2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10611021

Stein Variational Guided Model Predictive Path Integral Control: Proposal and Experiments with Fast Maneuvering Vehicles

Kohei Honda[1], Naoki Akai[1], Kosuke Suzuki[1],
Mizuho Aoki[1], Hirotaka Hosogaya[1], Hiroyuki Okuda[1], and Tatsuya Suzuki[1]

***Abstract*— This paper presents a novel Stochastic Optimal Control (SOC) method based on Model Predictive Path Integral control (MPPI), named Stein Variational Guided MPPI (SVG-MPPI), designed to handle rapidly shifting multimodal optimal action distributions. While MPPI can find a Gaussian-approximated optimal action distribution in closed form, *i.e.*, without iterative solution updates, it struggles with the multimodality of the optimal distributions. This is due to the less representative nature of the Gaussian. To overcome this limitation, our method aims to identify a target mode of the optimal distribution and guide the solution to converge to fit it. In the proposed method, the target mode is roughly estimated using a modified Stein Variational Gradient Descent (SVGD) method and embedded into the MPPI algorithm to find a closed-form "mode-seeking" solution that covers only the target mode, thus preserving the fast convergence property of MPPI. Our simulation and real-world experimental results demonstrate that SVG-MPPI outperforms both the original MPPI and other state-of-the-art sampling-based SOC algorithms in terms of path-tracking and obstacle-avoidance capabilities.**
`https://github.com/kohonda/proj-svg_mppi`

I. INTRODUCTION

Path tracking and obstacle avoidance are essential capabilities required for autonomous mobile robots. These tasks become especially challenging for fast maneuvering vehicles because the optimal action distribution may be multimodal and rapidly shifting. To solve these tasks, sampling-based Model Predictive Control (MPC) [1], [2] is a widely adopted approach that can handle the non-linearity and non-differentiability of the environment, such as system dynamics and cost maps, in contrast to gradient-based methods [3]–[6].

Among various sampling-based MPCs, sampling-based Stochastic Optimal Control (SOC) is a relatively sample-efficient approach that approximates the optimal action distribution as the solution, based on a given prior action distribution [7]–[9]. In particular, Model Predictive Path Integral control (MPPI) [9] stands out as a promising framework because it can estimate a Gaussian-approximated optimal action distribution by analytically minimizing the Kullback-Leibler (KL) divergence. That is, it can find the optimal solution in *closed form* (*i.e.*, without iterative solution updates, similar to gradient descent) when given a sufficient number of samples.

MPPI, however, has limitations in capturing complex optimal distributions due to its less representative of the Gaussian. For instance, as illustrated in Fig. 1, an MPPI-based motion planner may produce a collision trajectory as its *optimized* trajectory because the Gaussian-approximated action distribution encompasses the multimodal optimal distribution (see Section III-B for details). To address these shortcomings, several studies have proposed advanced techniques that capture the multimodality using more representative prior distributions [10]–[14] or find *mode-seeking* solutions by leveraging the asymmetry of the KL divergence [15]. However, unlike MPPI, these methods cannot find the solution in closed form. In other words, they require iterative solution updates with sampling and evaluation to converge the solution. The iterative updates degrade the convergence

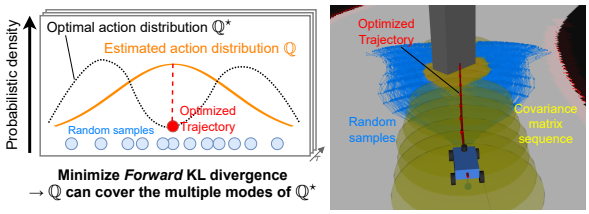
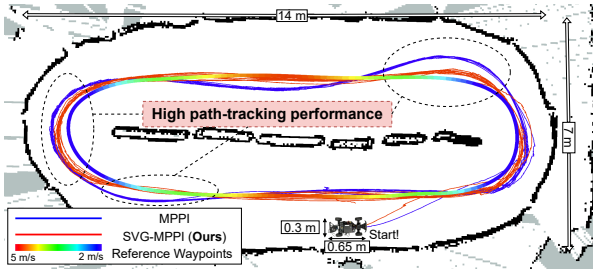
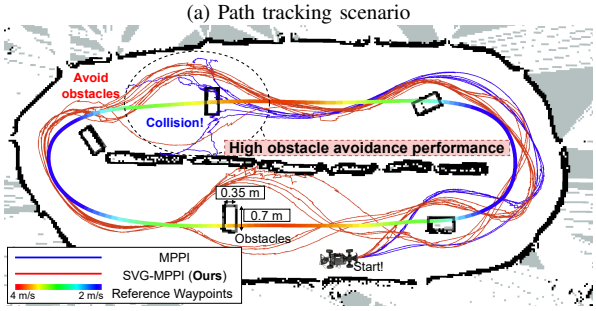


Fig. 1. An open issue of MPPI. The MPPI algorithm minimizes the *Forward* KL divergence instead of the original stochastic optimal control problem. As a result, the estimated action distribution $\mathbb{Q}$ may cover the multiple modes of the optimal action distribution $\mathbb{Q}^*$, and it may lead to finding the collision trajectory as the *optimized* one.



(a) Path tracking scenario



(b) Obstacle avoidance scenario

Fig. 2. Vehicle trajectories in the real-world experiment. Our proposed method, named SVG-MPPI, shows smaller curve overshoots than the original MPPI (Fig. 2a) and robustly avoids obstacles where vanilla MPPI collides (Fig. 2b). These results can be found at: https://www.youtube.com/watch?v=ML_aOYQIDL0

*This work was supported by Tateishi Science and Technology Foundation Research Grants (C) and JSPS KAKENHI Grant Number JP23KJ1067.

[1]The Department of Mechanical Systems Engineering, Graduate School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Aichi, Japan, honda.kohei.b0@s.mail.nagoya-u.ac.jp

of the solution compared to MPPI because the solution is suboptimal and the sequential iteration process restricts the parallelism and the number of samples per iteration.

