2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

Path Integral Methods with Stochastic Control Barrier Functions

Chuyuan Tao[†], Hyung-Jin Yoon[*], Hunmin Kim[‡], Naira Hovakimyan[†], and Petros Voulgaris[*]

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9993190

***Abstract*— Safe control designs for robotic systems remain challenging because of the difficulties of explicitly solving optimal control with nonlinear dynamics perturbed by stochastic noise. However, recent technological advances in computing devices enable online optimization or sampling-based methods to solve control problems. For example, Control Barrier Functions (CBFs) have been proposed to numerically solve convex optimization problems that ensure the control input to stay in the safe set. Model Predictive Path Integral (MPPI) control uses forward sampling of stochastic differential equations to solve optimal control problems online. Both control algorithms are widely used for nonlinear systems because they avoid calculating the derivatives of the nonlinear dynamic functions. In this paper, we use Stochastic Control Barrier Functions (SCBFs) constraints to limit sample regions in the sampling-based algorithm, ensuring safety in a probabilistic sense and improving sample efficiency with a stochastic differential equation. We also show that our algorithm needs fewer samples than the original MPPI algorithm does by providing a sampling complexity analysis.**

I. Introduction

Safety verification is crucial when applying control algorithms to robotic systems in the presence of uncertainties. Failure to ensure safety may cause severe damage to robots, properties, and people nearby. There are numerous existing papers that aim to guarantee safety. To list a few, the reachable sets method in [1] designs control trajectories while monitoring whether the reachable sets violate safety constraints. The barrier certificate method in [2] uses the dynamics and the certificate functions associated with the safety constraint inequality to ensure safety. The robust model predictive control (MPC) in [3], [4] employs min-max optimization to improve the robustness against disturbances. Another seminal safety verification method is the Control barrier function (CBF), which utilizes a Lyapunov-like function (i.e., the reciprocal CBF) to guarantee that the control output is forward invariant in a defined safe set [5]. Since the CBF can be augmented to a nominal control by solving a quadratic optimization program which can be implemented online, the CBF method is widely used in real-time robotic systems with extensions. For example, the authors in [6] combine the CBF with the Rapidly exploring Random Tree (RRT) algorithm to robustly satisfy the collision constraints in real-time. Also, the authors in [7] use *Backup Sets* to find admissible inputs and construct Measurement-Robust CBF that provides a margin to the robustness against noise.

There are two popular ways to augment the CBF into path planning problems: using gradient-based optimization and using sampling-based optimization. A gradient-based optimization programming problem for the nonlinear path planning problems becomes nonlinear optimization, inducing an optimal local solution with excessive computation time. This further results in low performance of the controlled systems and hinders real-time applications. The sampling-based methods can address the aforementioned drawbacks [8]. They usually do not require calculating the gradients that might be computationally expensive and impractical for complex systems and finish calculation in a designated time that only depends on sample size.

Model Predictive Path Integral Control (MPPI) algorithm in [9] is one of the sampling-based algorithms that generates a lot of forward-sampled trajectories to solve the stochastic optimal control problem. By sampling the forward trajectories of the dynamical system, the MPPI algorithm avoids calculating the derivative of the dynamic functions and cost functions [10]. Since the forward sampling of random trajectories can be calculated by parallel computing devices, the computation time of the MPPI algorithm is significantly lesser than other traditional methods [11]. Also, the sample size has a significant influence on the computation time and the performance of the sampling algorithm. It remains an open question how the CBF constraints will influence the sample size of the MPPI algorithms.

In this paper, we formulate a stochastic CBF-MPPI (SCBF-MPPI) algorithm that enhances safety with a probabilistic guarantee for a stochastic system in an obstacle-rich environment. Taking advantage of the MPPI and CBF, the proposed SCBF-MPPI algorithm has benefits in terms of safety and sampling efficiency compared to the standard MPPI. In particular, the proposed algorithm improves the sample efficiency by confining sampled trajectories in safe regions with changing the variance of the random perturbation. We formally analyze the sampling complexity to show how many sampled trajectories are required for the given stochastic optimization problem, and to show improved sampling efficiency compared to the standard MPPI. Furthermore, in the simulation with an obstacle-rich environment, we show that the proposed SCBF-MPPI algorithm has better performance in terms of collision avoidance than the MPPI when the same sample size is used for both algorithms. However, we note that the proposed algorithm induces a sub-optimal solution because the augmentation of the CBF trades

This research is supported by NSF CPS #1932529, NSF CMMI #1663460, UNR internal funding, and UIUC STII-21-06.

[†]Chuyuan Tao and Naira Hovakimyan are with the Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, USA. {chuyuan2, nhovakim}@illinois.edu

[*]Hyung-Jin Yoon and Petros Voulgaris are with the Department of Mechanical Engineering, University of Nevada, Reno, USA. {hyungjiny, pvoulgaris}@unr.edu

[‡]Hunmin Kim is with the Department of Electrical and Computer Engineering, Mercer University, USA. kim_h@mercer.edu

off the (infeasible) optimality with enhanced safety.

