

Uncertainty Guided Exploratory Trajectory Optimization for Sampling-Based Model Predictive Control

O. Goktug Poyrazoglu¹, Yukang Cao¹, Rahul Moorthy¹ and Volkan Isler²

Abstract—Trajectory optimization depends heavily on initialization. In particular, sampling-based approaches are highly sensitive to initial solutions, and limited exploration frequently leads them to converge to local minima in complex environments. We present Uncertainty Guided Exploratory Trajectory Optimization (UGE-TO), a trajectory optimization algorithm that generates well-separated samples to achieve a better coverage of the configuration space. UGE-TO represents trajectories as probability distributions induced by uncertainty ellipsoids. Unlike sampling-based approaches that explore only in the action space, this representation captures the effects of both system dynamics and action selection. By incorporating the impact of dynamics, in addition to the action space, into our distributions, our method enhances trajectory diversity by enforcing distributional separation via the Hellinger distance between them. It enables a systematic exploration of the configuration space and improves robustness against local minima. Further, we present UGE-MPC, which integrates UGE-TO into sampling-based model predictive controller methods. Experiments demonstrate that UGE-MPC achieves higher exploration and faster convergence in trajectory optimization compared to baselines under the same sampling budget, achieving 72.1% faster convergence in obstacle-free environments and 66% faster convergence with a 6.7% higher success rate in the cluttered environment compared to the best-performing baseline. Additionally, we validate the approach through a range of simulation scenarios and real-world experiments. Our results indicate that UGE-MPC has higher success rates and faster convergence, especially in environments that demand significant deviations from nominal trajectories to avoid failures. The project and code are available at https://ogpoyrazoglu.github.io/cuniform_sampling/.

I. INTRODUCTION

Trajectory optimization is a powerful concept in robotics for generating efficient robot motions under task-specific objectives. It has been successfully applied to applications including navigation [1], manipulation [2], and aerial robotics [3]. Classical gradient-based methods focus on smooth settings, where differentiability can be exploited for efficient optimization, and have demonstrated strong performance in such settings [4]- [5]. However, their reliance on differentiability limits their direct use in complex multimodal environments. In contrast, sampling-based approaches relax these assumptions by allowing non-differentiable cost functions, making them applicable to a broader range of robotic systems and tasks. They are most commonly applied in Model Predictive Control (MPC) settings since they can handle nonlinear dynamics with arbitrary cost functions [6].

The authors are with the Robotics, Sensing and Networks Laboratory (RSN). ¹University of Minnesota, Minneapolis, MN 55455, USA. ²The University of Texas at Austin, Austin, TX 78712, USA. Correspondence: poyra002@umn.edu.

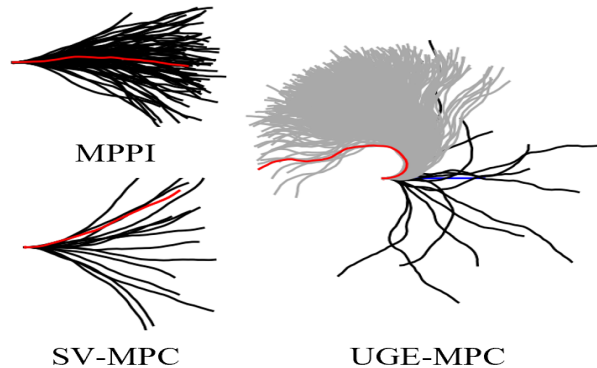


Fig. 1: Comparison of first-iteration trajectories when the robot starts heading +x, and the nominal trajectory is forward, with the goal located behind the robot. Under the same sample budget of 2048. Red trajectories indicate the solutions obtained in this iteration for each method. Black trajectories denote the sampled particles for SV-MPC and UGE-MPC, and the rollouts for MPPI. Leveraging distributional separation in trajectory space, UGE-MPC produces backward-reaching candidates and a feasible solution in one iteration. SV-MPC enhances diversity through repulsion but does not consistently cover the required full-turn maneuvers, whereas MPPI remains concentrated around the nominal trajectory.

Despite these advantages, sampling-based methods remain highly sensitive to the selection of the underlying sampling distribution and the initialization. For example, Model Predictive Path Integral (MPPI) [6] samples action sequences from a Gaussian distribution centered on the nominal trajectory, resulting in trajectories that are concentrated around it. As a result, when the distribution lacks sufficient diversity or the nominal trajectory is poorly initialized, the resulting trajectories may fail to cover promising areas of the configuration space. This often leads to convergence to local minima and reduces the controller’s reliability in complex environments. Addressing this limitation is crucial to enhancing the robustness of sampling-based approaches.

Several methods have been proposed in the literature to enhance exploration, thereby improving the reliability of the methods. Early approaches directly modify the sampling distribution to encourage broader exploration. For example, Log-MPPI [7], which increases trajectory diversity by reshaping the underlying distribution. Another line of work is to incorporate entropy-based regularization terms into the objective to promote diversity in the resulting policies [8],

[9], thereby penalizing narrow policy distributions. More recently, multimodal approaches have been presented to capture a distribution over trajectories rather than a single solution. A class of approaches [10], [11] uses Stein Variational Gradient Descent (SVGD) [12], which uses particles to approximate multimodal trajectory distributions by the repulsive feature of SVGD to maintain the diversity among trajectories, while shifting them toward low-cost regions. While these methods improve exploration, they often rely solely on action-space perturbation and may not fully capture the impact of system dynamics on trajectory diversity.

In this work, we present a new exploration technique that extends the idea of exploration via action perturbations by accounting for the impact of long-range system dynamics. Specifically, we introduce Uncertainty Guided Exploratory Trajectory Optimization (UGE-TO) that constructs probability distributions for trajectories by using uncertainty ellipsoids around trajectories. After the trajectory distributions are constructed, we enforce distributional separation using the Hellinger Distance to reduce overlap between the distributions. In other words, it will lead to exploration of the configuration space by forcing trajectories to visit distinct regions of the configuration space. We further present UGE-MPC, which integrates UGE-TO into a sampling-based MPC framework. In this setting, we use UGE-TO to generate diverse trajectory candidates and identify a better nominal trajectory, which helps the controller to perform a local search around low-cost regions.