In this paper, we propose Stein Variational Guided MPPI (SVG-MPPI), a sampling-based SOC method based on MPPI to address rapidly shifting multimodal optimal action distributions. Contrary to the existing methods that capture the complex multimodal distribution, our approach aims to narrow down a single target mode within the multimodal distribution and approximate it with a Gaussian distribution by the MPPI algorithm. As a result, our method can obtain a mode-seeking action distribution in closed form, *i.e.*, preserving the fast solution convergence property of MPPI.

Specifically, our proposed method first roughly identifies the target mode by utilizing a small set of samples (hereafter referred to as *guide particles*) and a modified Stein Variational Gradient Descent (SVGD) method [16]. The SVGD method transports the guide particles to near the peak of the target mode within the optimal distribution using its surrogate gradients. Based on the insight that the transport trajectory leading to the peak represents a part of the shape of the target mode, we estimate the rough variance of the target mode using the Gaussian fitting method. Our method then incorporates the peak and variance into the MPPI algorithm to converge the solution to cover only the target mode.

Our method has been validated through simulation and real-world experiments, focusing on path tracking and obstacle avoidance tasks for a 1/10th scale vehicle[1]. Empirical results demonstrate that SVG-MPPI outperforms standard MPPI [9] and other state-of-the-art SOC algorithms [10], [15] regarding the path-tracking and obstacle-avoidance.

II. RELATED WORK

To address the rapidly shifting complex optimal distribution in the sampling-based SOC, existing approaches can be broadly categorized into the following three types:

1) Increasing Effective Samples: One approach to enhance the MPPI algorithm is to boost the amount of low-state-cost, feasible samples for superior solutions. Besides learning task-specific prior distributions offline [17], [18], the practical approach involves the online adaptation of random samples from a fixed prior distribution, *e.g.*, using auxiliary controllers [19]–[21], gradient descent techniques [22], [23], or adaptive importance sampling [24]. However, most methods are limited to differentiable optimal control problems. In contrast, SV-MPC [10] can transport samples online even in non-differentiable cases by using surrogate gradients. This method leads to a significant increase in computational cost per sample, which limits the number of samples. Instead of transporting all samples, our method transports a few samples (guide particles) for integration into MPPI, facilitating solution exploration without spiking computational complexity, and is applicable to non-differentiable problems.

[1]Our system was deployed in a public competition called F1TENTH (https://f1tenth.org/) held at IEEE ICRA 2023, London, where we placed 4th out of 22 teams.

2) Approximating Multimodal Distribution: The second approach is to approximate the complex optimal action distribution with a more representative model, such as Gaussian mixture models, through variational inference methods [10]–[14]. Although these approaches excel at capturing multimodality, they cannot find the solution in closed form and necessitate iterative solution updates, unlike MPPI. Moreover, practical control problems require a deterministic solution for the system input rather than a probabilistic multimodal solution. Extracting a smoothly shifting single deterministic solution from the obtained multimodal solution is far from trivial. Therefore, a *mode-seeking* solution is necessary for practical application to control problems.

3) Finding a Mode-Seeking Solution: The third approach aims to identify one of the modes of the optimal distribution as the target for convergence, leveraging the asymmetry of the KL divergence [15]. The KL divergence exhibits an asymmetric property when its arguments are reversed. This approach can obtain a mode-seeking solution by directly minimizing KL divergence reversed from the ones used in the original MPPI. However, minimizing the reverse KL divergence cannot be solved in closed form. Our approach can obtain a mode-seeking solution without sacrificing the closed-form optimality inherent in MPPI by guiding the solution with the reverse KL divergence.

III. REVIEW OF MPPI

In this section, we first review the MPPI theory described in the original literature [9] and then derive the open issue of MPPI to be addressed in this work. Finally, we provide a key property of the KL divergence to address this issue.

A. Review of the MPPI Theory

1) Problem Formulation: The MPPI theory models the control target system as a general discrete time dynamical system $\mathbf{x}_{\tau+1} = \boldsymbol{F}(\mathbf{x}_\tau, \mathbf{v}_\tau),\ \tau \in \{0, \dots, T-1\}$, where $\mathbf{x}_\tau \in \mathbb{R}^n$ is the predictive state vector at time τ, $\mathbf{v}_\tau \in \mathbb{R}^m$ is the input vector at time τ, T is the prediction horizon length. The input vector $\mathbf{v}_\tau$ is injected with Gaussian noise: $\mathbf{v}_\tau \sim \mathcal{N}(\mathbf{u}_\tau, \Sigma_\tau)$, as a prior action distribution, where $\mathbf{u}_\tau \in \mathbb{R}^m$ is the actual input vector and $\Sigma_\tau = \mathrm{diag}\{\sigma_\tau^0, \dots, \sigma_\tau^{m-1}\} \in \mathbb{R}^{m\times m}$ is a given covariance matrix of the Gaussian noise.

MPPI aims to estimate a Gaussian action distribution $\mathbb{Q}$ from randomly generated K input sample sequences, $\boldsymbol{V} = \{\boldsymbol{V}_k\}_{k=0}^{K-1} \in \mathbb{R}^{K\times T\times m}$, where $\boldsymbol{V}_k = \{\mathbf{v}_\tau\}_{\tau=0}^{T-1} \in \mathbb{R}^{T\times m}$. The Probability Density Function (PDF) $q(\boldsymbol{V}_k \mid \boldsymbol{U}_t, \boldsymbol{\Sigma})$ corresponding to $\mathbb{Q}$ is analytically expressed as:

$$q(\boldsymbol{V}_k|\cdot) = Z^{-1}\exp\left(-\frac{1}{2}\sum_{\tau=0}^{T-1}(\mathbf{v}_\tau - \mathbf{u}_\tau)^\top \Sigma_\tau^{-1}(\mathbf{v}_\tau - \mathbf{u}_\tau)\right), \quad (1)$$

where $Z = \sqrt{(2\pi)^{mT}|\boldsymbol{\Sigma}|}$, $\boldsymbol{U}_t = \{\mathbf{u}_\tau\}_{\tau=0}^{T-1} \in \mathbb{R}^{T\times m}$ is the actual control input sequence, and $\boldsymbol{\Sigma} = \mathrm{diag}\{\Sigma_\tau\}_{\tau=0}^{T-1} \in \mathbb{R}^{mT\times mT}$ is a given covariance matrix sequence. To find an optimal control input sequence $\boldsymbol{U}_t^*$, MPPI solves the following SOC problem:

