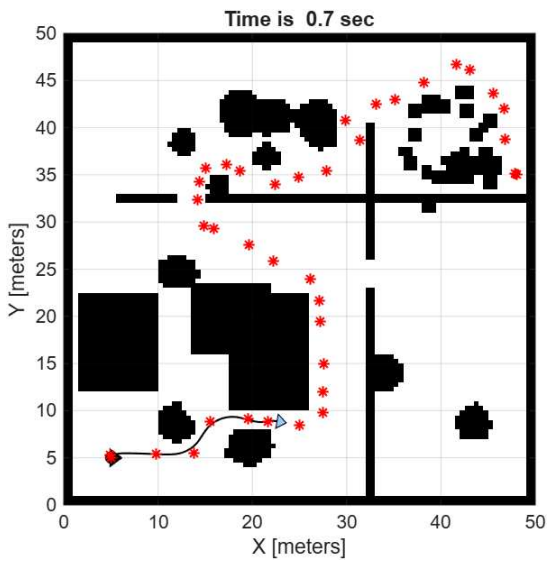
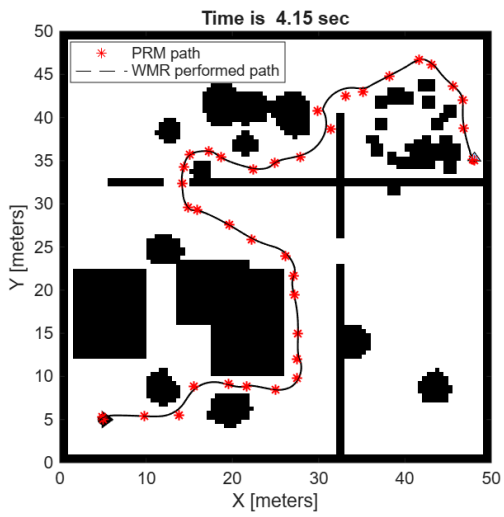


Figure 3: Generated paths (PRM and Smoothed path).



a- Tracking of the smoothed path for $0 < t < 0.7s$.



b- Navigation of the robot from the starting point to the goal.

Figure 4 Tracking performance along the smoothed path.

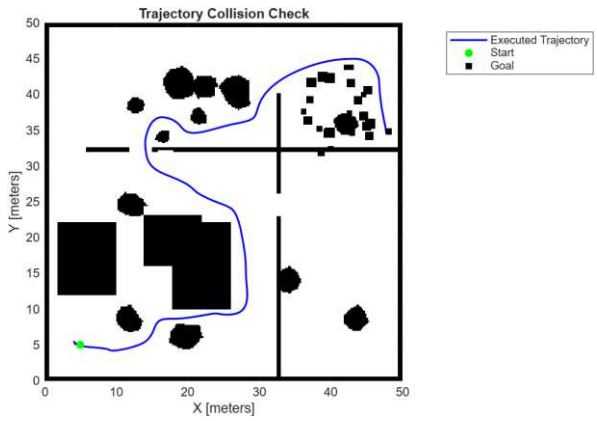


Figure 5 Clearance maintained during obstacle-avoidance manoeuvres.

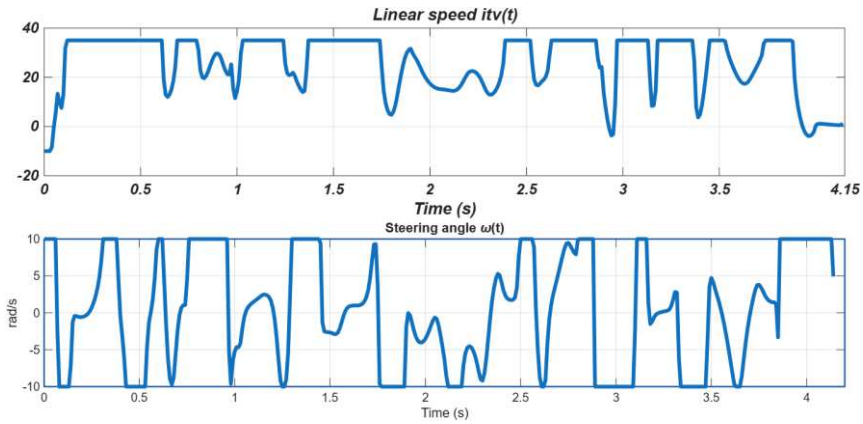


Figure 6. The applied control signals during the motion.