In summary, our contributions of this work are:

- We introduce Uncertainty Guided Exploratory Trajectory Optimization (UGE-TO), which utilizes the distribution separation of trajectory probability distributions generated by uncertainty ellipsoids to effectively explore the configuration space.
- We present UGE-MPC, a new sampling-based model predictive controller that utilizes UGE-TO to enhance exploration of representative trajectory rollouts and then performs a local search around the selected minimum cost one. By achieving high exploration of the configuration space, our method reduces the likelihood of becoming trapped in local minima.
- We evaluate our approach in simulation and real-world scenarios, demonstrating improved exploration, faster convergence, and higher success rates compared to baselines, particularly in environments requiring significant deviations from nominal trajectories.

Integrating UGE-TO into a sampling-based MPC (UGE-MPC) provides a systematic approach that augments exploration with local search to refine exploitation. We begin with an overview of the related work.

II. RELATED WORK

A. Trajectory Optimization

Existing approaches to local trajectory optimization can be divided into two branches: direct and indirect methods [13]. One approach is to use direct methods that translate the

optimization problem into a nonlinear problem [4], [5], and solve it with nonlinear solvers, such as IPOPT [14] and SNOPT [15]. On the other side, indirect methods aim to convert the problem into subproblems and solve for the local optimality conditions. Differential dynamic programming (DDP) [16] and iterative Linear Quadratic Regulator (iLQR) [17] are well-known examples of this method. A major limitation of both methods is their strong dependence on initialization. Since they seek to find a single local-optimal solution, this property increases the importance of the initialization. Later work combines maximum entropy policies with DDP and avoids local minima via exploration capacity of their multimodal policy [18], [19]. Similarly, CSVTO [20] uses Stein Variational Gradient Descent (SVGD) [12] to generate a diverse set of trajectories to avoid possible local minima. Our work also aligns with this line of research by aiming to overcome the limitations of a single local solution. However, in contrast to these methods that rely on convexity or differentiability of the cost functions, we focus on sampling-based approaches that naturally handle non-convex and non-differentiable objectives. This allows us to enhance diversity and robustness in trajectory optimization without depending on restrictive assumptions.

B. Sampling-Based Trajectory Optimization

Sampling-based approaches have gained popularity and become state-of-the-art for handling non-smooth cost landscapes by directly evaluating candidate control sequences drawn from their distributions [21]. Early methods, such as the Cross-Entropy Method (CEM) [22] and Model Predictive Path Integral control (MPPI) [6], demonstrated strong performance across various robotic systems by iteratively adapting their action-sampling distributions based on rollout costs. However, similar to gradient-based methods, these approaches also struggle with local minima in the cost landscape due to their limited exploration in the configuration space. Recent works have sought to address these issues by improving sampling strategies. One line of research focused on directly modifying the sampling distribution. Log-MPPI [7] leverages the Normal-Log-Normal (NLN) distribution to increase the sample diversity. Similarly, colored-noise-based variants use correlated noises to generate both smoother and diverse trajectories [23], [24]. Although these methods have increased the exploration capabilities, they still rely on unimodal sampling strategies that lead to a single solution.

On the other hand, another class of methods introduces multimodal sampling distributions. In the SVGD-based approach, SV-MPC [10] uses particle approximations of trajectories to represent a multimodal trajectory distribution with multiple particles. SV-MPC's MPPI extension, SVG-MPPI [25], guides trajectories to low-cost regions by modifying both the nominal trajectory and the covariances. Similarly, C-Uniform trajectory sampling [26] has been presented to maintain uniform coverage of configurations over level sets, and that leads to high exploration over the configuration space. Its extension, CU-MPPI [27], incorporates C-Uniform

trajectories for better nominal trajectory selection into the MPPI framework to avoid local minima. In contrast, UGE-MPC applies Hellinger distance-based separation directly in the space of trajectory distributions, ensuring diversity under both actions and dynamics. This approach avoids the scalability limitations of entropy-based grid methods, such as CU-MPPI, while providing more principled configuration-space clustering than action-level kernels, like SV-MPC. Another method closely related to our work is U-MPPI [28], which utilizes the unscented transform to enhance state-space exploration. Compared with U-MPPI, our method also uses uncertainty ellipsoids, but unlike U-MPPI, we explicitly represent trajectories as probability distributions and separate them distributionally using the Hellinger Distance-based metric. This property explicitly prevents overlap between trajectory distributions, which promotes exploration and reduces the likelihood of collapsing into a local minimum.

III. PROBLEM FORMULATION

We begin by formulating trajectory optimization as a general stochastic optimal control problem, where $\mathbf{x}_t \in \mathbb{R}^n$ denotes the system state and $\mathbf{u}_t \in \mathbb{R}^m$ denotes the control input at discrete time step t . The system evolves according to the following stochastic dynamics:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \quad (1)$$

where f represents nonlinear dynamics, and ϵ_t captures process noise modeled as zero-mean Gaussian distribution with covariance \mathbf{Q} .

Given an initial state distribution $\mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0, \mathbf{Q})$, the objective is to find a control sequence $\mathbf{U} = \{\mathbf{u}_0, \dots, \mathbf{u}_{T-1}\}$ that minimizes the expected cost over a finite horizon T :

$$\min_{\mathbf{U}} J(\mathbf{U}) = \mathbb{E}_{\mathbf{u} \sim \pi(\cdot | \mathbf{x}_t)} \left[\phi_T(\mathbf{x}_T) + \sum_{t=0}^{T-1} c(\mathbf{x}_t, \mathbf{u}_t) \right]$$

$$\begin{aligned} \text{subject to } & \mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \\ & \mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0, \mathbf{Q}), \\ & \mathbf{u}_t \in \mathcal{U}, \quad \mathbf{x}_t \in \mathcal{X}, \quad \forall t = 0, \dots, T-1 \end{aligned} \quad (2)$$

where $c(\cdot, \cdot)$ is the running cost and $\phi_T(\cdot)$ is the terminal cost. While the above formulation formulates the general setting, solving it requires systematic exploration. To this end, we introduce Uncertainty Guided Exploratory Trajectory Optimization (UGE-TO).