$$\boldsymbol{U}_t^* = \underset{\boldsymbol{U}_t\in\mathcal{U}}{\mathrm{argmin}}\, \mathbb{E}_{\boldsymbol{V}_k\sim\mathbb{Q}}\left[\phi(\mathbf{x}_T) + \sum_{\tau=0}^{T-1}\mathcal{L}(\mathbf{x}_\tau, \mathbf{u}_\tau)\right], \quad (2)$$

where $\mathcal{U}$ is the admissible control set [25], $\mathbf{x}_0 = \mathbf{x}_t$ is the observed state at time t, ϕ is the terminal cost function, and $\mathcal{L}$ is the stage cost function. Note that the MPPI theory assumes that the stage cost can be divided into state and input costs as, $\mathcal{L}(\mathbf{x}_\tau, \mathbf{u}_\tau) = c(\mathbf{x}_\tau) + \frac{\lambda}{2}(\mathbf{u}_\tau^\top \Sigma_\tau^{-1} \mathbf{u}_\tau + \beta_t^\top \mathbf{u}_\tau)$, where β_τ is a given parameter. In the following, the kth *sequence state cost* is defined as, $S(\boldsymbol{V}_k) = \phi(\mathbf{x}_T) + \sum_{\tau=0}^{T-1} c(\mathbf{x}_\tau)$.

Since (2) is difficult to solve directly, MPPI instead minimizes the *Forward* KL (FKL) divergence $\mathbb{D}_{\mathrm{KL}}(\mathbb{Q}^* \parallel \mathbb{Q})$ that the action distribution $\mathbb{Q}$ covers the optimal action distribution $\mathbb{Q}^*$, as follows:

$$\boldsymbol{U}_t^* \simeq \underset{\boldsymbol{U}_t \in \mathcal{U}}{\operatorname{argmin}} \{\mathbb{D}_{\mathrm{KL}}(\mathbb{Q}^* \parallel \mathbb{Q})\}. \tag{3}$$

2) Analytical PDF of $\mathbb{Q}^*$: To minimize the FKL divergence in (3), the optimal PDF $q^*(\boldsymbol{V}_k)$ of the $\mathbb{Q}^*$ is derived in the following. First, the *Free-energy* of the system is defined as, $\mathcal{F}(\boldsymbol{V}_k) = \log\left(\mathbb{E}_{\mathbb{P}}\left[\exp\left(-\frac{1}{\lambda} S(\boldsymbol{V}_k)\right)\right]\right)$, where λ is a given temperature parameter and $\mathbb{P}$ is a base distribution, which is roughly analogous to a Bayesian prior. The PDF of the $\mathbb{P}$ is $q(\boldsymbol{V}_k \mid \tilde{\boldsymbol{U}}, \boldsymbol{\Sigma})$, where $\tilde{\boldsymbol{U}} = \{\tilde{\mathbf{u}}_\tau\}_{\tau=0}^{T-1} \in \mathbb{R}^{T \times m}$ represents a given nominal control input sequence (hereafter referred to as *nominal sequence*). The Free-energy is expanded by Jensen's inequality and (1) as,

$$-\lambda \mathcal{F}(\boldsymbol{V}_k) \leq \mathbb{E}_{\boldsymbol{V}_k \sim \mathbb{Q}}\left[\phi(\mathbf{x}_T) + \sum_{\tau=0}^{T-1} \mathcal{L}(\mathbf{x}_\tau, \mathbf{u}_\tau)\right]. \tag{4}$$

Because the right hand side of (4) is equivalent to the cost function in (2), $-\lambda\mathcal{F}(\boldsymbol{V}_k)$ represents the lower bound of the original SOC problem. Thus, the optimal PDF $q^*(\boldsymbol{V}_k)$ is derived from the equality condition of (4) as,

$$q^*(\boldsymbol{V}_k \mid \tilde{\boldsymbol{U}}, \boldsymbol{\Sigma}) = \eta^{-1} \exp\left(-\frac{1}{\lambda} S(\boldsymbol{V}_k)\right) q(\boldsymbol{V}_k \mid \tilde{\boldsymbol{U}}, \boldsymbol{\Sigma}), \tag{5}$$

where $\eta = \int_{\mathbb{R}^{T \times m}} p(\boldsymbol{V}_k \mid \lambda) \exp\left(-\frac{1}{\lambda} S(\boldsymbol{V}_k)\right) \mathrm{d}\boldsymbol{V}_k$.

3) Forward KL Divergence Mimimization: As mentioned in section III-A.1, the approach of MPPI is minimizing the FKL divergence in (3), which can be transformed by the definition of the FKL divergence and (1) into the following closed-form expression:

$$\begin{aligned} \boldsymbol{U}_t^* &= \underset{\boldsymbol{U}_t \in \mathcal{U}}{\operatorname{argmin}} \mathbb{E}_{\boldsymbol{V}_k \sim \mathbb{Q}^*}\left[\frac{1}{2}\sum_{\tau=0}^{T-1} (\mathbf{v}_\tau - \mathbf{u}_\tau)^\top \Sigma_\tau^{-1} (\mathbf{v}_\tau - \mathbf{u}_\tau)\right] \\ &= \mathbb{E}_{\boldsymbol{V}_k \sim \mathbb{Q}^*}[\boldsymbol{V}_k]. \end{aligned} \tag{6}$$

