2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Towards Local Minima-free Robotic Navigation: Model Predictive Path Integral Control via Repulsive Potential Augmentation

Takahiro Fuke[1], Masafumi Endo[1], Kohei Honda[2] and Genya Ishigami[1]

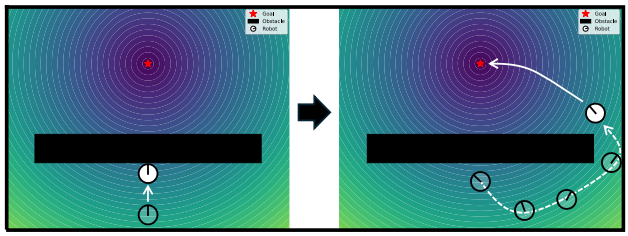
***Abstract*— Model-based control is a crucial component of robotic navigation. However, it often struggles with entrapment in local minima due to its inherent nature as a finite, myopic optimization procedure. Previous studies have addressed this issue but sacrificed either solution quality due to their reactive nature or computational efficiency in generating explicit paths for proactive guidance. To this end, we propose a motion planning method that proactively avoids local minima without any guidance from global paths. The key idea is *repulsive potential augmentation*, integrating high-level directional information into the Model Predictive Path Integral control as a single repulsive term through an artificial potential field. We evaluate our method through theoretical analysis and simulations in environments with obstacles that induce local minima. Results show that our method guarantees the avoidance of local minima and outperforms existing methods in terms of global optimality without decreasing computational efficiency.**

I. Introduction

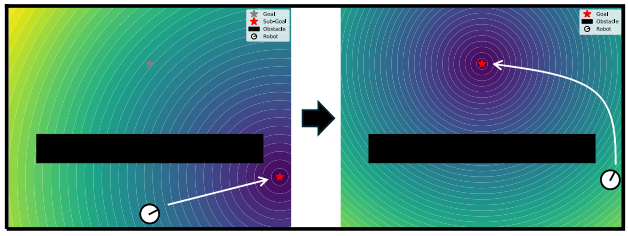
Model Predictive Path Integral (MPPI) control [1], [2], a variant of sampling-based Model Predictive Control (MPC) [3], is a powerful tool for robotic navigation that enables high-frequency optimal control by solving finite-horizon optimization problems. MPPI offers advantages over traditional MPC in handling non-convex and nonlinear problems owning to its gradient-free trajectory rollouts. However, much like traditional MPC, MPPI's inherent focus on generating sequential solutions over finite time horizons often leads robots into *local minima*—suboptimal positions that appear optimal within a local neighborhood. This myopic approach can result in performance far from global optimality and, in the worst case, mission failure due to entrapment in local minima. Achieving global optimality is thereby crucial to solving real-world problems, such as long-horizon navigation in unstructured environments.

Existing studies have explored two main strategies to secure global optimality in model predictive techniques: 1) enhancing the quality of solutions within the *local optimization* framework, and 2) incorporating *global guidance* through integration with path planners. The former strategy includes reactive planning, such as the recovery behavior in Nav2 [4], which can recognize and avoid encountered local minima, but often remains suboptimal due to its ad hoc nature. The latter strategy takes proactive planning over a long horizon by generating reference paths a priori as global guidance. Various path planners, including search- and sampling-based algorithms, have been employed for different use cases [5], [6], [7]. Nevertheless, this strategy requires additional computational costs for reference path search, preventing tight integration of distinct planning steps at high frequency. Global path planners also disregard robot dynamics, posing challenges in translating planned paths into feasible local motions [8], [9]. Recent studies have attempted to address these challenges by using machine learning techniques instead of conventional path planners [10]. Yet, these approaches often involve complex implementations, limiting their broader applicability.

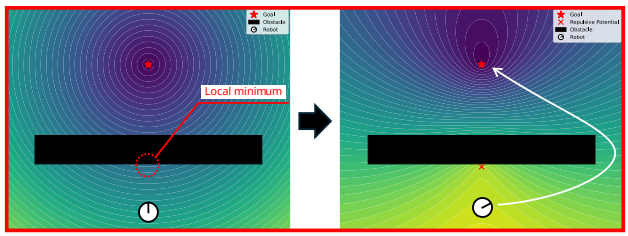
To free from the curse of finite time horizons, we propose a motion planning method that bridges the gap between local optimization and global guidance to avoid local minima (Fig. 1). The key idea is *repulsive potential augmentation*



(a) Reactive methods: Myopic, suboptimal solutions



(b) Proactive methods: Better optimality, requiring path search efforts



(c) Ours: Near-optimal solutions by proactive view via APF with local optimization, without additional computational effort

Fig. 1. Illustrative comparison of motion planning methods, motivated by the local minima problem around a large obstacle. Methods vary in solution optimality and computational efficiency, as detailed in subcaptions (a)-(c).