Related Work. The augmentation of the CBF in this work is inspired by the previous works on the CBF for fully known and deterministic systems [5]. Recent works demonstrate that it is augmentable for uncertain systems [12], [13], [14]. In [12], the authors unify the adaptive control Lyapunov function and adaptive control barrier function to guarantee safety in systems with parametric uncertainties. The paper [13] uses the piece-wise control update law to eliminate the effect of the disturbance. In [14], the authors introduce adaptive CBF, in the form of a penalty function, to ensure safety for uncertain systems.

Another line of work in safety ensures that control deals with stochastic differential equations instead of parametric uncertainties. In [15], the authors add chance constraints with the sampling-based MPC method for improving safety. In [16], the authors use $\mathcal{L}_1$ adaptive control augmentation with the MPPI to compensate for the gap between nominal and unknown dynamics. Also, in our previous work [17], we proposed a CBF-based MPPI algorithm that increases the sample efficiency and ensures safety in a nonlinear stochastic path planning problem.

II. Problem Statement

We consider a nonlinear control affine system:

$$dx_t = (f(x_t) + g(x_t)u_t)dt + \sigma(x_t)dW_t, \tag{1}$$

where $x_t \in \mathbb{R}^n$ is the state, $f : \mathbb{R}^n \times \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ and $\sigma : \mathbb{R}^n \to \mathbb{R}^n$ are locally Lipschitz continuous functions, and dW_t is a Wiener process with $\langle dW_k dW_l \rangle = \nu_{kl}(x_t, u_t, t)dt$. We also assume that the stochastic differential equation (1) has a strong solution for any control signal u_t. We consider an optimal control problem with a quadratic control cost and a state-dependent cost. The value function $V(x_t, t)$ is then defined as:

$$\min_u \mathbb{E}_{\mathbb{Q}} \left[\phi(x_T) + \int_t^T (q(x_t, t) + \frac{1}{2} u_t^T R(x_t, t) u_t) dt \right], \tag{2}$$

where $\phi(x_T)$ denotes a terminal cost, and $q(x_t, t)$ is a state-dependent cost. $R(x_t, t)$ is a positive definite matrix and needs to satisfy $\nu_{kl}(x_t, u_t, t) = \lambda g(x_t) R^{-1}(x_t, t) g^T(x_t)$, where λ is a constant. Let $\mathbb{E}_{\mathbb{Q}}$ and $\mathbb{E}_{\mathbb{P}}$ represent the expectations of the trajectories taken with the controlled dynamical system (1) and the uncontrolled dynamics of the system ($u \equiv 0$ for the dynamical system (1)). Let $\mathcal{C}$ represent a specified safe set, which is described by a locally Lipschitz function $h : \mathbb{R}^n \to \mathbb{R}$ as

$$\mathcal{C} = \{x : h(x) \geq 0\}, \qquad \partial\mathcal{C} = \{x : h(x) = 0\}.$$

Problem 1: Given the initial state $x_0 \in \mathcal{C}$, the problem is to design a control policy that solves the optimization problem defined in (2) subject to (1) while guaranteeing $x_t \in \mathcal{C}$ for $\forall t \geq 0$.

III. SCBF-Constrained MPPI Algorithm

A. Stochastic Control Barrier Function (SCBF)

The CBF algorithms use a Lyapunov-like constraint to guarantee safety, when the state approaches the boundary of the safe region $\mathcal{C}$. However, most CBF algorithms are based on a formulation that uses a deterministic system. Yet, the stochastic formulation has a different treatment of uncertainty. To resolve the gap between the stochastic and deterministic systems, we use the SCBF, first proposed in [18], using the Itô derivative instead of the Lie derivative to guarantee safety. The use of the Itô derivative adds additional terms in the following definition of SCBF compared to the deterministic one.

Definition 1: [18] The function $h : \mathbb{R}^n \to \mathbb{R}$ is a SCBF for system (1), if for all x satisfying $h(x) > 0$, there exists u satisfying

$$L_f h(x) + L_g h(x) u + \frac{1}{2} \mathrm{Tr}(\sigma^T \frac{\partial^2 h}{\partial x^2} \sigma) \geq -h(x). \tag{3}$$

Definition 1 extends the definition of the CBF to the stochastic system and guarantees safety, as shown in the following theorem.

Theorem 1: [18] If u satisfies (3) for all time t, then $Pr(x_t \in \mathcal{C} \forall t) = 1$, provided $x_0 \in \mathcal{C}$.