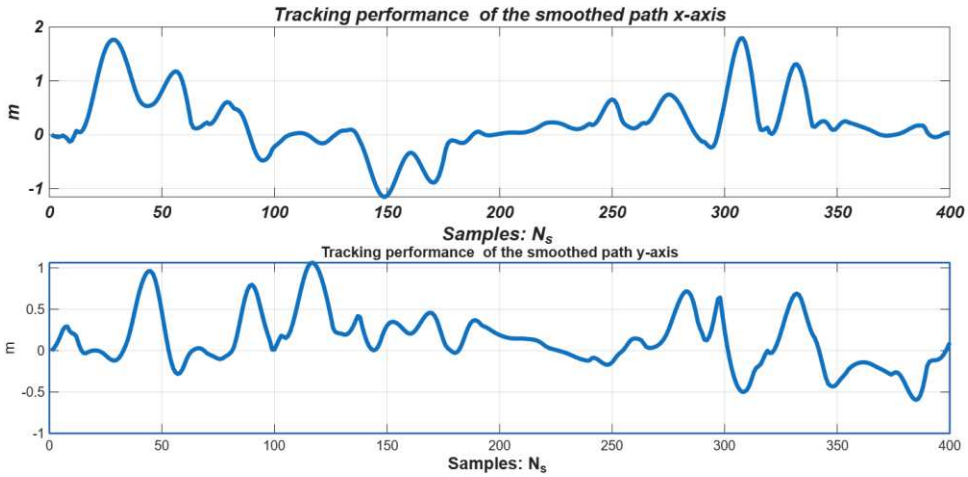


Figure 7. NMPC tracking of the smoothed path.

Figure 8 shows that when the robot attempts to follow the non-smoothed path (PRM path), numerous waypoints are not reached, leading to motion that approaches the obstacle boundary. The corresponding control actions become notably large relative to the earlier unconstrained case. Indeed, the lower waypoint density relative to the smoothed path allowed the robot to move faster, enabling it to reach the target in 0.42 s.

Scenario 2:

In this part, the complex map available on the MATLAB toolbox with the size of the map as: 52×41 isw used. **Parameters for PRM**: Robot radius is 0.4m, N_{nodes} = 1000 nodes; $D_{link} = 5m$ (Max connection distance between nodes) ; $dim(P) = N \times 2 = 24 \times 2$; $P(0) = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$ initial location with initial orientation $\theta(0) = 0\, rad$, $P(23) = \begin{bmatrix} 45 \\ 35 \end{bmatrix}$ end location with end location $\theta(23) = \frac{\pi}{2}$. The robot's radius has dilated the resulting map to incorporate a safety buffer. For both the smoothing procedure and the NMPC controller, the same numerical parameter values of the optimization algorithm were used, except for $N_s = 100$ samples.

Figure 9 shows the original obstacle map and the equivalent map obtained after applying a 1-meter infilling operation. Figure 10 shows the generated paths and NMPC-based tracking of the smoothed trajectory by the mobile robot. Observe that the robot begins and concludes at the designated poses, with the NMPC controller ensuring both accurate tracking of the smoothed path and proper regulation toward the desired final pose.