*This work was not supported by any organization

[1]T. Fuke, M. Endo, and G. Ishigami are with the Space Robotics Group, Department of Mechanical Engineering, Keio University, Yokohama 223-8522, Japan {taka162uke, masafumi.endo}@keio.jp, ishigami@mech.keio.ac.jp

[2]K. Honda is with the Mobility System Group, Department of Mechanical Systems Engineering, Nagoya University, Nagoya 464-8603, Japan honda.kohei.f4@a.mail.nagoya-u.ac.jp

(RPA), which integrates high-level directional information into the MPPI framework. This integration introduces a single *repulsive term* into the optimization objective using an artificial potential field [11]. Such a soft global guidance improves global optimality without explicit path search, resulting in computationally efficient planning. Our approach requires minimal modifications to existing optimization problems, making it practical and extensible to various model predictive techniques. However, we prefer MPPI due to its high compatibility with our approach, especially in terms of solving speed and stability, given its gradient-free nature.

We evaluate our method through theoretical analysis and simulations in environments with different sizes of obstacles, which induce potential local minima entrapment. Our analysis guarantees its ability to avoid entrapment in local minima. Experimental results also confirm that our method outperforms existing strategies in terms of global optimality and computational efficiency.

II. MPPI Review

MPPI is an advanced sampling-based method for solving stochastic optimal control problems. It can handle nonlinear systems, non-convex cost functions, and constraints in complex motion planning tasks. This section outlines MPPI's fundamental principles and key features relevant to our proposal.

A. Basics of MPPI

MPPI considers nonlinear dynamical systems with Gaussian noise added to the control input:

$$\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}_t, \mathbf{u}_t + \delta\mathbf{u}_t), \tag{1}$$

where $\mathbf{x}_t \in \mathbb{R}^n$, $\mathbf{u}_t \in \mathbb{R}^m$, and $\delta\mathbf{u}_t \in \mathbb{R}^m$ represent the state, control input, and Gaussian control noise with zero mean and covariance matrix Σ at time t, respectively. For a finite horizon $t \in \{0, 1, \cdots, T\}$, the stochastic optimal control problem to find the optimal control sequence $\mathbf{U}^* = \{\mathbf{u}_0^*, \mathbf{u}_1^*, \cdots, \mathbf{u}_{T-1}^*\}$ can be formulated as:

$$\mathbf{U}^* = \underset{\mathbf{U}}{\operatorname{argmin}}\, \mathbb{E}\left[\phi(\mathbf{x}_T) + \sum_{t=0}^{T-1}\left(c(\mathbf{x}_t) + \frac{\lambda}{2}\mathbf{u}_t^{\mathrm{T}}\Sigma^{-1}\mathbf{u}_t\right)\right], \tag{2}$$

subject to

$$\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}_t, \mathbf{u}_t + \delta\mathbf{u}_t),\ \delta\mathbf{u}_t \sim \mathcal{N}(0, \Sigma), \tag{3}$$

$$\mathbf{x}_0 = \mathbf{x}_{\mathrm{init}}. \tag{4}$$

Here, $\phi(\cdot)$ and $c(\cdot)$ represent the arbitrary terminal and state-dependent running costs, respectively. The term following the state-dependent running cost is the control cost, expressed as a quadratic form of the control input. $\mathbf{x}_{\mathrm{init}}$ denotes the initial condition, which corresponds to the current state.

MPPI solves the problem in (2) using the importance sampling technique. The method begins with an initial estimated control sequence $\widehat{\mathbf{U}} = \{\widehat{\mathbf{u}}_0, \widehat{\mathbf{u}}_1, \cdots, \widehat{\mathbf{u}}_{T-1}\}$. It then generates K perturbed control sequences $\mathbf{U}_k$ by adding Gaussian noise $\mathcal{N}(0, \Sigma_\epsilon)$ to $\widehat{\mathbf{U}}$, where Σ_ϵ is the covariance matrix for the sampling distribution. These perturbed sequences simulate K system trajectories, or *rollouts*. The costs associated with these rollouts are used to compute a weighted average, which forms the basis for updating the control sequence. The cost of the k-th rollout is denoted as:

$$J(\mathbf{x}_{\mathrm{init}}, \mathbf{U}_k) = \phi(\mathbf{x}_{T,k}) + \sum_{t=0}^{T-1}\left(c(\mathbf{x}_{t,k}) + \lambda\widehat{\mathbf{u}}_t^{\mathrm{T}}\Sigma_\epsilon^{-1}\mathbf{u}_{t,k}\right), \tag{5}$$

and the optimal control input $\mathbf{u}_t^*$ is updated as follows:

$$\mathbf{u}_t^* = \widehat{\mathbf{u}}_t + \frac{\sum_{k=0}^{K-1}\exp\left(-(1/\lambda)(J(\mathbf{x}_{\mathrm{init}}, \mathbf{U}_k) - \rho)\right)\delta\mathbf{u}_{t,k}}{\sum_{k=0}^{K-1}\exp\left(-(1/\lambda)(J(\mathbf{x}_{\mathrm{init}}, \mathbf{U}_k) - \rho)\right)}, \tag{6}$$