The CBF algorithm defines a quadratic programming problem (QP) that minimizes the difference between the safe control output u_s and the nominal control input u_n and simultaneously satisfies the CBF constraints. We can define the following convex QP problem for the stochastic safe control design:

$$\begin{aligned} &\arg\min_{u_s} \quad \frac{1}{2} \left\| u_s - u_n \right\|_2^2, \\ &\text{s.t.} \quad L_f h(x) + L_g h(x) u_s + \frac{1}{2} \mathrm{Tr}(\sigma^T \frac{\partial^2 h}{\partial x^2} \sigma) \geq h(x). \end{aligned}$$

The SCBF algorithm also averts calculating the derivatives of the dynamic system; the convex QP optimization problems in the algorithm can be solved in polynomial time [19]. However, the CBF algorithm can only guarantee the safety of the dynamics. To solve the stochastic path planning problem defined in Problem 1, we will use the MPPI algorithm [10], which also evades the calculation of the derivatives of the dynamic functions.

B. Model Predictive Path Integral Control (MPPI)

To solve the nonlinear stochastic optimization described in problem 1, we apply the MPPI algorithm using Monte Carlo (MC) methods to approximate the optimal control solution.

First, the MPPI algorithm samples K trajectories with T being the time horizon. In each trajectory τ_i, let $v_i = [v_{i,0}, \ldots, v_{i,T-1}]^T$ be the mean of the control sequence. Let $u_i = [u_{i,0}, \ldots, u_{i,T-1}]^T$ be the actual control input sequence. $\epsilon_i = [\epsilon_{i,0}, \ldots, \epsilon_{i,T-1}]^T$ represents the disturbance of the control input, and $\epsilon_{i,t} \sim \mathcal{N}(0, \Sigma_{i,t})$. $[x_{i,0}, \ldots, x_{i,T-1}]^T$ denotes the states of the current sampled trajectory.

The iterative update law is:

$$u(x_t, t)^* = u(x_t, t) + \frac{\mathbb{E}_{\mathbb{Q}}[\exp(-(1/\lambda) S(\tau) \sigma(x_t) dW(t)]}{\mathbb{E}_{\mathbb{Q}}[\exp(-(1/\lambda) S(\tau)]},$$

where $u(x_t, t)$ is the initial control input to be optimized, and $S(\tau) = \phi(x_T) + \int_{t_0}^{T} q(x_t, t)dt$.

The continuous-time trajectories are sampled as a *discretized* system $x_{t+1} = x_t + dx_t$ according to

$$\begin{aligned} dx_t &= (f(x_t, t) + g(x_t, t)\mathbf{u}(x_t, t))\,\Delta t + \sigma(x_t, t)\delta_t\sqrt{\Delta t}, \\ &= \tilde{f}(x_t, t)\Delta t + \sigma(x_t, t)\delta_t\sqrt{\Delta t}, \end{aligned}$$

where δ_t is the Gaussian random vector with independent and identically distributed (i.i.d.) standard normal Gaussian random variables, i.e., $[\delta_t]_i \sim \mathcal{N}(0, 1)$, and Δt denotes the time step of the time-discretization using Euler–Maruyama method [20]. Then the discrete-time control update law to approximate the optimal control will be:

$$u(x_{t_i}, t_i)^* \approx u(x_{t_i}, t_i) + \frac{\sum_{i=0}^{K-1} \exp(-(1/\lambda)\tilde{S}(\tau_{i,t}))\delta u_{i,t}}{\sum_{i=0}^{K-1} \exp(-(1/\lambda)\tilde{S}(\tau_{i,t}))}, \tag{4}$$

where $\delta u_{i,t} = \frac{\delta t}{\sqrt{\Delta t}}$ can be considered as a random control input, and $\tilde{S}(\tau_{i,t}) = \phi(x_{i,t}) + \sum_{k=0}^{T-1} \tilde{q}(x_{i,t}, v_{i,t}, \epsilon_{i,t})$. The running cost function is $\tilde{q}(x_{i,t}, v_{i,t}, \epsilon_{i,t}) = q(x_{i,t}) + \frac{1-\nu \;{}^{1}}{2}\epsilon_{i,t}^T R\epsilon_{i,t} + v_{i,t}^T R\epsilon_{i,t} + \frac{1}{2}v_{i,t}^T Rv_{i,t}$, where ν is the ratio between the covariance of the injected disturbance ϵ_i and the covariance of the disturbance of the original dynamics.

The MPPI algorithm solves the nonlinear stochastic problems efficiently based on the dynamic trajectories sampled from normal distributions. However, most sampled trajectories may violate the safety constraints and receive a penalty in the reward functions in some extreme environments, such as the obstacle-rich environment. As a result, the sample efficiency of the algorithms becomes poor and eventually influences the performance of the algorithms. To solve these problems, we provide a safe and sampling-efficient algorithm by using SCBF to constrain the sample distributions.