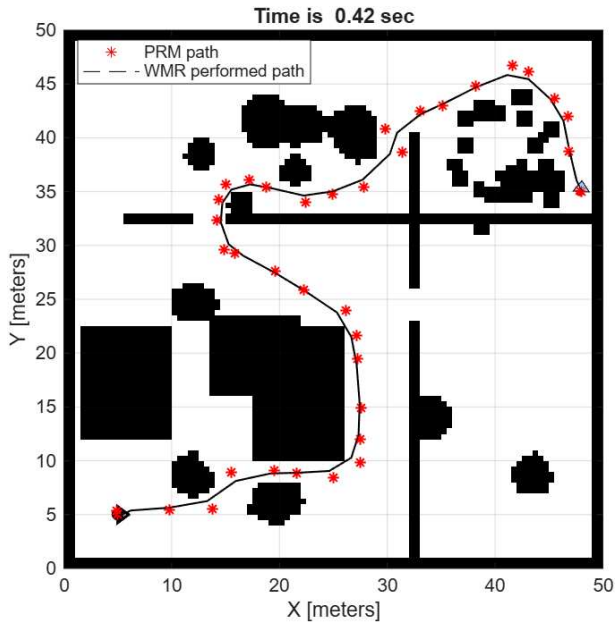
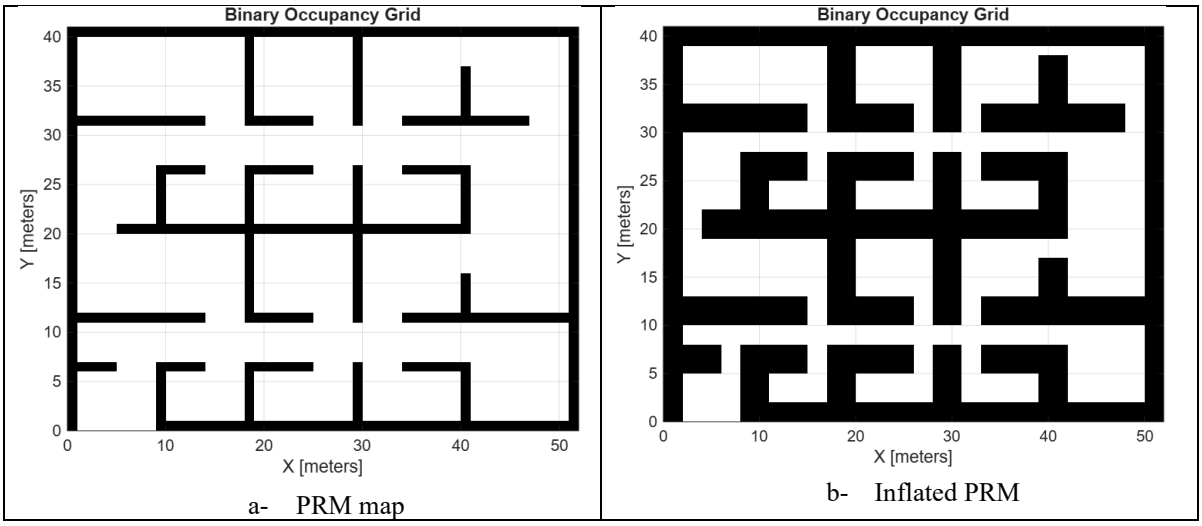


Figure8. Tracking performance of the PRM path by the robot under NMPC.