where $\lambda \in \mathbb{R}^+$ is the hyperparameter called temperature parameter, and ρ is the minimum cost to prevent numerical over or underflow. The updated solution in (6) minimizes the Kullback-Leibler divergence with the optimal control distribution characterized by the Boltzmann distribution [1]. The rollouts in MPPI are independent one another and therefore, MPPI is applicable for GPU-based parallel computation. MPPI does not need gradient calculations and iterative solution updates. This flexibility enables the use of arbitrary prediction models and cost designs, making it a subject of recent interest [12], [13], [14], [15], [16], [17].

B. State and Control Input Constraints of MPPI

MPPI's sampling-based scheme handles constraints differently from traditional MPC. In motion planning, two primary constraints are control and state constraints. The control constraints are satisfied by clipping perturbed control sequences, ensuring sampled inputs remain within feasible range. The state constraints are addressed by imposing large penalties on rollouts violating these constraints. While conceptually similar to soft constraints in traditional MPC, large penalties in MPPI transform these into hard constraints, expressed with indicator functions.

III. Local Minima-Free Navigation

A. Task Statement

Point-to-goal navigation using MPPI without global reference paths is challenging, as the robot may become trapped in *local minima*—suboptimal positions that appear optimal within a local neighborhood. This problem is due to the cost design and constraints. To describe the local minima issue, we consider a typical scenario in which a robot tries to avoid a rectangular obstacle in a planar environment (Fig. 2). This obstacle represents the simplest convex shape with straight edges, which primarily causes local minima. We define a coordinate system with an obstacle of width W and height H, centered at $\mathbf{p}_{\mathrm{obs}} = [0, H/2]^\top$. The goal is at $\mathbf{p}_{\mathrm{goal}} = [0, y_{\mathrm{goal}}]^\top$, where $y_{\mathrm{goal}} > H$.

While this coordinate system simplifies our analysis, the derived properties remain valid for any chosen coordinate system. Although we analyze this scenario in a planar environment, the configuration exhibits axial symmetry with

respect to the y-axis. This symmetry extends our discussion and results to three-dimensional scenarios, where the obstacle could be a vertical wall or similar structure.

B. Definition of Global Minimum and Local Minima

MPPI is a zeroth-order optimization method that does not require gradient information of the cost function. We define *global minimum* and *local minima* in MPPI analogously to other zeroth-order methods considering the specific context of robotic navigation.

In robotic navigation, the global minimum is an optimal position where no further change for control input is needed, while local minima are suboptimal positions that appear optimal within a local neighborhood. We formalize these concepts by $\mathbf{U}_{\text{const}} \in \mathbb{R}^{m\times T}$, a constant control sequence that forces the robot to stay at the current position, assumed to be position-independent for simplicity. In our planner environment configuration, this concept can be represented by the zero control sequence. With $\mathbf{U}_{\text{const}}$, we define global minimum and local minimum for MPPI as follows:

Definition of global minimum. *A position $\mathbf{p}^*$ is a global minimum if it satisfies the following:*

$$J(\mathbf{p}^*, \mathbf{U}_{\text{const}}) \leq J(\mathbf{p}, \mathbf{U}_{\text{const}}),\ \forall \mathbf{p}.$$

Definition of local minimum. *A position $\hat{\mathbf{p}}^*$ is a local minimum if it satisfies the following:*

1) *Local minimality: $\exists \delta > 0$ such that*

$$J(\hat{\mathbf{p}}^*, \mathbf{U}_{\text{const}}) \leq J(\mathbf{p}, \mathbf{U}_{\text{const}}),\ \forall \mathbf{p} \text{ with } \|\hat{\mathbf{p}}^* - \mathbf{p}\| < \delta.$$

2) *Non-global minimality:* $J(\mathbf{p}^*, \mathbf{U}_{\text{const}}) < J(\hat{\mathbf{p}}^*, \mathbf{U}_{\text{const}})$.

MPPI employs Gaussian random noise to explore the solution space independently of the cost function, aiding in avoiding local minima. However, MPPI requires careful parameter tuning such as planning horizon and Gaussian variance, often increasing computation time or degrading solution quality. These challenges highlight the importance of designing local minima-free cost functions for MPPI-based local minima-free navigation.