IV. UNCERTAINTY GUIDED EXPLORATORY TRAJECTORY OPTIMIZATION

In this section, we explain the core concepts of UGE-TO. It has two main stages. First, a set of candidate trajectories is generated by perturbing the nominal control sequence. Then, each trajectory is represented as a distribution by propagating its mean state and uncertainty ellipsoids through the system dynamics. This propagation captures both process noise and action selection. In the second stage, these trajectories are refined through a distributional separation step, where the Hellinger distance is used to measure the overlap between

distributions. Fig. 2 demonstrates how these distributions are getting separated through time. Minimizing this overlap encourages diversity in trajectories, allowing them to cover more unique areas in the configuration space while maintaining their dynamic feasibility. Alg. 1 shows the pseudocode of a single iteration of the optimization method. Finally, the resulting set of trajectories will be used as a nominal trajectory selection for the sampling-based MPC, which is explained in detail in Sec. V. We start with the formulation of trajectory distribution modeling.

Algorithm 1: Uncertainty Guided Exploratory Trajectory Optimization (UGE-TO)

Input: Nominal sequence $\bar{\mathbf{U}}$, horizon T , number of trajectories N , rollouts per trajectory M , covariances Σ_u, \mathbf{Q} , divergence metric $\mathcal{H}^2(\cdot, \cdot)$,
Output: Optimized nominal control sequence $\bar{\mathbf{U}}$

- 1 **for** $i = 1, \dots, N$ **do in parallel**
- 2 $\mathbf{U}^{(i)} = \bar{\mathbf{U}} + \eta^{(i)}, \quad \eta^{(i)} \sim \mathcal{N}(\mathbf{0}, \Sigma_u)$
- 3 **for** $t = 1, \dots, T$ **do**
- 4 $\bar{\mathbf{x}}_{t+1}^{(i)} = f(\bar{\mathbf{x}}_t^{(i)}, \mathbf{u}_t^{(i)})$
- 5 $\Sigma_{t+1}^{(i)} = A_t^{(i)} \Sigma_t^{(i)} A_t^{(i)\top} + \mathbf{Q}$
- 6 $\mathcal{T}^{(i)} = \mathcal{N}(\bar{\mathbf{x}}^{(i)}, \Sigma^{(i)})$
- 7 **Distributional Separation:**
- 8 **for** $i = 2, \dots, N$ **do in parallel**
- 9 **for** $m = 1, \dots, M$ **do**
- 10 $\mathbf{U}^{(i,m)} = \mathbf{U}^{(i)} + \eta^{(i,m)}, \quad \eta^{(i,m)} \sim \mathcal{N}(\mathbf{0}, \Sigma_u)$
- 11 **build** $\mathcal{T}^{(i,m)} = \mathcal{N}(\bar{\mathbf{x}}_c^{(i,m)}, \Sigma_c^{(i,m)})$
- 12 **for** $m = 1, \dots, M$ **do**
- 13 $h^{(i,m)} =$
 $\sum_{n \neq i}^N \mathcal{H}^2(\mathcal{N}(\bar{\mathbf{x}}_c^{(i,m)}, \Sigma_c^{(i,m)}), \mathcal{N}(\bar{\mathbf{x}}^{(n)}, \Sigma^{(n)}))$
- 14 $m^* = \arg \max_{m \in \{1, \dots, M\}} h^{(i,m)}$
- 15 $\mathbf{U}^{(i)} \leftarrow \mathbf{U}^{(i,m^*)}$

A. Trajectory Distribution Modeling

Given a nominal control sequence $\bar{\mathbf{U}} = \{\bar{\mathbf{u}}_0, \dots, \bar{\mathbf{u}}_{T-1}\}$, we generate N sampled control sequences by injecting Gaussian perturbations:

$$\mathbf{U}^{(i)} = \bar{\mathbf{U}} + \boldsymbol{\eta}^{(i)}, \quad \boldsymbol{\eta}^{(i)} \sim \mathcal{N}(\mathbf{0}, \Sigma_u) \quad (3)$$

Each control sequence is propagated through the system dynamics to obtain a trajectory $\mathbf{X}^{(i)} = \{\mathbf{x}_0^{(i)}, \dots, \mathbf{x}_T^{(i)}\}$:

$$\mathbf{x}_{t+1}^{(i)} = f(\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)}) + \epsilon_t^{(i)}, \quad \epsilon_t^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \quad (4)$$

We approximate trajectory uncertainty by propagating the covariance forward using the linearized dynamics. With $A_t^{(i)} = \frac{\partial f}{\partial \mathbf{x}} \Big|_{\mathbf{x}_t^{(i)}, \mathbf{u}_t^{(i)}}$, the covariance update is:

$$\Sigma_{t+1}^{(i)} = A_t^{(i)} \Sigma_t^{(i)} A_t^{(i)\top} + \mathbf{Q} \quad (5)$$

Thus, each trajectory is modeled as a Gaussian distribution:

$$\mathcal{T}^{(i)} = \mathcal{N}(\boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}^{(i)}) \quad (6)$$