C. SCBF-Based Chance Constrained MPPI Algorithm

We consider the SCBF-based chance cosntrained algorithm, where SCBF constraints are no longer used as a safe filter for the control input. In our previous work [17], we mentioned that if the CBF method compensates for the control output of the MPPI algorithm directly, the exploration of the MPPI algorithm will be hindered. So we design CBF chance constraints to find a trust region for the algorithms to sample. We formulate a convex optimization based on the trust region of the SCBF functions as follows:

$$\begin{aligned} &\arg\min_{u} \|u - u_n\|, \\ &\text{s.t.}\Pr\left(L_g h(x)u \geq -h(x) - L_f h(x) - \frac{1}{2}\operatorname{Tr}(\sigma^T H_x \sigma)\right) \\ &\geq 1 - \delta \end{aligned} \tag{5}$$

where u_n is a nominal random control input distribution, $H_x = \frac{\partial^2 h}{\partial x_{i,t}{}^2}$ is the hessian matrix of function h.

Theorem 2: Let the Gaussian distributions $\mathbb{Q}_0 = \mathcal{N}(\mu_0, \Sigma_0)$ and $\mathbb{Q}_s = \mathcal{N}(\mu_s, \Sigma_s)$ satisfy $\Sigma_0 > \Sigma_s$. For fixed probability $1 - \delta$ defined in (5), if the mean μ and variance Σ satisfy the constraint:

$$A_{i,t}\mu - \alpha A_{i,t}\Sigma A_{i,t}^T \geq b_{i,t}, \tag{6}$$

where α is the confidence interval corresponding to the probability $1 - \delta$, $A_{i,t} = L_g h(x_{i,t})$, and $b_{i,t} = -h(x_{i,t}) - L_f h(x_{i,t}) - \frac{1}{2}\operatorname{Tr}(\sigma^T \frac{\partial^2 h}{\partial x_t^{i\,2}}\sigma)$, then the sample control input $u_{i,t} \sim \mathbb{Q}(\mu_s, \Sigma_s)$ satisfies the SCBF chance constraints in the optimization problem (5).

Proof: The left hand side of the inequality can be cast as a Gaussian random variable with mean $A_{i,t}\mu$ and variance $A_{i,t}\Sigma A_{i,t}$. Considering the upper bound of the confidence interval for the Gaussian variable, we can simplify the SCBF chance constraint to a linear combination of mean $A_{i,t}\mu$ and variance $A_{i,t}\Sigma A_{i,t}$, which satisfy the inequality $A_{i,t}\mu \geq \alpha A_{i,t}\Sigma A_{i,t}^T + b_{i,t}$. ■

Since the variance of the distribution should always be positive semidefinite $\Sigma \succeq 0$, to guarantee the convexity of the previous optimization problem (5) we reformulate it to:

$$\begin{aligned} &\arg\min_{\mu,\Sigma} \|\mu - \mu_0\|_1 + \|\Sigma - \Sigma_0\|_p, \\ &\text{s.t.} \quad A_{i,t}\mu - \alpha A_{i,t}\Sigma A_{i,t}^T \geq b_{i,t}, \\ &\qquad\; \Sigma \succeq 0. \end{aligned} \tag{7}$$

Remark 1: Suppose that there exists a state x such that the SCBF chance constraint is active. Then the following inequality holds $\alpha A_{i,t}\Sigma A_{i,t}^T \leq A_{i,t}\mu - b_{i,t}$. This inequality shows that there is an upper bound on the variance Σ. Hence, we can assume that $\Sigma_0 \succeq \Sigma$.

For any positive semidefinite matrices Σ_0, Σ, there exist matrices P_0, P such that $\Sigma_0 = P_0 P_0^T$ and $\Sigma = PP^T$. Using this fact, we can simplify the constraints in (7), and the optimization problem becomes:

$$\begin{aligned} &\arg\min_{\mu,P} \|\mu - \mu_0\|_1 + \|P - P_0\|_p, \\ &\text{s.t.} \begin{pmatrix} I & \sqrt{\alpha}A_{i,t}P \\ \sqrt{\alpha}P^T A_{i,t}^T & \mu A_{i,t} - b_{i,t} \end{pmatrix} \succeq 0. \end{aligned} \tag{8}$$

The solution to the optimization problem (8) provides the safe mean μ_s and variance $\Sigma_s = PP^T$. In our previous work [17], we proved that the optimization can be simplified to a Semidefinite Programming optimization problem (SDP). In [21], it is shown that using parallel computing, the SDP problems can be solved almost as efficiently as linear programming. Based on the safe mean and variance from SDP optimization (8), we will generate one safe control variation $\delta u_{i,t} \sim \mathcal{N}(\mu_s, \Sigma_s)$ for each sample state $x_{i,t}$ and propagate through discrete dynamics:

$$x_t^k = x_{t-1}^k + \left(f(x_t^k) + g(x_t^k)(u_{i,t} + \delta u_{i,t}\right). \tag{9}$$