C. Conventional Cost Formulation

In MPPI, the cost-to-go function J includes a terminal cost $\phi(\cdot)$ and a state-dependent stage cost $c(\cdot)$ according to the task objectives. For point-to-goal navigation tasks, these costs are generally designed using the squared Euclidean distance to the target position:

$$\begin{aligned}\phi_{\text{baseline}}(\mathbf{p}) &= c_{\text{baseline}}(\mathbf{p}) \\ &= \|\mathbf{p}_{\text{goal}} - \mathbf{p}\|^2 + w_{\text{obst}} \cdot \mathbb{1}_{\text{obst}}(\mathbf{p}),\end{aligned} \tag{7}$$

where $w_{\text{obst}} \in \mathbb{R}^+$ is the positive collision penalty coefficient. It works with the indicator function $\mathbb{1}_{\text{obst}}(\mathbf{p})$ to represent the obstacle constraint. The indicator function is defined using a set of positions A occupied by obstacles:

$$\mathbb{1}_{\text{obst}}(\mathbf{p}) = \begin{cases} 1 & \text{if } \mathbf{p} \in A \\ 0 & \text{if } \mathbf{p} \notin A \end{cases}. \tag{8}$$

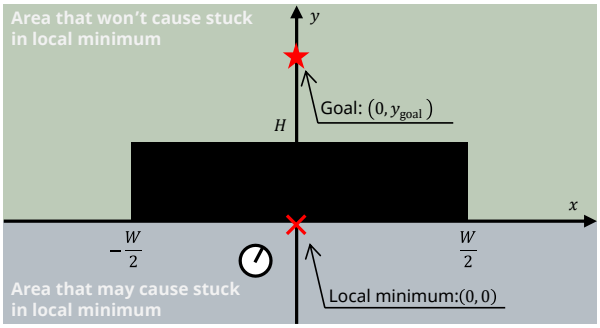


Fig. 2. Coordinate system setting for theoretical analysis. The position indicated as local minimum and the associated regions illustrated are based on the conventional squared Euclidean distance-based cost.

Under the configuration described in Section III-A, the conventional cost formulation in (7) results in a local minimum at $\mathbf{p}_{\text{minimum}} = [0, 0]^\top$. This makes it challenging for the robot to reach the target position when motion planning starts anywhere from $y < 0$, as illustrated in Fig. 2.

D. Repulsive Potential-Augmented Cost Formulation

As discussed in the III-C, local minima are inevitable as long as the conventional cost formulation is applied. We thus propose a new *repulsive potential-augmented cost* that incorporates two key modifications: the use of Euclidean distance instead of squared Euclidean distance, and the addition of a repulsive term to encourage the robot to move away from local minimum. Our proposed formulation is

$$\begin{aligned}\phi_{\text{proposed}}(\mathbf{p}) &= c_{\text{proposed}}(\mathbf{p}) \\ &= \|\mathbf{p}_{\text{goal}} - \mathbf{p}\| - \alpha\|\mathbf{p}_{\text{minimum}} - \mathbf{p}\| \\ &\quad + w_{\text{obst}} \cdot \mathbb{1}_{\text{obst}}(\mathbf{p}),\end{aligned} \tag{9}$$

where $\alpha \in (0, 1)$ balances the repulsion from local minimum and attraction to the target position. The repulsive potential-augmented cost function in (9) is simple to implement, only adding a single term when a local minimum is detected, which offers advantages for system integration.

E. Analysis of Repulsive Potential-Augmented Cost

To demonstrate the cost described in (9) is free from local minima, we show that $\mathbf{p}_{\text{goal}}$ remains the global minimum and no other local minima exist. Based on (9), the cost-to-go $J_{\text{proposed}}(\mathbf{p}, \mathbf{U}_{\text{const}})$ is calculated by (5) as follows:

$$J_{\text{proposed}}(\cdot) = (T+1)(g(\mathbf{p}) + w_{\text{obst}} \cdot \mathbb{1}_{\text{obst}}(\mathbf{p})) + C_{\text{control}}, \tag{10}$$

$$g(\mathbf{p}) = \|\mathbf{p}_{\text{goal}} - \mathbf{p}\| - \alpha\|\mathbf{p}_{\text{minimum}} - \mathbf{p}\|. \tag{11}$$

where C_{control} is the constant term dependent on the constant control sequence, and $(T+1)$ is the horizon-dependent coefficient, both independent of position.

We focus on $g(\mathbf{p})$, representing the cost function in the obstacle-free region to analyze the local minima-free property of $J_{\text{proposed}}(\mathbf{p}, \mathbf{U}_{\text{const}})$. Our analysis shows that $g(\mathbf{p}) = g(x, y)$ satisfies the following four properties in the established coordinate system:

1) $\frac{\partial g(x,0)}{\partial x} < 0 \quad \text{for } 0 < x \leq \frac{W}{2}$
2) $0 < \frac{\partial g(x,0)}{\partial x} \quad \text{for } -\frac{W}{2} \leq x < 0$
3) $\frac{\partial g(x,y)}{\partial y} < 0 \quad \text{for } 0 \leq y \leq H$
4) $g(0, y_{\text{goal}})$ is the unique global minimum, and there are no local minima.