Thus, the optimal control input sequence $\boldsymbol{U}_t^*$ is obtained by sampling $\boldsymbol{V}_k$ from the optimal action distribution $\mathbb{Q}^*$ and taking its expected value, as in (6). However, since sampling directly from $\mathbb{Q}^*$ is not possible, importance sampling [26] is used with the $q^*(\boldsymbol{V}_k \mid \tilde{\boldsymbol{U}}, \boldsymbol{\Sigma})$ in (5), as follows:

$$\boldsymbol{U}_t^* = \mathbb{E}_{\boldsymbol{V}_k \sim \mathbb{Q}}\left[\frac{q^*(\boldsymbol{V}_k \mid \tilde{\boldsymbol{U}}, \boldsymbol{\Sigma})}{q(\boldsymbol{V}_k \mid \hat{\boldsymbol{U}}, \boldsymbol{\Sigma})} \boldsymbol{V}_k\right] \simeq \sum_{k=0}^{K-1} w(\boldsymbol{V}_k) \boldsymbol{V}_k, \tag{7}$$

$$w(\boldsymbol{V}_k) = \eta^{-1} \exp\left(-\frac{1}{\lambda} S(\boldsymbol{V}_k) - \sum_{\tau=0}^{T-1} (\hat{\mathbf{u}}_\tau - \tilde{\mathbf{u}}_\tau)^\top \Sigma_\tau^{-1} \mathbf{v}_\tau\right),$$

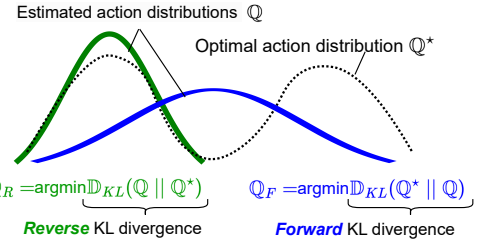


Fig. 3. Asymmetry properties of the KL divergence [15]. When minimizing the *Forward* KL (FKL) divergence, it results in covering multiple modes within the optimal action distribution. This is because FKL divergence imposes a significant penalty when $q^* > 0$ and $q \approx 0$. In contrast, minimizing the *Reverse* KL (RKL) divergence converges to a single mode in the optimal distribution, a property known as *mode-seeking*. This happens because RKL divergence takes on a larger value when $q > 0$ and $q^* \approx 0$.

where $\hat{\boldsymbol{U}} = \{\hat{\mathbf{u}}_\tau\}_{\tau=0}^{T-1}$ is the initial estimated control input sequence, often using the previous optimal solution. Importantly, based on the law of large numbers, the MPPI algorithm can obtain a globally optimal solution in the sense of minimizing the FKL divergence in closed form, *i.e.*, the optimal solution can be obtained by sampling sufficiently large K samples once and taking their weighted average, without iterative solution updates.

B. An Open Issue of MPPI

The MPPI algorithm can estimate the Gaussian-approximated optimal distribution by analytically minimizing the FKL divergence in closed form. However, the Gaussian approximation causes an open issue that the estimated action distribution may cover the multiple modes of the optimal distribution, as illustrated in Fig. 3. In the context of an obstacle avoidance task, this issue may lead MPPI to identify the collision trajectory as the optimal solution, as shown in Fig.1. This occurs because the two branching paths for avoiding obstacles correspond to the peaks of the two modes, and the estimated action distribution covers both modes. To solve this problem, we need to find a mode-seeking action distribution that covers a single target mode of the multimodal distribution.

C. Asymmetry Property of KL divergence

One key to finding the mode-seeking solution is to utilize the asymmetry property of the KL divergence, as shown in Fig. 3. The standard MPPI algorithm minimizes *Forward* KL (FKL) divergence, denoted as $\mathbb{D}_{\mathrm{KL}}(\mathbb{Q}^* \parallel \mathbb{Q})$, where $\mathbb{Q}$ may cover the entire $\mathbb{Q}^*$. In contrast, when the input arguments are reversed, the *Reverse* KL (RKL) divergence $\mathbb{D}_{\mathrm{KL}}(\mathbb{Q} \parallel \mathbb{Q}^*)$ exhibits a different characteristic, where $\mathbb{Q}$ converges to a single mode of the optimal distribution $\mathbb{Q}^*$.

IV. STEIN VARIATIONAL GUIDED MPPI

Although some existing methods [10], [15] can find the mode-seeking solution by directly minimizing the RKL divergence described in Section III-C, the solution cannot be obtained in closed form, resulting in a degradation of convergence. To overcome the limitation, our approach guides the MPPI solution to converge to a single target mode roughly identified using a modified SVGD method.

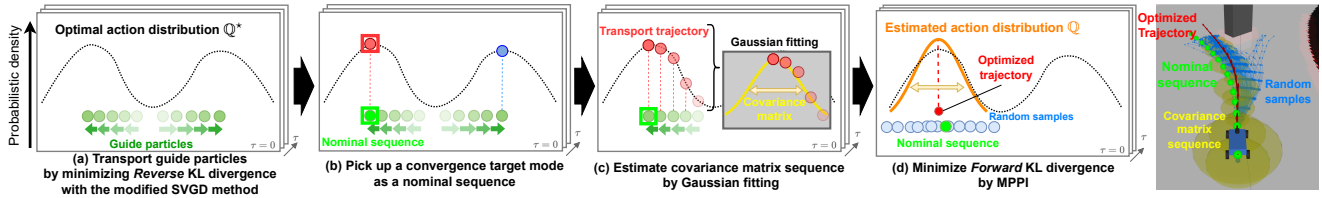


Fig. 4. Overview of the proposed method. (a) SVG-MPPI first transports guide particles by minimizing the RKL divergence using the modified SVGD method. These transport trajectories of the guide particles are employed to roughly estimate a convergent target mode. (b) We then identify the peak of the target mode by simply taking the lowest sequence cost guide particle as a nominal sequence. (c) The variances are also roughly estimated by the Gaussian fitting algorithm with the transport trajectory. (d) Finally, SVG-MPPI minimizes the FKL divergence, incorporating the nominal sequence and adaptive covariance matrices in the process. The SVG-MPPI can efficiently find a mode-seeking action distribution compared to the original MPPI (Fig. 1).

As an overview of our method illustrated in Fig. 4, we first spread a small set of samples (guide particles) and transport them by the modified SVGD method. Because the SVGD method iteratively reduces the RKL divergence based on the surrogate gradients of the optimal action distribution $\mathbb{Q}^*$, one of the guide particles is transported to near the peak of the target mode. We then roughly estimate the variance of the target mode by fitting the transport trajectory leading to the peak as Gaussian. Finally, we incorporate the peak and variance of the target mode (as the nominal sequence and covariance matrix sequence) into the MPPI algorithm to converge the solution to cover only the target mode.