We can obtain a sampled trajectory $\tau_i = \{x_0, ..., x_T\}$, where T is the time horizon of the MPPI algorithm. Then we will calculate the cost of the i^{th} sampled trajectory by using the cost function $\tilde{S}(\cdot)$, and using the following equation we will calculate the weight of each trajectory:

$$\omega_i = \exp\left(-\frac{1}{\lambda}\tilde{S}(\tau_i)\right). \tag{10}$$

We use the following control update law:

$$u^*(x_{t_i}, t_i) \approx u(x_{t_i}, t_i) + \frac{\sum_{i=1}^{K} \omega_i \delta u_{i,t}}{\sum_{k=1}^{K} \omega_i}. \tag{11}$$

In conclusion, we have the following algorithm:

Algorithm 1 SCBF-MPPI algorithm

```
$u_{init}$ : Random initialize control input;
$K, T$ : Size of sampled trajectories and timesteps;
$\phi, \tilde{q}, R, \lambda$ : Cost function parameters;
while task is not completed do
    for $i \leftarrow 0$ to $K-1$ do
        $x \leftarrow x_0$;
        for $t \leftarrow 1$ to $T$ do
            Solving SDP in (8) to get $\mu_s, \Sigma_s$;
            Generate control variations $\delta u_{i,t} \sim \mathcal{N}(\mu_s, \Sigma_s)$;
            Simulate discrete dynamic (9) to obtain $x_{i,t}$;
            Calculate cost function $S(\tau_i) \mathrel{+}= \tilde{q}(x_{i,t}, \delta u_{i,t})$;
        end for
        Calculate the terminal cost $S(\tau_i) \mathrel{+}= \phi(x_{i,T})$
    end for
    $\beta \leftarrow \min_i[S(\tau_i)]$;
    Get sample weights $\omega_{i,t}$ using (10);
    Update control input using $\omega_{i,t}$ and $\delta u_{i,t}$ using (11);
    Send $u_{t_0}$ to actuator;
    for $i \leftarrow 0$ to $T-2$ do
        $u_i = u_{i+1}$;
    end for
    $u_{N-1} = u_{init}$;
end while
```

IV. Sample Size Analysis

The size of sampled trajectories has a significant effect on the performance and the computation time of the sampling-based algorithms. With more sampled trajectories, the performance of the algorithm will be better, but the computation time of the algorithm will increase significantly as well. Our previous work on the sampling complexity of the PI method [22] uses Hoeffding's inequality and Chebyshev's inequality. We will provide a sampling complexity analysis of this intuitive idea. We discuss the case of one dimensional control input $[\delta u_{i,t}]_i \sim \mathcal{N}([\mu_t]_i, [\Sigma_t]_i)$ in this paper, and a similar result can be extended to high dimensional control input straightforwardly.

Assumption 1: Assume that the error bound ϵ_1 of the Chebyshev's inequality is smaller than the expectation of ω

$$\epsilon_1 < \mathbb{E}[\exp(-\frac{1}{\lambda}S(\tau_i)].$$

Assumption 2: We suppose that the running cost function $\tilde{q}(x_{i,t})$ and terminal cost function $\phi(x_{i,T})$ are quadratic functions.

Theorem 3: Under Assumptions 1 and 2, the size of sampled trajectory N of the original MPPI control update law defined in (4) is larger than the size of sampled trajectory N^s of the control update law of the SCBF-MPPI algorithm (11) given the same sampling complexity error bound ϵ and risk probability ρ.

To prove the above theorem we have the following lemmas and propositions.

Lemma 1: For any random variables X, Y, we have: $\text{Var}\,[XY] \leq 2\text{Var}\,[X]\text{Var}\,[Y] + 2\text{Var}\,[Y]\,\mathbb{E}[X]^2$.

Due to the space of the limitation, we defer the details of the proofs in the paper to online full version [23].

Lemma 2: We have $\text{Var}\,[\omega] \leq (1 - \mathbb{E}[\omega])\,\mathbb{E}[\omega] \leq \mathbb{E}[\omega] \leq 1$.