Property 1 to 3 collectively show that $J_{\text{proposed}}(\mathbf{p}, \mathbf{U}_{\text{const}})$ does not exhibit local minima along the perimeter of the obstacle. Property 4 establishes that $\mathbf{p}_{\text{goal}}$ is the unique global minimum and that no local minima exists in the obstacle-free region. Together, these properties indicate that the proposed cost formulation satisfies the criteria for a local minima-free cost. Detailed mathematical proofs for Property 1, 3, and 4 are in the appendix, while the proof for Property 2 follows analogously to that of property 1, with appropriate sign changes due to the reversed x coordinate range.

IV. Simulation Experiments

This section demonstrates the effectiveness of Repulsive Potential-Augmented MPPI (RPA-MPPI), integrating repulsive potential-augmented cost into the MPPI framework. We prepare rectangular obstacle scenarios, as this configuration aligns with our theoretical analysis.

A. Transition Model for 2D Navigation

The robot is modeled as a simple point-mass non-holonomic system, represented as follows:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} v_t \cos\theta_t \\ v_t \sin\theta_t \\ \omega_t \end{bmatrix} \Delta t, \tag{12}$$

where x_t, y_t represent the robot's position [m], and θ_t represents heading angle [rad] at the discrete time t. The input linear velocity v_t is constrained to $[0, 1]$ m/s, and the input angular velocity ω_t is limited to $[-0.5, 0.5]$ rad/s. The time step Δt is set to 0.1 s. This model is used for both the simulation state transition and the MPPI prediction model.

B. Validation Settings

Our validation process included three experimental scenarios, as shown in Fig. 3, characterized by varying obstacle widths: Short, Middle, and Long. For each scenario, goal tolerance was set to 1.0 m and all obstacles had a consistent 0.5 m safety margin. We tested 24 different initial robot states in each setup to ensure comprehensive evaluation. The comparative analysis included: (1) Standard MPPI (Std-MPPI) with a horizon of 50, (2) Std-MPPI with a horizon of 150, (3) Std-MPPI with a horizon of 50, guided by the A* algorithm [18] as a global path planner (A*-MPPI), and (4) RPA-MPPI (ours) with a horizon of 50. The A* algorithm provides a resolution-optimal path when such a path exists. Therefore, A*-MPPI, which incorporates this algorithm, serves as a reliable baseline for comparison. They are implemented in PyTorch for GPU acceleration and run in the simulator [19].

Throughout the algorithm execution, the number of samples K was set to 1000, covariance matrix Σ_ϵ to $\text{diag}\{1.00, 1.00\}$, and temperature parameter λ to 0.10. For

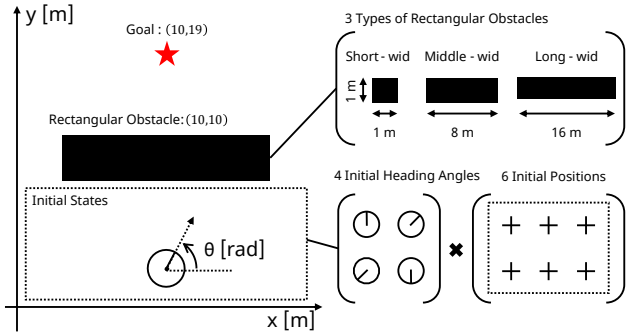


Fig. 3. Three obstacle configurations are tested across 24 initial states. Initial Heading Angles [rad]: $\{\frac{\pi}{4}, \frac{\pi}{2}, \frac{5}{4}\pi, \frac{3}{2}\pi\}$; Initial positions [m]: {(10, 1), (1, 1), (19, 1), (10, 8.5), (1, 8.5), (19, 8.5)}

A*-MPPI, we used octile distance as the A* heuristic, set the grid size to 0.50 m, and used a 2.00 m look-ahead distance for determining subgoals on the reference path. For RPA-MPPI, taking account of the safety margin, $\mathbf{p}_{\text{minimum}} = [0, 9.0]^\top$ was assumed to be given in advance and set the repulsion coefficient to $\alpha = 0.75$, which provided a strong repulsive effect while maintaining sufficient goal attraction. The time limit is set as 50 seconds per navigation trial, and the following four metrics quantify algorithm performance:

- **Success Rate** (SR) [%] evaluates robot safety based on the percentage of successful goal reaches without exceeding the time limit or colliding with obstacles.
- **Relative Difference in Success Time** (RDST) [%] evaluates robot goal-reaching efficiency based on the average relative difference in success time compared to A*-MPPI for successfully completed scenarios.
- **Relative Difference in Path Length** (RDPL) [%] evaluates path optimality based on the average relative difference in total traversed path length compared to A*-MPPI for successfully completed scenarios.
- **Computation Time** (CT) [s] evaluates computational efficiency based on average computational time for optimization per step.