A. Transport Guide Particles by the Modified SVGD method

Our proposed method first spreads K_g guide particles $\{\boldsymbol{V}_k^g\}_{k=0}^{K_g-1} \in \mathbb{R}^{K_g \times T \times m}$ in addition to the K samples used in the standard MPPI algorithm. Note that $K_g \ll K$, in particular, we use $K_g = 1$ in our experiments in section V. The guide particles are transported to make the RKL divergence smaller by applying the following gradient descent update with L iterations per control cycle given a small step size ε,

$$\hat{\boldsymbol{V}}_k^g \leftarrow \boldsymbol{V}_k^g - \varepsilon \nabla_{\boldsymbol{V}_k^g} \mathbb{D}_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{Q}^*), \ {}^\forall k \in \{0, \dots, K_g - 1\}. \quad (8)$$

The RKL divergence $\mathbb{D}_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{Q}^*)$ is expanded from its definition and Jensen's inequality as,

$$\begin{aligned} \mathbb{D}_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{Q}^*) &= -\mathbb{E}_{\boldsymbol{V}_k^g \sim \mathbb{Q}} \left[\log \frac{q^*}{q} \right] = -\mathbb{E}_{\boldsymbol{V}_k^g \sim \mathbb{Q}} \left[\log w(\boldsymbol{V}_k^g) \right] \\ &\leq -\log \mathbb{E}_{\boldsymbol{V}_k^g \sim \mathbb{Q}} \left[w(\boldsymbol{V}_k^g) \right], \end{aligned} \quad (9)$$

where $w(\cdot)$ is the weight function in (7). Now, we can reduce the RKL divergence of the guide particles by assigning the gradient of (9) into (8) since the right hand side of (9) is the upper bound of the RKL divergence. However, the weight function $w(\cdot)$ has the sequence state cost $S(\boldsymbol{V}_k^g)$, which is generally assumed to be non-differentiable. Therefore, we estimate the stochastic surrogate gradient described in [10]:

$$\begin{aligned} \nabla_{\boldsymbol{V}_k^g} \mathbb{D}_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{Q}^*) &= -\nabla_{\boldsymbol{V}_k^g} \log \mathbb{E}_{\boldsymbol{V}_k^g \sim \mathbb{Q}} [w(\boldsymbol{V}_k^g)] \\ &= -\frac{\mathbb{E}_{\boldsymbol{V}_k^g \sim \mathbb{Q}} \left[w\left(\boldsymbol{V}_k^g\right) \nabla_{\boldsymbol{V}_k^g} \log q \right]}{\mathbb{E}_{\boldsymbol{V}_k^g \sim \mathbb{Q}} [w(\boldsymbol{V}_k^g)]} \\ &\simeq -\frac{\sum_{i=0}^{N-1} \left\{ w\left(\boldsymbol{V}_k^g[i]\right) \nabla_{\boldsymbol{V}_k^g} \log q\left(\boldsymbol{V}_k^g[i]\right) \right\}}{\sum_{i=0}^{N-1} w\left(\boldsymbol{V}_k^g[i]\right)}, \end{aligned} \quad (10)$$

where $\boldsymbol{V}_k^g[i]$ is a sample for the Monte Carlo estimation, *i.e.*, $\boldsymbol{V}_k^g[i] \sim q(\boldsymbol{V}_k^g[i] \mid \boldsymbol{V}_k^g, \Sigma_g)$, given a constant covariance matrix Σ_g. The update equation in (8) converges to a simplified version of the update equation using the SVGD method in [10], with the kernel function and log-prior terms removed. These removed terms penalize particle cohesion and preserve the diversity of the samples. However, for our approach, the penalty terms are arbitrarily removed because we expect the guide particles to aggregate to some of the peaks of the optimal distribution. Note that the computational complexity for the L times transport in (8) is $\mathcal{O}(LK_gN)$ and relatively tiny compared to the MPPI's one (*i.e.*, $\mathcal{O}(TKm)$) when $K_g \ll K$.

B. FKL divergence Minimization with the Nominal Sequence and Adaptive Covariance Matrix Sequence

We now obtain K_g transport trajectories for each guide sample as, $\tau_k^g = \{\boldsymbol{V}_k^g[l]\}_{l=0}^{L-1} \in \mathbb{R}^{L \times T \times m}, k \in \{0, \dots, K_g - 1\}$, where L is the number of iterations for (8). We then roughly identify the peak and variance of the target mode using the transport trajectories and incorporate them as the nominal sequence $\tilde{\boldsymbol{U}}$ and covariance matrix sequence $\boldsymbol{\Sigma}$ in the MPPI algorithm described in Section III-A.

1) Picking a Target Mode: The guide particles transported in sufficiently large iterations L are expected to be close to some of the peaks. Since the convergence goal of a gradient descent depends on the initial value, the guide particles can be transported to different peaks within the optimal distribution after the iterations. The strategy for determining the peak of the target mode is simply to pick up a transported guide particle that has the lowest sequence state cost. Therefore, the peak, as the nominal sequence $\tilde{\boldsymbol{U}}$ of the MPPI algorithm, is determined, as follows:

$$\tilde{\boldsymbol{U}} = \boldsymbol{V}_{k^*}^g, \ \text{where } k^* = \underset{k}{\operatorname{argmin}} \left\{ S\left(\boldsymbol{V}_k^g[L-1]\right) \right\}_{k=0}^{K_g-1}. \quad (11)$$

The nominal sequence $\tilde{\boldsymbol{U}}$ is used in the weight function (7), which penalizes the deviation of the nominal sequence (the peak of the target mode) and the mean of the action distribution, which corresponds to determining the coverage target mode within the optimal action distribution.