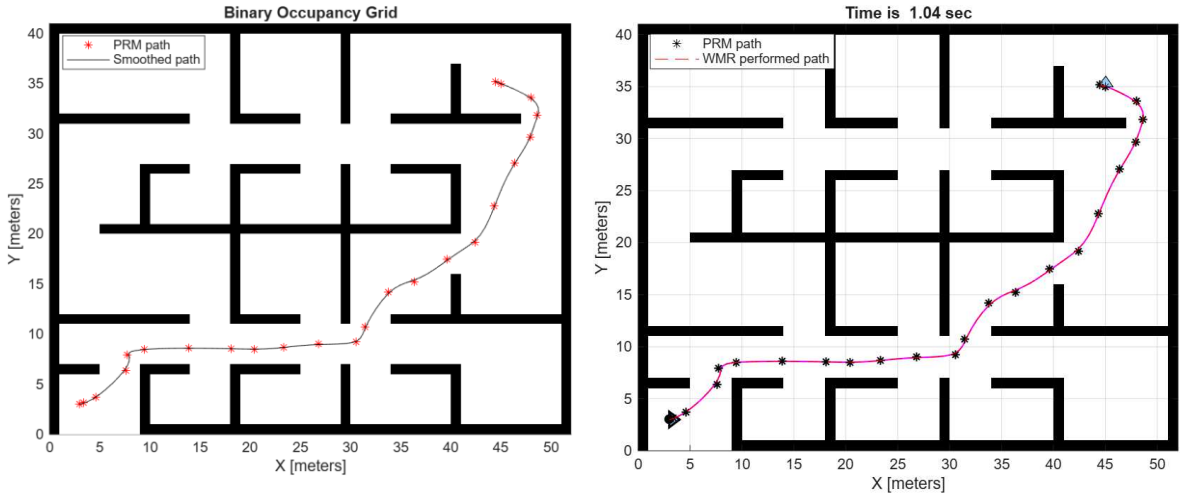


a- PRM map

b- Inflated PRM

Figure 9. Generated map for scenario 2.

Consequently, the results confirm that the proposed approach can generate feasible, smooth, and dynamically consistent trajectories, even in complex and cluttered environments. Moreover, the NMPC successfully tracks the smoothed path with high precision and regulates the robot's pose to the desired final configuration.



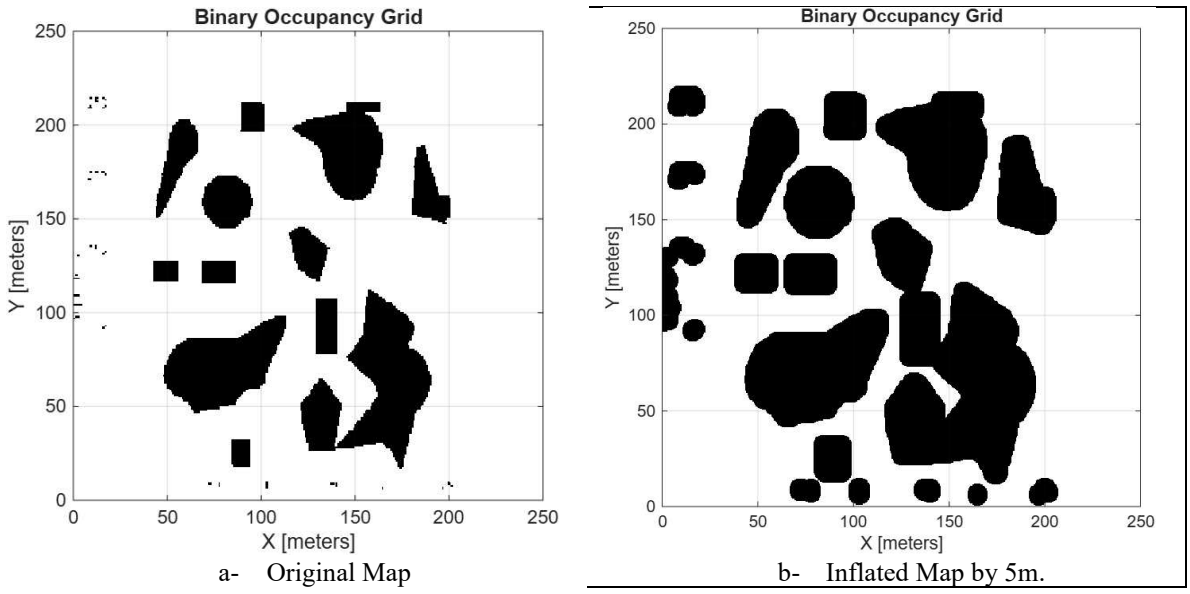
a- Generated PRM and smoothed paths.

b- The executed trajectory of the robot.