C. Results

Table I summarizes the results across the three obstacle scenarios. RPA-MPPI outperforms Std-MPPI in SR across all scenarios. In the Long-wid scenario, longer horizons help Std-MPPI avoid local minima. However, this doesn't address the underlying issue noted in III-B. A*-MPPI performs best in SR due to its explicit guidance; however, it also requires computational efforts across all scenarios. RPA-MPPI achieves comparable computational efficiency since it embeds proactive guidance steps in its optimization procedure without relying on additional path computations. It's worth noting that A*-MPPI requires a complete map prior to path planning. This assumption is fragile, particularly in unstructured off-road environments, as recent studies have focused on robot navigation using only local motion planning [20]. Our method is applicable even in an incomplete map as it incrementally identifies local minima. However,

TABLE I

PERFORMANCE COMPARISON OF MPPI-BASED NAVIGATION ACROSS VARYING OBSTACLE CONFIGURATIONS

		Short-wid				Middle-wid				Long-wid			
Planner Name	Horizon	SR↑	RDST↓	RDPL↓	CT↓	SR↑	RDST↓	RDPL↓	CT↓	SR↑	RDST↓	RDPL↓	CT↓
A*-MPPI	50	100	-	-	0.248	100	-	-	0.378	100	-	-	0.572
Std-MPPI	50	100	**-10.87**	**+0.30**	**0.044**	67	**+4.27**	**+0.80**	**0.045**	21	**-4.26**	**+1.32**	**0.045**
Std-MPPI	150	100	+25.18	+9.92	0.131	92	+31.05	+8.31	0.131	42	+39.27	+6.97	0.133
RPA-MPPI	50	100	+7.54	+23.16	**0.045**	**96**	+9.29	+21.73	**0.046**	**88**	+10.11	+23.11	**0.047**

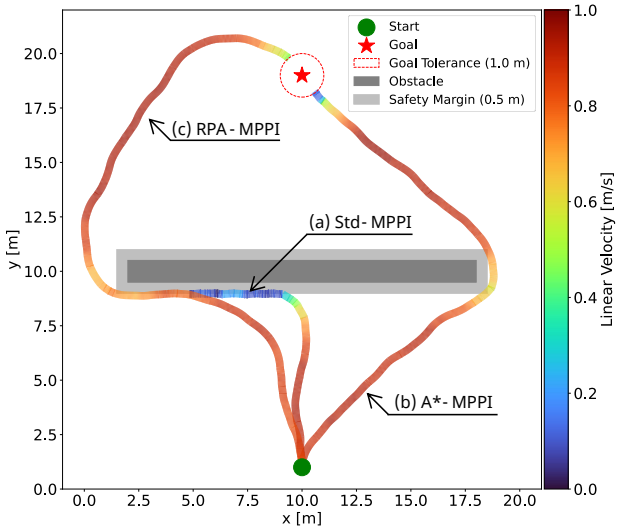


Fig. 4. Comparison of executed trajectories with Long-wid obstacle. (a) Std-MPPI: Fails to reach the goal due to local minima. (b) A*-MPPI: Reaches the goal with a reference path. (c) RPA-MPPI (ours): Reaches the goal without a reference path, adjusting speed and trajectory near obstacles.

RPA-MPPI shows higher values in RDST compared to Std-MPPI, suggesting lower goal-reaching efficiency. RPA-MPPI also shows higher values in RDPL, indicating that the main factor for this inefficiency is the deviation from the shortest path. Fig. 4 provides qualitative insights into how different planners behave: RPA-MPPI successfully avoids local minima as the robot maintains its velocity throughout navigation. In contrast, Std-MPPI struggles to recover once trapped in local minima. Furthermore, comparing the paths traversed by RPA-MPPI and A*-MPPI reveals that the former deviates from the shortest path.

D. Limitation and Possible Extensions

1) Dynamic Local Minima Detection: As mentioned in IV-C, RPA-MPPI tends to traverse inefficient paths. This suggests that depending on the size of obstacles encountered, there may be situations where employing Std-MPPI is advantageous. Additionally, although we assume known local minima coordinates, it is rarely the case in real-world applications. We thus conducted additional experiments: RPA-MPPI with incorrectly identified local minima. As a result, it showed a failure in reaching the goal. These results highlight the importance of accurate local minima detection and dynamic cost function adaptation in real-world applications. We assume such detection can be achieved by analyzing MPPI's output trajectories, potentially enabling local minima identification without global search and maintaining our method's performance advantage.

2) Non-Convex Obstacle Handling: Our method is optimized for convex obstacles, which presents challenges when dealing with non-convex shapes. While it may handle simple non-convex obstacles to some extent, it is likely to struggle with complex configurations such as U-shaped obstacles. Future work will focus on extending our approach to manage non-convex obstacles, possibly by decomposing or abstracting them into convex representations.

V. CONCLUSION

We propose a motion planning method integrating a repulsive potential-augmented cost into the MPPI framework, called RPA-MPPI. This approach guides robots around obstacles while ensuring global convergence. Theoretical analysis and simulation experiments has validated its effectiveness, particularly in environments with wide convex obstacles where squared Euclidean distance-based cost often fails. Key challenges for real-world applications of RPA-MPPI include dynamic local minima detection and non-convex obstacle handling. Future work will focus on developing robust methods for these challenges, as well as extending the approach to 3D navigation scenarios.