2) Estimation of the Adaptive Covariance Matrix Sequence: To converge the action distribution to cover only the target mode, we should set an appropriate covariance matrix sequence $\boldsymbol{\Sigma}$, which is used in the prior action distribution of random samples and weight function in (7) for the MPPI algorithm. Our method utilizes the entire k^*th transport trajectory $\tau_{k^*}^g$ to estimate the rough variance of the target mode be-

cause the transport trajectory leading to the peak is expected to represent a part of the shape of the target mode. Specifically, we use a Gaussian fitting algorithm known as fast and rough [27]. For all predictive times and control inputs ${}^{\forall}\tau \in [0,\dots,T-1]$ and ${}^{\forall}i \in [0,\dots,m-1]$, let the input vector be $\mathbf{a} = \left[\boldsymbol{V}^g_{k^*}[0,\tau,i],\dots,\boldsymbol{V}^g_{k^*}[L-1,\tau,i]\right] \in \mathbb{R}^L$ and the corresponding q^* in (5) be $\mathbf{b} = \left[q^*\left(\boldsymbol{V}^g_{k^*}[0]\right),\dots,q^*\left(\boldsymbol{V}^g_{k^*}[L-1]\right)\right] \in \mathbb{R}^L$, and the variance $\Sigma_\tau[i] = \sigma^i_\tau$ for each control input can be estimated by solving the first-order equation:

$$\sigma^i_\tau = \sqrt{\frac{-1}{2z_2}}, \text{ where} \tag{12}$$

$$\begin{bmatrix} \sum \mathbf{b}^2 & \sum \mathbf{a}\mathbf{b}^2 & \sum \mathbf{a}^2\mathbf{b}^2 \\ \sum \mathbf{a}\mathbf{b}^2 & \sum \mathbf{a}^2\mathbf{b}^2 & \sum \mathbf{a}^3\mathbf{b}^2 \\ \sum \mathbf{a}^2\mathbf{b}^2 & \sum \mathbf{a}^3\mathbf{b}^2 & \sum \mathbf{a}^4\mathbf{b}^2 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \sum \mathbf{b}^2 \log \mathbf{b} \\ \sum \mathbf{a}\mathbf{b}^2 \log \mathbf{b} \\ \sum \mathbf{a}^2\mathbf{b}^2 \log \mathbf{b} \end{bmatrix}.$$

As a result, when the change of the q^* due to the transport is small, the variance is estimated to be small, as shown in Fig. 4. In this case, the sequence state cost of the guide particle is largely shifted by the transport. In the opposite case, the variance is kept to be large.

V. EXPERIMENTS

A. Experiment Setup

We implemented our method for a path-tracking and obstacle-avoidance motion planner of a 1/10th-scale vehicle. The motion planner controls the vehicle's steering angle based on given reference waypoints (path and speed profile), the vehicle's position and velocity, and a cost map [28] constructed from onboard 2D LiDAR scans for obstacles and course detection.

1) MPC Formulation: The implemented vehicle dynamics predictive model $\boldsymbol{F}$ was the kinematic bicycle model [29] considering the dead time and first-order delays with the steering angle input, which have nonlineality. The non-differentiable sequence state cost was $S(\boldsymbol{V}_k) = \sum_{\tau=0}^{T=15} \left\{\Delta d_r(\tau)^2 + 0.01\Delta\theta_r(\tau)^2 + 1000\,\mathbf{1}^{\text{collide}}(\mathbf{x}_\tau)\right\}$, where Δd_r and $\Delta\theta_r$ were position and yaw angle deviations from the reference waypoints, and $\mathbf{1}^{\text{collide}}$ is a binary indicator function for collision, provided by the cost map. The step width of the prediction horizon $\Delta\tau$ is 0.05 s. Please refer to the source code[2] for the detailed algorithm and settings.

2) Validation Scenarios: To validate the performance of the proposed method, we conducted tests under two types of driving scenarios, both in simulation and in the real world:

- Path Tracking (PT): The vehicle completed N_{pt} laps of the course without encountering unforeseen obstacles.
- Obstacle Avoidance (OA): The vehicle drove N_{oa} laps of the course, encountering five unforeseen static obstacles. These obstacles were randomly repositioned around reference waypoints on each lap.

3) Evaluation Metrics: To evaluate the path-tracking and obstacle-avoidance capabilities, we employ the following two evaluation metrics:

- Mean Sequence state cost $S(\boldsymbol{U}_t)$ per lap, denoted as MS: A lower value of this metric indicates better performance in path tracking and obstacle avoidance. In scenario PT, this metric is primarily focused on evaluating path-tracking performance. Conversely, scenario OA, mainly evaluates the obstacle-avoidance performance.
- Collision Rate (CR): This represents the rate of collisions with obstacles or the course, relative to the total number of obstacles encountered in scenario OA.

4) Baseline Methods: We implemented the vanilla MPPI [9], Reverse-MPPI with sample rejection [15], and SV-MPC [10] to compare their performance to our proposed method, SVG-MPPI. Reverse-MPPI minimizes the RKL divergence based on mirror descent algorithm [30], and SV-MPC also directly minimizes RKL divergence with variational inference approach, *i.e.*, using SVGD method [16]. Unlike the MPPI that minimizes the FKL divergence, they cannot be solved in closed form, and the number of random samples for the iterative updates must be set smaller than the MPPI and SVG-MPPI to keep real-time processing.

All algorithms, including SVG-MPPI, were developed in ROS and C++ and employed CPU multi-threading via OpenMP [31]. The same MPC formulation and common parameters were used across all algorithms, except the number of random samples. To keep a calculation time of approximately 20 ms on a desktop PC equipped with an Intel Core i9-10850K CPU, in the simulation, we set the number of random samples for MPPI, Reverse-MPPI, SV-MPC, and SVG-MPPI at 10k, 200, 500, and 8k, respectively. Similarly, in the real-world experiment, 5k and 2k random samples are used for MPPI and SVG-MPPI, respectively.

B. Simulation Results

We executed each method for 100 laps in both the PT and OA scenarios to provide a comprehensive comparison ($N_{\text{pt}} = N_{\text{oa}} = 100$). For the simulation environment, we customized the F1TENTH Gym environment [32] to include unforeseen five obstacles that were randomly replaced around reference waypoints (within 0.1 m) on each lap.