Figure 10: Robot trajectory tracking of the smoothed path using the NMPC controller.

Comparison with similar framework: To benchmark the proposed hierarchical navigation framework against recent approaches that integrate a path-planning algorithm with a tracking controller, it is essential to consider representative work from the literature. Indeed, most existing studies emphasize the design and enhancement of path-planning algorithms. For example, S. Chen *et al.* introduced an improved PRM variant and coupled it with a Lyapunov-based tracking controller. Their evaluation compared the improved PRM (IPTM) with classical PRM, RRT, and Bi-RRT planners, while the resulting tracking performance was assessed using PD and PI controllers.

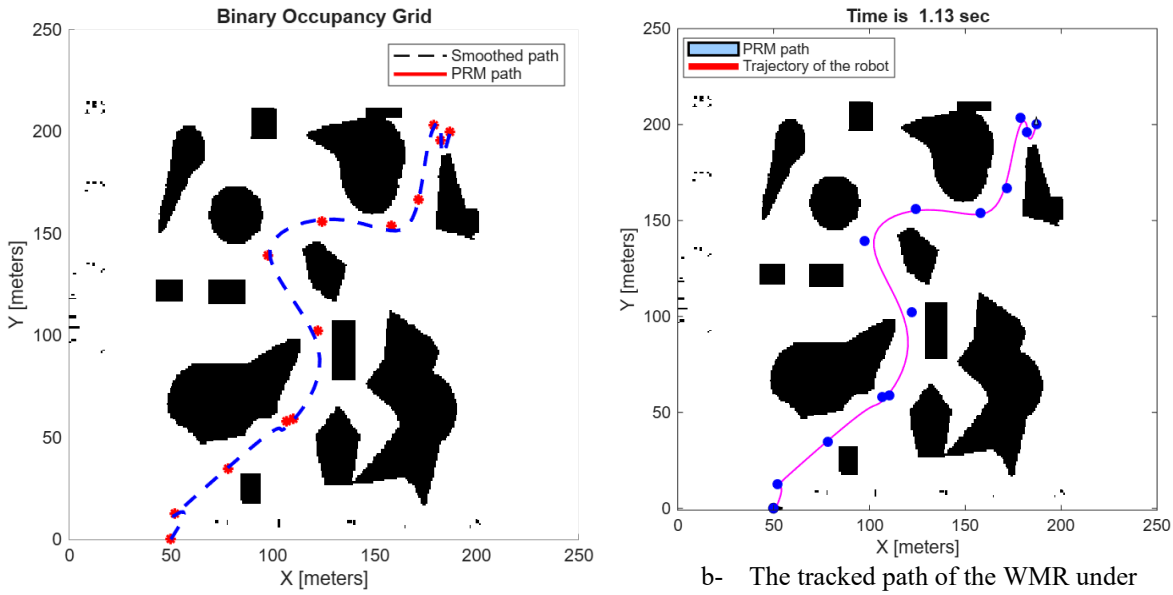
In contrast, M. Ali *et al.* developed a mobile-robot navigation approach that integrates the PRM planner with a pure-pursuit tracking algorithm. To assess the performance of their method, the authors evaluated it across six different maps, each inflated by 5 m and exhibiting progressively increasing levels of complexity. For comparison with our framework, we have produced the most challenging map (map number 6), illustrated in Figure 11. Similar to their study, we adopt the same evaluation metrics—path length, direct distance, and particularly path efficiency, defined as the ratio between the direct distance and the actual path length. In our work, the primary emphasis is placed on the total distance travelled by the wheeled mobile robot. The initial position of the robot is specified as $P(0) = \begin{bmatrix} 50 \\ 0 \end{bmatrix}$ with the initial orientation of the robot: $\theta(0) = 0\, rad$, while the target location is $P(12) = \begin{bmatrix} 187 \\ 200 \end{bmatrix}$, and the desired oriemtation is: $\theta_{ref} = \frac{\pi}{2}$. Thus, the number of waypoints is 13 ($\dim(P) = 13 \times 2$); and the total of Ns=100 sampled points along the smoothed path.



a- Original Map b- Inflated Map by 5m.