Proof: Since $\omega = \exp(-\frac{S(\tau)}{\lambda})$ and since the cost-to-go function $S(\tau) \geq 0$ by Assumption 2, then $\omega \in [0, 1]$ is a bounded random variable and its variance is also bounded. ■

Corollary 1: We have

$$\mathbb{P}\{|\hat{E}_1 - \mathbb{E}[\omega]| \geq \epsilon_1\} \leq \rho_1 := 2\exp(-N_1\epsilon_1^2), \tag{12}$$

where $\hat{E}_1 = \sum_{i=1}^{T} \exp(-\frac{1}{\lambda}S(\tau_i))$, ϵ_1 is the error bound of the discrete sampled estimate, and ρ_1 is referred to as risk probability of not satisfying the error bound. The sample size N_1 can be calculated as:

$$N_1 = -\frac{1}{\epsilon_1^2}\log\frac{\rho_1}{2}. \tag{13}$$

Proof: By Hoeffding's inequality and with $\omega \in [0, 1]$:

$$\begin{aligned}\mathbb{P}\{|\hat{E}_1 - \mathbb{E}[\omega]| \geq \epsilon_1\} &\leq 2\exp\left(-\frac{N_1\epsilon_1^2}{(\omega_{\max} - \omega_{\min})^2}\right)\\ &\leq 2\exp(-N_1\epsilon_1^2).\end{aligned}$$

■

Lemma 3: Let $\hat{E}_2$ denote the resample control update $\hat{E}_2 := \frac{1}{N}\sum_i^N(\frac{\omega_i\delta u_i}{\mathbb{E}[\omega]})$ and $E_2 = \mathbb{E}\left[\frac{\omega\delta u}{\mathbb{E}[\omega]}\right]$. We have the following error bound ϵ_2 and sample size N_2:

$$\mathbb{P}\left\{\left|\left[\hat{E}_2 - E_2\right]_i\right| \geq \epsilon_2\right\} \leq \rho_2 := \frac{\Gamma}{N_2\epsilon_2^2}\left(\exp\left(\frac{2\mathbb{E}[S(\tau)]}{\lambda}\right)\right), \tag{14}$$

where $\Gamma = 4\text{Var}\,[\delta u]$.

Due to the space of the limitation, we defer the details of the proofs in the paper to online full version [23].

Corollary 2: Under Assumptions 1 and 2, the MC error bound in (14) becomes

$$\mathbb{P}\left\{\left|\left[\hat{E}_2 - E_2\right]_i\right| \geq \epsilon_2\right\} \leq \frac{\Gamma}{N_2\epsilon_2^2}\left(\frac{1}{\hat{E}_1 - \epsilon_1}\right)^2,$$

where ϵ_1 is the first MC error bound from (12). Then we conclude that for the error bound ϵ_2 and the risk probability ρ_2, the sample size N_2 can be calculated:

$$N_2 = \frac{4\text{Var}\,[\delta u]}{\rho_2\epsilon_2^2}\left(\frac{1}{\hat{E}_1 - \epsilon_1}\right)^2. \tag{15}$$

Proof: From the proof in Lemma 3, we have

$$\text{Var}\left[\frac{\omega\delta u_t}{\mathbb{E}[\omega]}\right] \leq \frac{4\text{Var}\,[\delta u]}{\mathbb{E}[\omega]^2}.$$

Using the inequality $|\hat{E}_1 - \mathbb{E}[\omega]| \geq \epsilon_1$, we have the following relation for $(\mathbb{E}[\omega])^2$:

$$\frac{1}{(\hat{E}_1 + \epsilon_1)^2} \leq \frac{1}{(\mathbb{E}[\omega])^2} \leq \frac{1}{(\hat{E}_1 - \epsilon_1)^2}.$$

Then the above follows using Chebyshev's inequality. ■

Since the control output distribution of the MPPI algorithm δu has greater variance than the safe control output distribution of the SCBF-MPPI algorithm δu_s, by using the conclusion of Corollary 2, we can conclude that to reach the same error bound ϵ_2 and risk probability ρ_2, the sample size of the MPPI algorithm N_2 is larger than the size of SCBF-MPPI algorithm N_2^s. For the same error bound ϵ_1 and risk probability ρ_1, the sample size of both algorithms are the same based on the Corollary 1. So the required samples size for the MPPI algorithm $N = \max(N_1, N_2)$ is also greater or equal to the size of samples for the SCBF-MPPI algorithm $N^s = \max(N_1, N_2^s)$.

V. Simulations

A. Unicycle Dynamics

We implement our algorithms on a two-dimensional unicycle dynamical system with:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix} + \sigma dW(t),$$

where x, y are the coordinates, θ is the angle, v is the linear velocity, ω is the angular velocity of the unicycle model, σ is the identity matrix, and $W(t)$ is a Brownian motion. The time step for the discrete-time simulation is $\Delta t = 0.05s$.

B. Simulation Setup

We demand a probability of $1 - \gamma = 0.997$ of avoiding all obstacles. The parameters for the sampling algorithm are set: the time horizon $T = 20$, and $\lambda = 1$.