1) Comparison with Baseline Methods: Figure 5 shows the evaluation results for the MS metrics in both PT and OA scenarios, and Table I lists the CRs for each method in the OA scenario. In the case of vanilla MPPI, a lower covariance of the control input correlates with lower MSs and CRs in the OA scenario; however, it negatively impacts MSs in PT. These findings suggest that vanilla MPPI faces a trade-off between PT and OA capability depending on the fixed covariance parameter. This is attributable to the fact that a small covariance prevents the solution from covering the entire optimal action distribution in the OA, but is not suitable for PT, which requires covering the entire single-mode distribution to track sharp curves rapidly.

Unlike vanilla MPPI, both Reverse-MPPI and SV-MPC aim to minimize RKL divergence directly. As shown in Fig. 5, Reverse-MPPI outperforms vanilla MPPI in PT, while SV-MPC outperforms in the OA scenario. However,

[2] https://github.com/kohonda/proj-svg_mppi

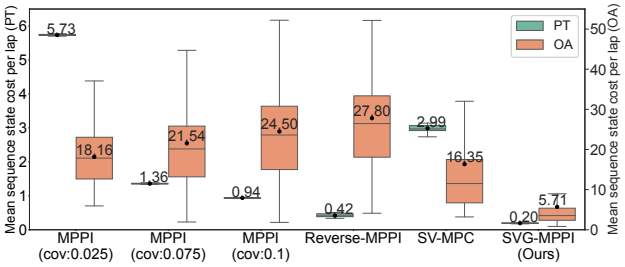


Fig. 5. Comparison of the MSs with baseline methods in the simulation.

TABLE I

COLLISION RATE (CR) FOR EACH METHOD IN THE SIMULATION

Method	MPPI			Reverse	SV	SVG-MPPI
Cov. [rad]	0.025	0.075	0.1	MPPI	MPC	(Ours)
CR [%] ↓	13.6	21.0	24.4	24.8	12.4	**4.0**

each method underperforms in the other scenario; Reverse-MPPI in OA and SV-MPC in PT. In particular, SV-MPC is similar to our method. It transports all samples and takes a weighted average of them, resulting in them being evenly separated, which can degrade the performance of OA, or the performance of PT can be degraded due to the samples being over-compressed.

On the other hand, our proposed method, SVG-MPPI, demonstrates superior performance in both scenarios, as shown in Fig. 5. These results confirm that SVG-MPPI excels in both PT and OA tasks, avoiding the trade-offs observed in all baseline methods. Furthermore, SVG-MPPI records the lowest CR of 4.0 % across all trials in the OA scenario in Table I. In the rare cases where collisions occurred, SVG-MPPI did not find a mode-seeking solution due to its inability to guarantee one.

We also measured the computation time for each method. The average/maximum time to find a solution for each method was 12.1 ms / 34.2 ms for MPPI, 26.0 ms / 55.9 ms for Reverse MPPI, 29.1 ms / 55.9 ms for SV-MPC, and 11.0 ms / 39.7 ms for SVG-MPPI. MPPI and SVG-MPPI can find solutions through closed-form expressions, so even with a large number of samples, the computation time is kept relatively low thanks to the benefits of parallelization. On the other hand, Reverse MPPI and SV-MPC need to update their solutions iteratively, which tends to increase the computational load relative to the number of samples compared to the others.

2) Ablation Study: SVG-MPPI is an extension of vanilla MPPI that incorporates both a Nominal Sequence (NS) and Adaptive Covariance matrix sequence estimation (AC). To demonstrate the complementary nature of these two extensions, we conducted an ablation study for the PT and OA scenarios in the simulation. The results are presented in Table II. Both MPPI and MPPI+NS employed a fixed steering covariance of 0.75 rad. As indicated in Table II, MPPI+NS outperforms vanilla MPPI in both PT and OA scenarios. Moreover, MPPI+AC exhibits even better performance in OA compared to MPPI+NS, although its performance in PT is significantly diminished. When both enhancements (MPPI+NS+AC) are incorporated, SVG-MPPI delivers the best performance in OA and ranks second-best in PT. Overall, these results confirm that the NS and AC functions are complementary and contribute to the enhanced path-tracking and obstacle-avoidance performance. However, as the results for SVG-MPPI and MPPI+NS show, AC slightly hinders PT performance. This is because the proposed AC method, prioritizing computational efficiency, is a rough estimation prone to errors. This is a future work to be addressed.

TABLE II

ABLATION STUDY: AVERAGE OF MSS FOR 100 TRIALS

method	Scenario PT	Scenario OA
MPPI (baseline)	1.36	21.54
MPPI + nominal sequence (NS)	**0.18**	11.03
MPPI + adaptive covariance (AC)	18.18	7.08
MPPI + NS + AC (SVG-MPPI)	0.20	**5.71**

C. Real-world Experiment Results

We also conducted real-world experiments using a 1/10th-scale vehicle equipped with a 2D LiDAR (HOKUYO UTM30-LX) and a compact computer (Intel NUC with a Core i7-1360P Processor). To estimate the vehicle's pose on the pre-built map, we employed an advanced Monte Carlo localization method [33].

Figure 2 shows the trajectories of both vanilla MPPI (steering covariance: 0.075 rad) and SVG-MPPI in PT and OA scenarios with a fixed obstacle layout. The figure reveals that SVG-MPPI exhibits smaller overshoots on curves compared to vanilla MPPI, as further illustrated in Fig. 2a. Additionally, Fig.2b indicates that SVG-MPPI successfully avoids obstacles where vanilla MPPI encounters collisions.

For more evaluation, we completed 10 laps in scenario PT ($N_{\text{pt}} = 10$) and 3 laps for each of six different static obstacle layouts in scenario OA ($N_{\text{pt}} = 18$). In scenario PT, the mean MS values were recorded at 5.83 and 2.03 for vanilla MPPI and SVG-MPPI, respectively. Turning our attention to scenario OA, the mean MS and CR values were (62.73, 55.6 %) for vanilla MPPI and (27.82, 15.4 %) for SVG-MPPI. Overall, SVG-MPPI demonstrated superior path-tracking and obstacle-avoidance capabilities compared to the vanilla MPPI method in real-world testing.