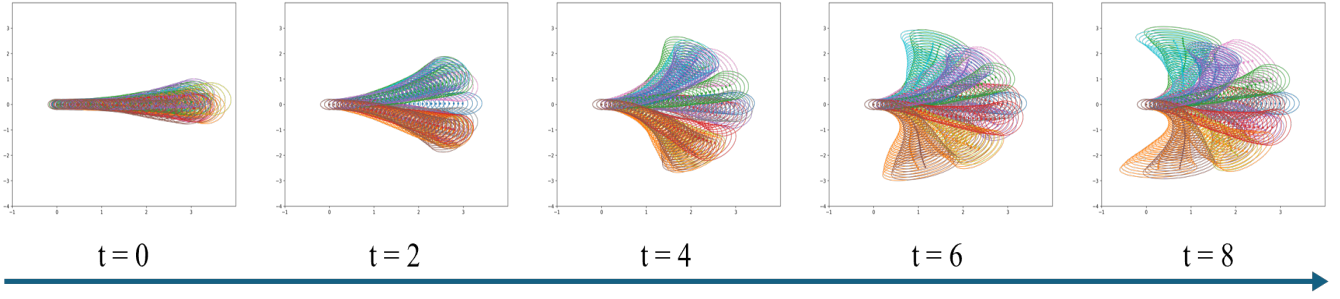


Fig. 2: Illustration of the Hellinger Distance-based Distributional Separation: Candidate rollouts (leftmost, $t = 0$) are represented as trajectory distributions with uncertainty ellipsoids. Optimization reduces overlap among distributions, resulting in a diverse set of nominal trajectories. The figure demonstrates how the distributional separation happens and presents two-dimensional projections of the three-dimensional state space $[x, y, \theta]$. The final trajectory distributions cover the configuration space more broadly, which enables better exploration and a nominal trajectory selection for SBMPCs.

with mean $\boldsymbol{\mu}^{(i)} = [\mathbf{x}_0^{(i)\top}, \dots, \mathbf{x}_T^{(i)\top}]^\top$ and block-diagonal covariance $\boldsymbol{\Sigma}^{(i)} = \text{blockdiag}(\boldsymbol{\Sigma}_0^{(i)}, \dots, \boldsymbol{\Sigma}_T^{(i)})$.

B. Distributional Separation

To promote diversity, we refine each trajectory by generating M additional candidate rollouts:

$$\mathbf{U}^{(i,m)} = \mathbf{U}^{(i)} + \boldsymbol{\eta}^{(i,m)}, \quad \boldsymbol{\eta}^{(i,m)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_u) \quad (7)$$

which yield trajectory distributions $\mathcal{T}^{(i,m)}$.

Each candidate is evaluated by measuring its dissimilarity to the other trajectories. We use the squared Hellinger distance to quantify distributional overlap because it is symmetric and bounded in $[0, 1]$, unlike KL-type divergences, which can be unbounded [29].

$$\mathcal{H}^2(\mathcal{T}^{(i,m)}, \mathcal{T}^{(j)}) = 1 - \frac{|\boldsymbol{\Sigma}^{(i,m)}|^{1/4} |\boldsymbol{\Sigma}^{(j)}|^{1/4}}{|\bar{\boldsymbol{\Sigma}}_{(i,m,j)}|^{1/2}} \times \exp\left(-\frac{1}{8} \Delta\boldsymbol{\mu}_{(i,m,j)}^\top (\bar{\boldsymbol{\Sigma}}_{(i,m,j)})^{-1} \Delta\boldsymbol{\mu}_{(i,m,j)}\right) \quad (8)$$

where $\Delta\boldsymbol{\mu}_{(i,m,j)} = \boldsymbol{\mu}^{(i,m)} - \boldsymbol{\mu}^{(j)}$, and $\bar{\boldsymbol{\Sigma}}_{(i,m,j)} = \frac{1}{2}(\boldsymbol{\Sigma}^{(i,m)} + \boldsymbol{\Sigma}^{(j)})$. The separation score of candidate (i, m) is:

$$h^{(i,m)} = \sum_{j \neq i} \mathcal{H}^2(\mathcal{T}^{(i,m)}, \mathcal{T}^{(j)}) \quad (9)$$

For each trajectory i , we select the candidate with the largest separation score:

$$m^* = \arg \max_{m \in M} h^{(i,m)}, \quad \mathbf{U}^{(i)} \leftarrow \mathbf{U}^{(i,m^*)} \quad (10)$$

This separation procedure is repeated for K iterations to enhance the diversity of the trajectory set.

V. UNCERTAINTY GUIDED EXPLORATORY MODEL PREDICTIVE CONTROL

We now incorporate the proposed UGE-TO into a sampling-based Model Predictive Controller. We use the Model Predictive Path Integral (MPPI) algorithm as the baseline, and augment it with the UGE-TO.

We start with the initialization of a nominal control sequence $\bar{\mathbf{U}} = \{\bar{\mathbf{u}}_0, \dots, \bar{\mathbf{u}}_{T-1}\}$. It is either initialized as a zero control sequence or it is obtained from the previous iteration.

After the initialization, we apply the UGE-TO procedure described in Alg. 1 to the nominal trajectory to generate distributionally separated trajectories that remain diverse and explore a broad region of the state space.

Algorithm 2: Uncertainty-Guided Exploratory MPC (UGE-MPC)

Input: Nominal sequence $\bar{\mathbf{U}}$, horizon T , refinement size N, M , rollout size L , covariances $\boldsymbol{\Sigma}_u, \boldsymbol{Q}$, temperature λ