Figure 11. The binary occupancy map (M. Ali et al.)

We note that all other parameters used in both the smoothing procedure and the NMPC controller remain unchanged. Figure 12(a) presents the original PRM path alongside its smoothed counterpart, while Figure 12(b) illustrates the robot's executed trajectory under NMPC (shown in red) compared with the PRM path (shown in blue). Figure 13 illustrates the robot's configuration at the end of the trajectory. The results clearly show that under the NMPC controller, the smoothed path is accurately tracked and the robot's configuration converges to the desired final pose.



a- PRM path and the smoothed path.

b- The tracked path of the WMR under NMPC.

Figure 12. (a) Generated paths; (b) Trajectory tracked by the robot closed by the NMPC controller.

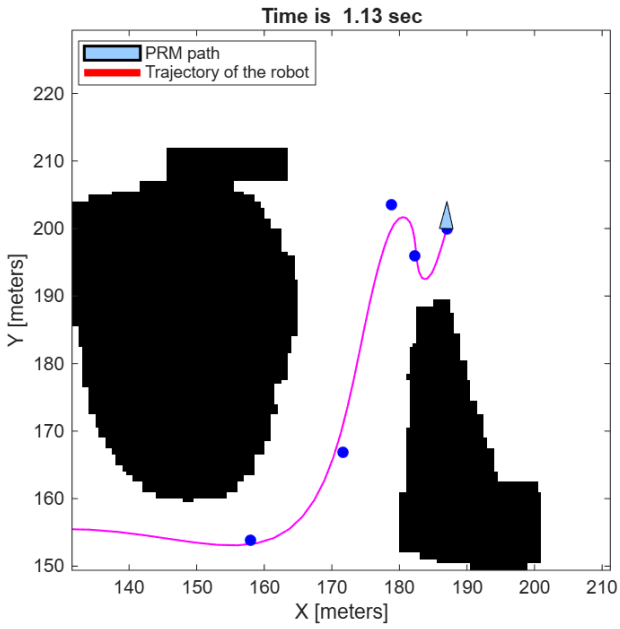


Figure 13. The configuration of the robot at the arrival location.

Table 1 summarizes the results obtained for the same complex map used in the study by M. Ali *et al.* and compares them with the outcomes produced by the proposed NMPC-based navigation framework. The comparison clearly demonstrates that our method achieves superior path efficiency, outperforming the results reported by M. Ali *et al.* on the same benchmark.

TABLE 1. Summary results obtained by M. Ali et al and the proposed smoothed PRM-NMPC.

	Num. Nodes	Path Length (m)	Direct distance (m)	Path efficiency	Num of Way points
M. A;I et al	260	452.54	313.45	0.678	27
		Travelled by WMR			
Proposed SPRM-NMPC	260	339.08	242.423	0.7149	13

6 CONCLUSIONS

In this paper, a framework for map-based navigation of a wheeled mobile robot was presented, integrating a smoothed PRM path with a constrained NMPC controller to generate dynamically feasible trajectories in cluttered environments. Since classical path-planning algorithms often produce non-feasible trajectories with sharp variations that may result in unsafe manoeuvres, the PRM path was first smoothed using a B-spline approach. The NMPC controller then solved the associated nonconvex optimization problem to accurately track the smoothed trajectory.

The proposed framework enhances adaptability and strengthens the robustness of path planning, providing a solid foundation for future developments in autonomous navigation. Stability of the closed-loop tracking system was established through Lyapunov analysis, and numerical simulations demonstrated the effectiveness and reliability of the approach.