VI. CONCLUSION AND POTENTIAL LIMITATION

This paper presents a novel MPPI-based SOC method to address rapidly shifting multimodal optimal action distributions. Our method is capable of finding a mode-seeking solution in closed form by guiding the convergence target of the MPPI solution using the SVGD method. While our experiments do not show any performance degradation of the proposed method, a potential limitation of our method arises in cases where the gradients of the optimal distribution are zero outside the peaks of the modes. This limitation is due to the fact that the SVGD method primarily tracks gradients, and the solution may become trapped at such terrace-like places within the optimal distribution. Furthermore, since the proposed method is not theoretically tailored for path tracking and obstacle avoidance, it is expected to be applicable to a wider range of robotic challenges.

REFERENCES

[1] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[2] T. M. Howard, C. J. Green, A. Kelly, and D. Ferguson, "State space sampling of feasible motions for high-performance mobile robot navigation in complex environments," *Journal of Field Robotics*, vol. 25, no. 6-7, pp. 325–345, 2008.

[3] M. Diehl and S. Gros, "Numerical optimal control," *Optimization in Engineering Center*, 2011.

[4] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[5] T. Ohtsuka, "A continuation/GMRES method for fast computation of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563–574, 2004.

[6] K. Honda, H. Okuda, T. Suzuki, and A. Ito, "MPC Builder for autonomous drive: Automatic generation of MPCs for motion planning and control," in *Intelligent Vehicles Symposium*. IEEE, 2023, pp. 1–8.

[7] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.

[8] Z. I. Botev, D. P. Kroese, R. Y. Rubinstein, and P. L' Ecuyer, "The cross-entropy method for optimization," in *Handbook of statistics*. Elsevier, 2013, vol. 31, pp. 35–59.

[9] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603–1622, 2018.

[10] A. Lambert, F. Ramos, B. Boots, D. Fox, and A. Fishman, "Stein variational model predictive control," in *Conference on Robot Learning*. PMLR, 2021, pp. 1278–1297.

[11] T. Osa, "Multimodal trajectory optimization for motion planning," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 983–1001, 2020.

[12] M. Okada and T. Taniguchi, "Variational inference MPC for Bayesian model-based reinforcement learning," in *Conference on robot learning*. PMLR, 2020, pp. 258–272.

[13] Z. Wang, O. So, J. Gibson, B. Vlahov, M. S. Gandhi, G.-H. Liu, and E. A. Theodorou, "Variational inference MPC using tsallis divergence," in *Robotics: Science and Systems*, 2021.

[14] L. Barcelos, A. Lambert, R. Oliveira, P. Borges, B. Boots, and F. Ramos, "Dual online Stein variational inference for control and dynamics," in *Robotics: Science and Systems*, 2021.

[15] T. Kobayashi and K. Fukumoto, "Real-time sampling-based model predictive control based on reverse kullback-leibler divergence and its adaptive acceleration," *arXiv preprint arXiv:2212.04298*, 2022.

[16] Q. Liu and D. Wang, "Stein variational gradient descent: A general purpose bayesian inference algorithm," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[17] T. Power and D. Berenson, "Variational inference mpc using normalizing flows and out-of-distribution projection," in *Robotics science and systems*, 2023.

[18] J. Sacks and B. Boots, "Learning sampling distributions for model predictive control," in *Conference on Robot Learning*. PMLR, 2023, pp. 1733–1742.

[19] M. S. Gandhi, B. Vlahov, J. Gibson, G. Williams, and E. A. Theodorou, "Robust model predictive path integral control: Analysis and performance guarantees," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1423–1430, 2021.

[20] J. Yin, Z. Zhang, E. Theodorou, and P. Tsiotras, "Trajectory distribution control for model predictive path integral control using covariance steering," in *International Conference on Robotics and Automation*. IEEE, 2022, pp. 1478–1484.

[21] F. Rastgar, H. Masnavi, B. Sharma, A. Aabloo, J. Swevers, and A. K. Singh, "PRIEST: Projection guided sampling-based optimization for autonomous navigation," *IEEE Robotics and Automation Letters*, 2024 (in print).

[22] H. Bharadhwaj, K. Xie, and F. Shkurti, "Model-predictive control via cross-entropy and gradient-based optimization," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 277–286.

[23] F. Heetmeyer, M. Paluch, D. Bolliger, F. Bolli, X. Deng, E. Filicicchia, and T. Delbruck, "RPGD: A small-batch parallel gradient descent optimizer with explorative resampling for nonlinear model predictive control," in *International Conference on Robotics and Automation*. IEEE, 2023, pp. 3218–3224.

[24] D. M. Asmar, R. Senanayake, S. Manuel, and M. J. Kochenderfer, "Model predictive optimized path integral strategies," in *International Conference on Robotics and Automation*. IEEE, 2023, pp. 3182–3188.

[25] E. C. Kerrigan, "Robust constraint satisfaction: Invariant sets and predictive control," Ph.D. dissertation, University of Cambridge UK, 2001.

[26] T. Kloek and H. K. Van Dijk, "Bayesian estimates of equation system parameters: an application of integration by monte carlo," *Econometrica: Journal of the Econometric Society*, pp. 1–19, 1978.

[27] H. Guo, "A simple algorithm for fitting a gaussian function," *IEEE Signal Processing Magazine*, vol. 28, no. 5, pp. 134–137, 2011.

[28] P. Fankhauser and M. Hutter, "A universal grid map library: Implementation and use case for rough terrain navigation," *Robot Operating System (ROS) The Complete Reference (Volume 1)*, pp. 99–120, 2016.

[29] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[30] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Operations Research Letters*, vol. 31, no. 3, pp. 167–175, 2003.

[31] R. Chandra, *Parallel programming in OpenMP*. Morgan kaufmann, 2001.

[32] M. O'Kelly, H. Zheng, D. Karthik, and R. Mangharam, "F1TENTH: An open-source evaluation environment for continuous control and reinforcement learning," *Proceedings of Machine Learning Research*, vol. 123, 2020.

[33] N. Akai, "Reliable monte carlo localization for mobile robots," *Journal of Field Robotics*, vol. 40, no. 3, pp. 595–613, 2023.