Output: Updated nominal control sequence $\bar{\mathbf{U}}$

- 1 **Apply UGE-TO**($\bar{\mathbf{U}}, T, N, M, \boldsymbol{\Sigma}_u, \boldsymbol{Q}$) \leftarrow Alg. 1
 - 2 Evaluate $J(\mathbf{U}^{(i)})$ for all i
 - 3 Select the best rollout: $i^* = \arg \min_i J(\mathbf{U}^{(i)})$
 - 4 Update nominal: $\bar{\mathbf{U}} \leftarrow \mathbf{U}^{(i^*)}$
 - 5 **for** $l = 1, \dots, L$ **do in parallel**
 - 6 Sample $\mathbf{U}^{(l)} = \bar{\mathbf{U}} + \boldsymbol{\eta}^{(l)}$, $\boldsymbol{\eta}^{(l)} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_u)$
 - 7 Propagate $\mathbf{U}^{(l)}$ through dynamics f to obtain $\mathbf{X}^{(l)}$
 - 8 Compute cost $J(\mathbf{U}^{(l)})$
 - 9 Compute weights $w^{(l)} = \exp(-J(\mathbf{U}^{(l)})/\lambda)$
 - 10 Update nominal: $\bar{\mathbf{U}} \leftarrow \bar{\mathbf{U}} + \frac{\sum_l w^{(l)} \boldsymbol{\eta}^{(l)}}{\sum_l w^{(l)}}$
-

After refinement, each trajectory $\mathbf{U}^{(i)}$ is evaluated using the cost function in Eq. 2. Instead of performing the importance-weighted update as in general sampling-based MPCs, UGE-MPC selects the best rollout as the new nominal sequence:

$$i^* = \arg \min_{i \in \{1, \dots, N\}} J(\mathbf{U}^{(i)}), \quad \bar{\mathbf{U}} \leftarrow \mathbf{U}^{(i^*)}. \quad (11)$$

This emphasizes that the nominal trajectory is directly related to the lowest-cost feasible rollout after diversity enforcement. After the new nominal control sequence, MPPI generates L candidate control sequences by injecting Gaussian perturbations:

$$\mathbf{U}^{(l)} = \bar{\mathbf{U}} + \boldsymbol{\eta}^{(l)}, \quad \boldsymbol{\eta}^{(l)} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_u). \quad (12)$$

Each sequence is propagated through the dynamics f in Eq. 1 obtain trajectories $\mathbf{X}^{(l)}$. As in the solution of the MPPI [6], the control sequence is updated as the cost-weighted average:

$$\bar{\mathbf{U}} \leftarrow \bar{\mathbf{U}} + \frac{\sum_{l=1}^L w^{(l)} \boldsymbol{\eta}^{(l)}}{\sum_{l=1}^L w^{(l)}}, \quad w^{(l)} = \exp\left(-\frac{1}{\lambda} J(\mathbf{U}^{(l)})\right). \quad (13)$$

Since UGE-TO only affects the nominal control sequence, we kept the MPPI update rule unchanged. Finally, after obtaining the MPPI solution, we apply the first control $\bar{\mathbf{u}}_0$ to the system, as in general MPC practice, and then shift the control sequence horizon. This procedure repeats until the stopping conditions are met.

VI. EXPERIMENTS

We evaluate the UGE-MPC algorithm by studying the following questions through experiments.

- 1) Can the UGE-MPC converge to a solution when initialized from a poor warm start, including cases where the initial control sequence drives the system in a direction opposite to the goal? (Sec. VI-B)
- 2) Can the UGE-MPC algorithm adapt and perform reliably in dynamic and complex environments in both simulation and real-world end-to-end navigation scenarios? (Secs. VI-C, VI-D)

A. Experimental Setup

1) *Baselines:* To evaluate our approach, we compare against three baseline controllers of increasing complexity: Model Predictive Path Integral control (MPPI) [6], MoG-MPPI [30], and Stein Variational MPC (SV-MPC) [10]. MPPI samples action sequences through a unimodal Gaussian distribution. MoG-MPPI extends this by introducing multiple Gaussian components to capture multimodality. Finally, SV-MPC adds a repulsion term among particles to encourage diversity. This progression of baselines demonstrates the incremental effects of adding multimodality and repulsion, and provides a benchmark for assessing our distributional separation in the configuration space. All methods use the same trajectory sampling budget (2048). We evaluate every method with a fixed number of trajectory samples to ensure a fair comparison. MoG-MPPI and SV-MPC use 16 particles, each with 128 samples. We adopt the same particle settings but adjust the sample allocations proportionally to the number of iterations, ensuring that the total budget remains constant while accounting for iterative updates. All baselines are implemented based on [10].

2) *Cost Function:* The cost function J has two main components: The state cost $\mathcal{C}_{\text{state}}(x_t)$, which includes both obstacle $\mathcal{C}_{\text{obs}}(x_t)$ and distance-to-goal cost $\mathcal{C}_{\text{dist}}(x_t, x_{\text{goal}})$. On the other hand, the second term is the action regulator $\mathcal{C}_{\text{u}}(u_t)$, which smoothes the action to reduce jitter in the resulting trajectory. Corresponding weights are assigned to each component to give relative importance to each term. Therefore, the total cost is calculated over a time horizon T as follows:

$$J(\tau, \mathbf{U}) = \phi(\mathbf{x}_T) + \sum_{t=0}^{T-1} (\lambda_{\text{u}} \mathcal{C}_{\text{u}}(\mathbf{u}_t) + \mathcal{C}_{\text{state}}(\mathbf{x}_t^{\tau}, \mathbf{x}_{\text{G}})), \quad (14)$$

where a trajectory $\tau = F(\mathbf{x}_{\text{curr}}, \mathbf{U}) = \{\mathbf{x}_t, \mathbf{u}_t\}_{t=0}^T$ and the initial state is equal to the current state $\mathbf{x}_0 = \mathbf{x}_{\text{curr}}$. The terminal cost is $\phi(\mathbf{x}_T) = \mathcal{C}_{\text{state}}(\mathbf{x}_T^{\tau}, \mathbf{x}_{\text{G}})$. We introduce an indicator variable δ_t :

$$\delta_t = \begin{cases} 1, & \exists \mathbf{x}_i^{\tau} \in \{\mathbf{x}_i^{\tau}\}_{i=0}^{t-1} \text{ s.t. } \mathbf{x}_i \text{ is in collision,} \\ 0, & \text{otherwise,} \end{cases} \quad (15)$$

and define the unified stage cost as:

$$\mathcal{C}_{\text{state}}(\mathbf{x}_t^{\tau}, \mathbf{x}_{\text{G}}) = \delta_t (\lambda_{\text{obs}} \mathcal{C}_{\text{collided}} + \lambda_{\text{dist}} \mathcal{C}_{\text{dist}}) + (1 - \delta_t) (\lambda_{\text{obs}} \mathcal{C}_{\text{obs}}(\mathbf{x}_t^{\tau}) + \lambda_{\text{dist}} \|\mathbf{x}_t^{\tau} - \mathbf{x}_{\text{G}}\|) \quad (16)$$

where the indicator variable acts as a switch, if a state is in a collision, it will turn the remaining part of the trajectory into a collision, and the distance to goal value of the collided state is used as the distance-to-goal cost for the remaining part of the trajectory. $\mathcal{C}_{\text{collided}} = 10^3$ is the maximum collision cost, and $\mathcal{C}_{\text{obs}}(\mathbf{x}_t^{\tau})$ calculates the cost of the robot footprint of the state based on the local costmap. The term $\mathcal{C}_{\text{dist}}$ denotes the distance-to-goal cost of the state where the collision occurred along a trajectory τ . It is important to note that if any state in a trajectory τ reaches the goal, we stop the cost calculation. This means the trajectory cost is measured only up to the goal-reaching state and terminal cost is calculated as $\phi(\mathbf{x}_T) = \mathcal{C}_{\text{state}}(\mathbf{x}_g^{\tau}, \mathbf{x}_{\text{G}})$ where \mathbf{x}_g is the first goal reaching state along a trajectory τ .

B. Trajectory Optimization

In this experiment, we evaluate the performance of the proposed approach compared to baseline methods in settings where the goal is reachable within the prediction horizon. This setting transforms the MPC formulation into a full, dynamically feasible trajectory optimization problem.

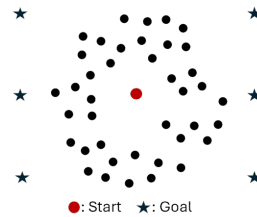


Fig. 3: The cluttered environment with start and goal positions, where the black circles are arbitrarily placed obstacles.

We consider two 8x8 m environments: (i) an obstacle-free space and (ii) a cluttered environment. Fig. 3 demonstrates the cluttered environment and the goal positions. The kinematic bicycle model is used for the dynamics. The start configuration is $[x, y, \theta] = [0, 0, 0]$ and the goal positions are selected in a circular manner to enforce nontrivial and challenging steering and velocity profiles. The nominal trajectory is initialized as a straight-line trajectory along the

TABLE I: Per-goal performance comparison in open and cluttered environments. The average goal-reaching time is computed using only the successful trials. The best results are shown in bold.

Method	Goal (6, 0)		Goal (6, -4)		Goal (-6, -4)		Goal (-6, 0)		Goal (-6, 4)		Goal (6, 4)	
	Open	Cluttered	Open	Cluttered	Open	Cluttered	Open	Cluttered	Open	Cluttered	Open	Cluttered
	SR(↑)/Time(↓)	SR(↑)/Time(↓)	SR(↑)/Time(↓)	SR(↑)/Time(↓)	SR(↑)/Time(↓)	SR(↑)/Time(↓)	SR(↑)/Time(↓)	SR(↑)/Time(↓)	SR(↑)/Time(↓)	SR(↑)/Time(↓)	SR(↑)/Time(↓)	SR(↑)/Time(↓)
MPPI	1.00 / 0.2	1.00 / 0.2	1.00 / 0.3	1.00 / 0.4	1.00 / 1.1	0.80 / 2.9	0.00 / -	0.00 / -	1.00 / 1.3	0.30 / 5.7	1.00 / 0.3	1.00 / 0.4
MoG-MPPI	1.00 / 0.3	1.00 / 0.6	1.00 / 0.7	1.00 / 1.2	1.00 / 1.7	1.00 / 5.2	1.00 / 5.3	0.60 / 8.2	1.00 / 1.7	0.90 / 6.1	1.00 / 0.6	1.00 / 1.3
SV-MPC	1.00 / 0.4	1.00 / 0.4	1.00 / 0.6	1.00 / 1.3	1.00 / 1.7	1.00 / 5.3	1.00 / 4.3	0.60 / 8.6	1.00 / 1.8	0.90 / 5.6	1.00 / 0.7	1.00 / 1.4
UGE-MPC (Ours)	1.00 / 0.1	1.00 / 0.1	1.00 / 0.1	1.00 / 0.1	1.00 / 0.8	1.00 / 2.1	1.00 / 0.8	1.00 / 2.7	1.00 / 1.1	0.90 / 2.7	1.00 / 0.1	1.00 / 0.2

$+x$ direction, with a linear velocity $v = 0.5$ m/s, and a steering angle, $\delta = 0$. The trajectory horizon is $T = 4$ s with the time discretization, $\Delta t = 0.05$ s. Method parameters are set as follows: the temperature value for MPPI iterations is $\alpha = 0.001$, and the epsilon value for SV-MPC is $\epsilon = 1.0$. In this experiment, we use the cost function J as in Eq. 14 without the action regularization term. The weights for the state and obstacle cost are assigned as follows: $\lambda_{\text{obs}} = 50$, and $\lambda_{\text{dist}} = 10$.