Future work will address the limitation observed during the simulations: when the control inputs are tightly constrained within the NMPC formulation, the robot is unable to avoid obstacles because the safe path is generated offline and not enforced during online optimization. To obtain more practical and robust control actions, obstacle-avoidance constraints should be incorporated directly into the NMPC optimization problem.

REFERENCES

Sebastian T., Wolfram B., Dieter F., (2006). *Probabilistic Robotics*, Massachusetts Institute of Technology.

Sreejib Pal1 · Soumitra Keshari Nayak (2024), Probabilistic Mapping and Navigation: A Survey of Bayesian Meta-Learning for Autonomous Robots, *Journal of Intelligent & Robotic Systems*, 111:75. Springer.

Ambrożkiewicz, M. (2025). Advanced Probabilistic Roadmap Path Planning with Adaptive Sampling and Smoothing. *Electronics*, 14(19), 3804.

Heiden, E., Palmieri, L., Koenig, S., Arras, K. O., & Sukhatme, G. S. (2018). Gradient-Informed Path Smoothing for Wheeled Mobile Robots. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1710–1717. https://doi.org/10.1109/icra.2018.8460818

Li, Q., Xu, Y., Bu, S., & Yang, J. (2022). Smart Vehicle Path Planning Based on Modified PRM Algorithm. *Sensors*, 22(17), 6581. https://doi.org/10.3390/s22176581

Liu, H. (2025). Fast Path Planning for Kinematic Smoothing of Robotic Manipulator Motion. *Sensors*, 25(17), 5598.

James B. Rawling, David Q. Mayne, and Moritz M. Diehl (2017). Model Predictive Control: Theory, Computation, and Design, Nob Hill Publishing, LLC.

Lars Grüne, Jürgen Pannek (2017). Nonlinear Model Predictive Control Theory and Algorithms. 2nd Ed., Springer International Publishing Switzerland.

G. Klancar, A. Zdešar, S. Blaži˘c, I. Škrjanc, (2016). *Wheeled mobile robotics from Fundamentals Towards Autonomous Systems*, Ljubljana, Slovenia.

Nardênio A. M., Douglas W. B.(2022). *Wheeled Mobile Robot Control: Theory, Simulation, and Experimentation*, Springer Nature Switzerland AG.

Bruno S., Luigi V., Giuseppe O., Alessandro De Luca (2025), Foundations of Robotics, Springer Cham, DOI: https://doi.org/10.1007/978-3-031-85523-8

Eduardo F. Camacho, Carlos Bordons, and José M. Maestre ,(2026). *Model Predictive Control*. Springer Cham.

James B. Rawling, David Q. Mayne, and Moritz M. Diehl (2022*)*. *Model Predictive Control: Theory, Computation, and Design* (2nd Edition), Nob Hill Publishing, LLC, Cheryl M. Rawlings, publisher.

Howie Choset, Kevin Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia Kavraki, and Sebastian Thrun (2005*), Principles of robot motion: theory, algorithms, and implementation*. The MIT Press. Cambridge, Massachusetts. London, England.

Kavraki, L. E., Svestka, P., Latombe, J.-C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, *12*(4), 566-580.

Graham C. Goodwin, Stefan F. Graebe, and Mario E. Salgado, (2000). Control System Design. Published, 2nd edition, Pearson.

Ezhov, N.; Neitzel, F.; Petrovic, S. Spline Approximation, Part 2: From Polynomials in the Monomial Basis to B-splines—A Derivation. Mathematics 2021, 9, 2198. https://doi.org/10.3390/math9182198

S. Chen, G. Yang, G. Cui, S. Yi and L. Wu, "Improved Path Planning and Controller Design Based on PRM," in IEEE Access, vol. 13, pp. 44156-44168, 2025, doi: 10.1109/ACCESS.2025.3548326.

M. Ali, S. Das and S. Townley, "Robot Differential Drive Navigation Through Probabilistic Roadmap and Pure Pursuit," in *IEEE Access*, vol. 13, pp. 22118-22132, 2025, doi: 10.1109/ACCESS.2025.3536383