We consider a narrow passage environment where the safe set $\mathcal{C}$ is defined as $\left\{(x, y) | \sin\frac{\pi}{2}x < y < \sin\frac{\pi}{2}x + \alpha\right\}$, where α is the width of the narrow passage. The control barrier functions are $\{h_1 := y - \sin x > 0\}$, and $\{h_2 := \sin x + \alpha - y > 0\}$. We consider a stochastic path planning problem where the initial state of the unicycle is $[0, 0.5, 0]^T$, and the target state is $[4, 0.5, 0]^T$. The running cost function is $q(x, y, \theta) = \|X - X_g\|_2^2 + 1000 * \mathbb{1}_{X\in\bar{\mathcal{C}}}$, where $\bar{\mathcal{C}}$ is the complementary set to the safe set $\mathcal{C}$ over $\mathbb{R}^2$, and $\mathbb{1}$ is the indicator function.

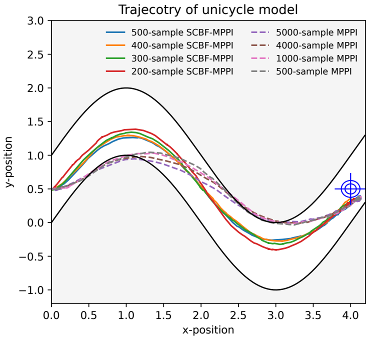


Fig. 1: Path planning result with different sample sizes.

C. Results

Figure 1 illustrates a result for the unicycle robots navigating through a narrow passage for at most 12.5 seconds. The black line represents the boundary of the safe set. The blue cross represents the target position. We first implemented the MPPI algorithm with 500, 1000, 4000 and 5000 sampled trajectories. The robotic systems can reach the target position but may violate safety in some states. Then, we implement the SCBF-MPPI algorithm with 200, 300, 400, and 500 sampled trajectories where the safety is guaranteed and also reaches the target successfully. However, the SCBF-MPPI algorithm behaves more conservatively and spends more time steps for reaching the goal. In Figure 2, we plot different algorithms' running cost function values with varying sample sizes. The SCBF-MPPI algorithm converges slower than the MPPI algorithm, which illustrates the same result in Figure 1: the SCBF-MPPI algorithm is more conservative. The CBF constrained sample distribution leads to the conservativeness of the control output.

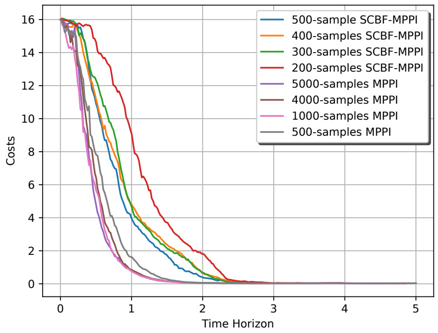
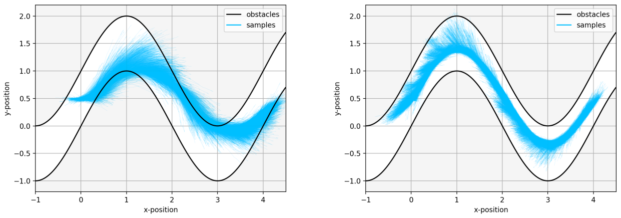


Fig. 2: Cost with different sample sizes.

We define the collision rate as how many states in the trajectory violate the safety constraint. We further define the average time to finish (TTF) to be the number of time steps to reach a vicinity area of the target. In this experiment, we define the vicinity area as a circle with a radius of 0.15. We repeat both algorithms ten times to calculate the average collision rate and average time steps. Table I shows that in both the MPPI and the SCBF-MPPI algorithms, the average TTF will decrease when the sample size is larger, which means the performance will be better. The MPPI algorithm has a lower average TTF than the SCBF-MPPI, which indicates that SCBF-MPPI is more conservative. However, the collision rate of SCBF-MPPI remains 0, which implies that the safety constraints are always satisfied during the experiments.

In Figure 3a and Figure 3b, we plot all the sampled trajectories of the MPPI algorithm and the SCBF-MPPI algorithm when the sample size is 200. The sampled trajectories are in blue color, and the boundary of the safe set is in black color. The sampled trajectories in the MPPI algorithm follow a Gaussian random variable with fixed mean and variance. For the SCBF-MPPI algorithm, the mean and variance of the sampled trajectory are obtained based on



(a) MPPI algorithm.

(b) SCBF-MPPI algorithm.

Fig. 3: The sampled trajectories of the algorithm

TABLE I: MPPI and CBF-MPPI of different sample size

Case	Collision Rate	Avg. TTF
200 samples MPPI	0.0454	140.4
500 samples MPPI	0.0294	129.6
200 samples SCBF-MPPI	0.0	163.6
500 samples SCBF-MPPI	0.0	156.1

the SDP optimization. As a result, all sampled trajectories are in the safe set. Note that some aggressive and dangerous sampled trajectories in the SCBF-MPPI algorithm will be turned opposite because of the confidence interval we set. But the costs of these samples are low, so they have minor effect on the control output.