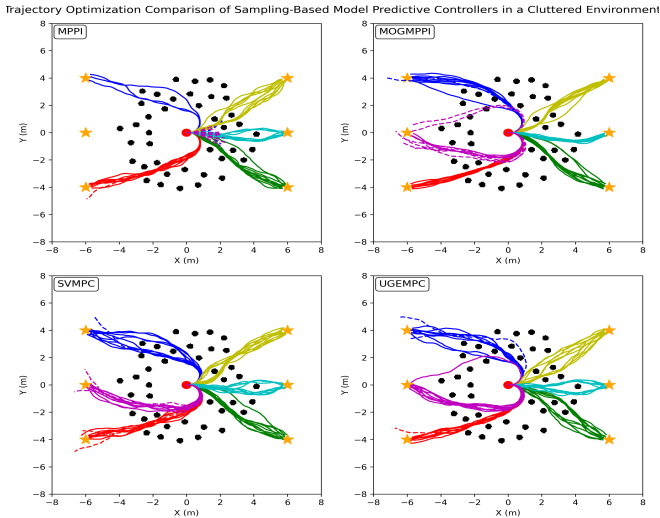


Fig. 4: Trajectory optimization in a cluttered environment: Each start-goal pair is illustrated in a different color, with dashed lines marking unsuccessful runs. MPPI often struggles to find solutions, especially when the goal requires nontrivial velocity and steering profiles (e.g., full steering turns), due to its unimodal sampling distribution. Multimodal methods (MoG-MPPI, SV-MPC) achieve higher success rates but typically require longer times to converge. In contrast, UGE-MPC consistently generates systematically separated candidate trajectories that lead to faster convergence to a solution and provides non-colliding trajectories that improve robustness.

We run 10 trials for each start-goal pair for every method. The maximum number of iterations is set to 100. If a method does not find a solution that reaches the goal within this limit, the trial is considered a failure. In Table I, we report the average success rates and the average first goal reaching time. We highlight the best-performing methods in bold. The results indicate that, in the obstacle-free setting, every method can successfully generate trajectories that reach

the goals. UGE-MPC outperforms all baselines in terms of average goal reaching time. This performance directly highlights that configuration space exploration helps find a candidate trajectory that reaches the goal more quickly than all baselines. Among the baselines, MPPI outperforms the other two multimodal methods in open-environment settings, where it selects a single mode, whereas the other baselines spend more time exploring multiple modes before converging.

Fig. 4 demonstrates the trajectories generated by each method in the cluttered environment. Each start and goal pair’s trajectories are colored differently, where dashed lines show the corresponding unsuccessful runs. In the cluttered environment scenario, UGE-MPC has achieved a higher success rate and lower average time compared to the baselines. On the other hand, in contrast to open space, now multimodal methods overperform the MPPI. Due to the unimodal structure of MPPI, it struggles to explore the environment, especially when the nominal trajectory is far from the optimal trajectory (The left-center goal in Fig. 3). At the same time, other baselines can shift their trajectories. Overall, these results demonstrate that our method, UGE-MPC, achieves high success rates while comparably lower goal-reaching time, making it both efficient and robust in complex environments.

C. Model Predictive Control in Cluttered Environments

In this experiment, we analyzed the performance of our method against baselines in unknown cluttered environments. No prior map information was provided. We evaluated a point-to-point navigation task performance in settings where the vehicle only uses the local costmap derived from the lidar sensor footprint as a sensory input. To increase complexity, each environment (20x20m) was generated with concave polygonal obstacles, which introduced local traps in the environment. The start ($[2, 2, \pi/2]$) and goal ($[18, 18]$) positions are fixed for this experiment. The lidar sensor range was set to 10m, and each controller was run 10 times per environment to capture its behavior. Similarly to the previous trajectory optimization setting, we use trajectories with a $T = 4$ s horizon with 0.05 s as the time discretization. We set method-based hyperparameters as follows: $\alpha = 10^{-3}$ for the MPPI temperature parameters. The cost function Eq. 14 without the action regularization term is used for this experiment, and the corresponding weights for obstacle and state cost are selected by $\lambda_{\text{obs}} = 10^3$, and $\lambda_{\text{dist}} = 10$.

Fig. 5 shows an instance where UGE-TO helps to generate diverse trajectories and shifts the nominal trajectory to

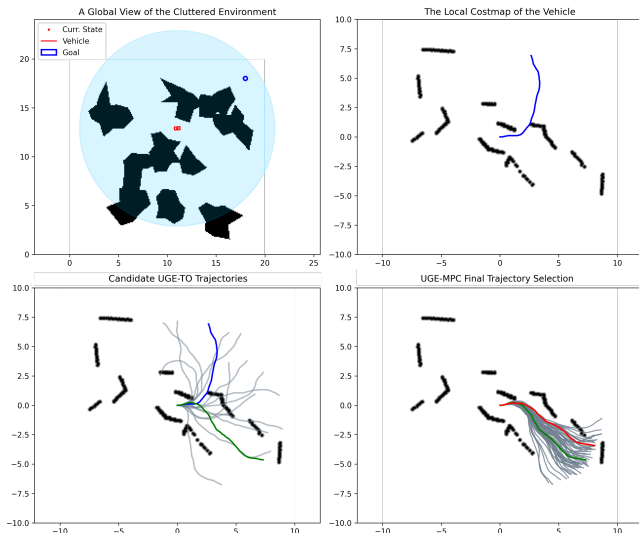


Fig. 5: Visualization of Single iteration of UGE-MPC. **Top-left:** Global view of the environment with the current robot state (red), vehicle footprint (red polygon), and the goal region (blue circle). A semi-transparent blue circle shows the LiDAR range. **Top-right:** Local costmap with the nominal candidate trajectory (blue) expressed in the robot’s local frame before any optimization started. **Bottom-left:** UGE-TO optimizes the candidate trajectories around the nominal trajectories to the trajectories in the figure by minimizing the overlap between their distribution. The lowest cost one is highlighted in green, while others are shown in gray, and the blue color is the nominal trajectory that comes from the previous time step solution. **Bottom-right:** Final step of UGE-MPC is selecting a final trajectory based on the update rule that is used in sampling-based MPC settings in Eq. 13. After the nominal trajectory is selected with UGE-TO iterations. We sample a new set of trajectories, and then the nominal trajectory is updated according to the rule. The new control sequence is highlighted in red.

one that avoids failure. Among these methods, UGE-MPC explored alternative configurations more effectively by explicitly encouraging distributional separation. That produces shorter and more direct trajectories. Table II summarizes the average success rate and goal-reaching time across successful runs.