APPENDIX

This appendix provides proofs for the properties of $g(\mathbf{p}) = g(x, y)$ discussed in III-E. As a preliminary step, we present the partial derivatives of $g(x, y)$ for $[x, y]^\top \notin \{[0, 0]^\top, [0, y_{\text{goal}}]^\top\}$:

$$\frac{\partial g(x,y)}{\partial x} = \frac{x}{\sqrt{x^2 + (y_{\text{goal}} - y)^2}} - \frac{\alpha x}{\sqrt{x^2 + y^2}}, \tag{13}$$

$$\frac{\partial g(x,y)}{\partial y} = \frac{-(y_{\text{goal}} - y)}{\sqrt{x^2 + (y_{\text{goal}} - y)^2}} - \frac{\alpha y}{\sqrt{x^2 + y^2}}. \tag{14}$$

Property 1. $\frac{\partial g(x,0)}{\partial x} < 0$ for $0 < x \leq \frac{W}{2}$.

Proof. For $0 < x$, the inequality $\frac{\partial g(x,y)}{\partial x} < 0$ can be transformed as follows:

$$\begin{aligned} &\frac{\partial g(x,y)}{\partial x} < 0 \\ \iff &\frac{1}{\sqrt{x^2 + (y_{\text{goal}} - y)^2}} - \frac{\alpha}{\sqrt{x^2 + y^2}} < 0 \\ \iff &x^2 < \frac{\alpha^2 (y_{\text{goal}} - y)^2 - y^2}{1 - \alpha^2}. \end{aligned} \tag{15}$$

Given that $y \leq 0$, $0 < y_{\text{goal}}$ and $0 < \alpha < 1$, the RHS of the final inequality in (15) is always positive. Therefore, it can be transformed as follows:

$$0 < x < \sqrt{\frac{\alpha^2(y_{\text{goal}} - y)^2 - y^2}{1 - \alpha^2}}. \tag{16}$$

In particular, at $y = 0$, equation (16) becomes

$$0 < x < y_{\text{goal}} \sqrt{\frac{\alpha^2}{1 - \alpha^2}}. \tag{17}$$

As α approaches 1, $y_{\text{goal}}\sqrt{\frac{\alpha^2}{1-\alpha^2}}$ grows without bound. By selecting an appropriate value for α sufficiently close to 1, this property can be theoretically satisfied for any given W.

Property 3. $\frac{\partial g(x,y)}{\partial y} < 0$ for $0 \leq y \leq H$.

Proof. Given that $H < y_{\text{goal}}$, the following inequality holds for $0 < y \leq H$:

$$\begin{aligned}\frac{\partial g(x,y)}{\partial y} &= \frac{-(y_{\text{goal}} - y)}{\sqrt{x^2 + (y_{\text{goal}} - y)^2}} - \frac{\alpha y}{\sqrt{x^2 + y^2}} \\ &\leq \frac{-(y_{\text{goal}} - y)}{\sqrt{x^2 + (y_{\text{goal}} - y)^2 + y^2}} - \frac{\alpha y}{\sqrt{x^2 + (y_{\text{goal}} - y)^2 + y^2}}.\end{aligned} \tag{18}$$

Since $0 < \alpha < 1$, RHS of (18) is always negative. $\frac{\partial g(x,y)}{\partial y}$ is also smaller than or equal to this negative RHS. Therefore, $\frac{\partial g(x,y)}{\partial y}$ is always negative, and the property is satisfied.

Property 4. $g(0, y_{goal})$ is the unique global minimum, and has no local minima.

Proof. $g(x, y)$ is differentiable and continuous for $[x, y]^\top \notin \{[0, 0]^\top, [0, y_{\text{goal}}]^\top\}$, and also continuous at $[x, y]^\top = [0, 0]^\top$ and $[0, y_{\text{goal}}]^\top$ shown by the following limits:

$$\begin{aligned}\lim_{[x,y]\to[0,0]} g(x, y) &= \lim_{r\to 0} g(r\cos\psi, r\sin\psi) \\ &= \lim_{r\to 0} \left(\sqrt{(r\cos\psi)^2 + (y_{\text{goal}} - r\sin\psi)^2} - \alpha r \right) \\ &= y_{\text{goal}} = g(0, 0),\end{aligned} \tag{19}$$

$$\begin{aligned}\lim_{[x,y]\to[0,y_g]} g(x, y) &= \lim_{r\to 0} g(r\cos\psi, r\sin\psi + y_{\text{goal}}) \\ &= \lim_{r\to 0} \left(r - \alpha\sqrt{(r\cos\psi)^2 + (r\sin\psi + y_{\text{goal}})^2} \right) \\ &= -\alpha y_{\text{goal}} = g(0, y_{\text{goal}}).\end{aligned} \tag{20}$$