In our last experiment, we set the sample size to be $N = 500$, the desired bound $\epsilon_1 = 0.05, \epsilon_2 = 0.1,$, the allowable risk of failure $\rho_1 = 0.05, \rho_2 = 0.1$. We calculate the sample size N_1 and N_2 based on the equations (13) and (15) at time step $T = 50$. In conclusion, it shows that our algorithm needs smaller sample size than the MPPI algorithm.

TABLE II: Sample size for N_1 and N_2

Algorithm	T	N_1	N_2
MPPI	50	1476	2973
SCBF-MPPI	50	1476	584

VI. Conclusion and Discussion

We propose a SCBF-MPPI algorithm that utilizes the safe SCBF constraints to determine the mean and variance of the random control trajectories for calculating the path-integral control. The augmentation will defer the control from the (potentially infeasible) optimal solution provided by the MPPI. The improved safety can also indirectly be observed through the smaller sample size required by the SCBF-MPPI than the MPPI for the same level of assurance. Using sampling complexity analysis and simulations, we show that the CBF-MPPI algorithm needs fewer sampled trajectories due to the smaller variance of the control distribution.

References

[1] A. Girard, C. L. Guernic, and O. Maler, "Efficient computation of reachable sets of linear time-invariant systems with inputs," in *International workshop on hybrid systems: Computation and control*, pp. 257–271, Springer, 2006.

[2] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *International Workshop on Hybrid Systems: Computation and Control*, pp. 477–492, Springer, 2004.

[3] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*, pp. 207–226, Springer, 1999.

[4] J. Lofberg, "Approximations of closed-loop minimax mpc," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 2, pp. 1438–1442, IEEE, 2003.

[5] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*, pp. 3420–3431, IEEE, 2019.

[6] K. Majd, S. Yaghoubi, T. Yamaguchi, B. Hoxha, D. Prokhorov, and G. Fainekos, "Safe navigation in human occupied environments using sampling and control barrier functions," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5794–5800, IEEE, 2021.

[7] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-robust control barrier functions: Certainty in safety with uncertainty in state," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6286–6291, IEEE, 2021.

[8] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *The Journal of Machine Learning Research*, vol. 11, pp. 3137–3181, 2010.

[9] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1433–1440, IEEE, 2016.

[10] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1714–1721, IEEE, 2017.

[11] 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.

[12] A. J. Taylor and A. D. Ames, "Adaptive safety with control barrier functions," in *2020 American Control Conference (ACC)*, pp. 1399–1405, IEEE, 2020.

[13] P. Zhao, Y. Mao, C. Tao, N. Hovakimyan, and X. Wang, "Adaptive robust quadratic programs using control lyapunov and barrier functions," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3353–3358, IEEE, 2020.

[14] W. Xiao, C. Belta, and C. G. Cassandras, "Adaptive control barrier functions," *IEEE Transactions on Automatic Control*, 2021.

[15] M. Koch, M. Spies, and M. Bürger, "Trust regions for safe sampling-based model predictive control," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9313–9319, IEEE, 2019.

[16] J. Pravitra, K. A. Ackerman, C. Cao, N. Hovakimyan, and E. A. Theodorou, "L1-adaptive mppi architecture for robust and agile control of multirotors," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7661–7666, IEEE, 2020.

[17] C. Tao, H. Kim, H. Yoon, N. Hovakimyan, and P. Voulgaris, "Control barrier function augmentation in sampling-based control algorithm for sample efficiency," *arXiv preprint arXiv:2111.06974*, 2021.

[18] A. Clark, "Control barrier functions for stochastic systems," *Automatica*, vol. 130, p. 109688, 2021.

[19] M. K. Kozlov, S. P. Tarasov, and L. G. Khachiyan, "Polynomial solvability of convex quadratic programming," in *Doklady Akademii Nauk*, vol. 248, pp. 1049–1051, Russian Academy of Sciences, 1979.

[20] E. Platen and N. Bruti-Liberati, *Numerical solution of stochastic differential equations with jumps in finance*, vol. 64. Springer Science & Business Media, 2010.

[21] R. Peng and K. Tangwongsan, "Faster and simpler width-independent parallel algorithms for positive semidefinite programming," in *Proceedings of the twenty-fourth annual ACM symposium on Parallelism in algorithms and architectures*, pp. 101–108, 2012.

[22] H.-J. Yoon, C. Tao, H. Kim, N. Hovakimyan, and P. Voulgaris, "Sampling complexity of path integral methods for trajectory optimization," *arXiv preprint arXiv:2203.10067*, 2022.

[23] C. Tao, H.-J. Yoon, H. Kim, N. Hovakimyan, and P. Voulgaris, "Path integral methods with stochastic control barrier functions," *arXiv preprint arXiv:2206.11985*, 2022.