Let $I = \{(x, y) \in \mathbb{R}^2 | x^2 + y^2 \leq R^2\}$ be a bounded closed set for a sufficiently large positive real number R. By the Extreme Value Theorem, maximum and minimum values are guaranteed on this set. The following limit demonstrates that $g(x, y)$ grows unbounded as r approaches infinity, ensuring that the minimum value exists within $x^2 + y^2 < R^2$:

$$\begin{aligned}&\lim_{r\to\infty} g(r\cos\psi, r\sin\psi) \\ &= \lim_{r\to\infty} \left(\sqrt{(r\cos\psi)^2 + (y_{\text{goal}} - r\sin\psi)^2} - \alpha r \right) \\ &= \lim_{r\to\infty} r \left(\sqrt{1 - \frac{2y_{\text{goal}}\sin\psi}{r} + \frac{y_{\text{goal}}^2}{r^2}} - \alpha \right) = \infty.\end{aligned} \tag{21}$$

Given $\alpha \in (0, 1)$, there are no stationary points where both equations (13) and (14) equal zero simultaneously. This implies that differentiable points cannot be extrema, and consequently, cannot be the minimum. Therefore, the minimum must occur at either $g(0, y_{\text{goal}})$ or $g(0, 0)$. From equations (19) and (21), we can deduce that $g(0, y_{\text{goal}}) < g(0, 0)$, so $g(0, y_{\text{goal}})$ is the minimum. Moreover, from Properties 1 and 2, we know that $g(0, 0)$ is not even a local minimum. Hence, we can conclude that $g(0, y_{\text{goal}})$ is the unique global minimum, and there are no local minima.

REFERENCES

[1] G. Williams *et al.*, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1603–1622, 2018.

[2] B. Vlahov *et al.*, "Mppi-generic: A cuda library for stochastic optimization," 2024. [Online]. Available: https://arxiv.org/abs/2409.07563

[3] M. Kazim *et al.*, "Recent advances in path integral control for trajectory optimization: An overview in theoretical and algorithmic perspectives," *Annu. Rev. Control*, vol. 57, p. 100931, 2024.

[4] S. Macenski *et al.*, "The marathon 2: A navigation system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020.

[5] G. Klančar and M. Seder, "Combined stochastic-deterministic predictive control using local-minima free navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 5788–5793.

[6] C. Wang *et al.*, "Chase and track: Toward safe and smooth trajectory planning for robotic navigation in dynamic environments," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 1, pp. 604–613, 2023.

[7] L. Campos-Macías *et al.*, "Autonomous navigation of mavs in unknown cluttered environments," *J. Field Robot.*, vol. 38, no. 2, pp. 307–326, 2020.

[8] Y. Kuwata *et al.*, "Motion planning in complex environments using closed-loop prediction," in *AIAA Guid. Nav. Control Conf. Exhib.*, ser. AIAA 2008-7166, 2008.

[9] A. Sarvesh *et al.*, "Reshaping local path planner," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6534–6541, 2022.

[10] T. Power and D. Berenson, "Learning a generalizable trajectory sampling distribution for model predictive control," *IEEE Trans. Robot.*, vol. 40, pp. 2111–2127, 2024.

[11] C. Warren, "Global path planning using artificial potential fields," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1989, pp. 316–321 vol.1.

[12] C. Pan *et al.*, "Flocking of under-actuated unmanned surface vehicles via deep reinforcement learning and model predictive path integral control," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–11, 2024.

[13] H. Lee *et al.*, "Learning terrain-aware kinodynamic model for autonomous off-road rally driving with model predictive path integral control," *IEEE Robot. Autom. Lett.*, vol. 8, no. 11, pp. 7663–7670, 2023.

[14] T. Kim, J. Mun, J. Seo, B. Kim, and S. Hong, "Bridging Active Exploration and Uncertainty-Aware Deployment Using Probabilistic Ensemble Neural Network Dynamics," in *Proc. Robot. Sci. Syst.*, Daegu, Republic of Korea, July 2023.

[15] S. Jung *et al.*, "V-STRONG: Visual self-supervised traversability learning for off-road navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2024, pp. 1766–1773.

[16] X. Cai *et al.*, "EVORA: Deep evidential traversability learning for risk-aware off-road autonomy," *IEEE Trans. Robot.*, vol. 40, pp. 3756–3777, 2024.

[17] H. Xue *et al.*, "Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing," 2024. [Online]. Available: https://arxiv.org/abs/2409.15610

[18] P. E. Hart *et al.*, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968.

[19] K. Honda, "mppi_playground," accessed: Aug. 1, 2024. [Online]. Available: https://github.com/kohonda/mppi_playground

[20] M. V. Gasparino *et al.*, "Wayfast: Navigation with predictive traversability in the field," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10 651–10 658, 2022.