TABLE II: Performance Comparison of Methods on Unknown Cluttered Environments. The average goal-reaching time is computed using only the successful trials.

Methods	Success Rate (%) (↑)	Average Goal-Reaching Time (s) (↓)
MPPI	0.54	17.3
MoG-MPPI	0.60	24.5
SV-MPC	0.64	25.6
UGE-MPC (Ours)	0.88	21.0

Figure 6 illustrates a performance comparison in one of the five cluttered environments. It can be seen that MPPI, with its unimodal sampling distribution, often converges towards a single mode. In cases where the mode is a local trap, it

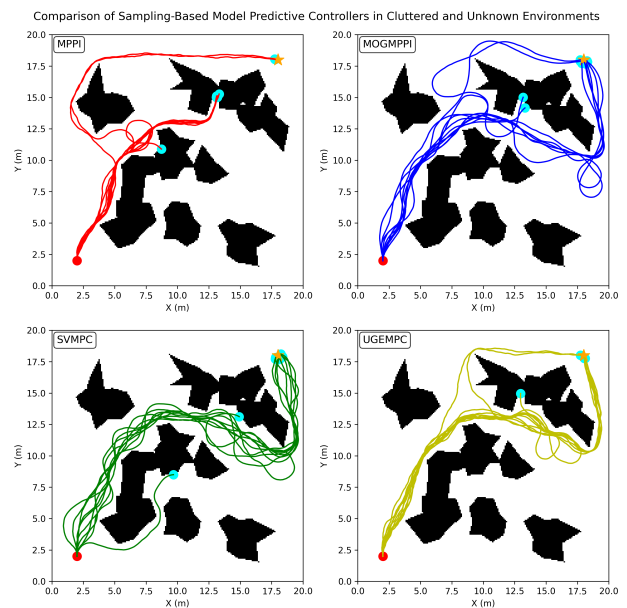


Fig. 6: Performance Comparison in cluttered environments: It shows that MPPI, with its unimodal sampling, struggles to avoid local minima and may end up in one of those minima. On the other hand, MoG-MPPI, SVMPC, and UGE-MPC achieve higher success rates by leveraging exploration. UGE-MPC explores the configuration space more systematically and completes the task with shorter trajectories.

is difficult for MPPI to escape from that local minimum. In contrast, the other baselines (MoG-MPPI and SVMPC) as well as our method, UGE-MPC, achieved higher success rates by leveraging exploration to escape such traps and adapt their nominal trajectories.

D. Real-World Experiments

We further evaluated our approach on a real-world point-to-point navigation task in a cluttered, unknown environment with fixed start and goal positions. We compared our method against two baselines, MPPI and log-MPPI, with five trials per method under a fixed sample budget of $N = 512$. We allocated 126 samples for the final MPPI iterations, using the remaining budget to generate 6 candidate particles with 8 iterations of UGE-TO. MPPI and log-MPPI covariances are both set to $\Sigma = [1.0, 10^\circ]$. Since log-MPPI uses a normal-log-normal distribution, it has higher exploration than MPPI. All controllers were run at 10 Hz to match the frequency of the onboard Slamtec R2 LiDAR. The 1/5-scale vehicle and the environment are shown in Fig. 7. Since SV-MPC and MoG-MPPI could not be executed consistently at the LiDAR frequency, they were excluded from the trials.

In all trials, MPPI became trapped in a local minimum and failed to reach the goal. Both log-MPPI and UGE-MPC consistently succeeded with average times 14.13s and 13.33s, respectively. Under this setup, we did not observe a significant performance gap between log-MPPI and our approach.

In summary, we conclude that the diversification of candidate nominal trajectories provided by our method is a

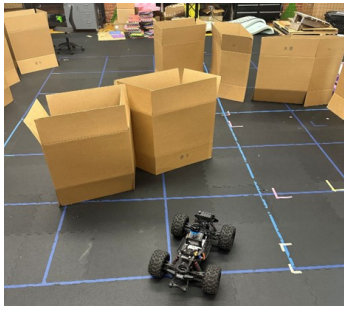


Fig. 7: The 1/5 scale truck platform and the cluttered environment setting

viable approach to improve MPPI performance and avoid local minima.

VII. CONCLUSION

In this work, we introduced Uncertainty Guided Exploratory Trajectory Optimization (UGE-TO) and its integration into a sampling-based model predictive control framework, UGE-MPC. We modeled trajectories as probability distributions and corresponding uncertainty ellipsoids. By distributionally separating them using a metric based on the Hellinger distance, our method enables systematic exploration of the configuration space beyond what existing action-space perturbation techniques can achieve. Through simulation and real-world experiments, we demonstrated that in most cases, UGE-MPC achieves higher success rates and faster convergence compared to MPPI, MoG-MPPI, and SV-MPC under the same sampling budget. In our future work, we aim to demonstrate UGE-MPC in much higher-dimensional systems, such as humanoid robots.

REFERENCES

- [1] Matt Zucker, Nathan Ratliff, Anca D. Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher M. Dellin, J. Andrew Bagnell, and Siddhartha S. Srinivasa. Chomp: Covariant hamiltonian optimization for motion planning. *The International Journal of Robotics Research*, 32(9-10):1164–1193, 2013.
- [2] Anca D. Dragan, Nathan D. Ratliff, and Siddhartha S. Srinivasa. Manipulation planning with goal sets using constrained trajectory optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 4582–4588, Shanghai, China, May 2011. IEEE.
- [3] Jose Luis Sanchez-Lopez, Manuel Castillo-Lopez, Miguel A. Olivares-Mendez, and Holger Voos. Trajectory Tracking for Aerial Robots: an Optimization-Based Planning and Control Approach. *Journal of Intelligent & Robotic Systems*, 100(2):531–574, November 2020.
- [4] John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9):1251–1270, August 2014. Publisher: SAGE Publications Ltd STM.
- [5] Riccardo Bonalli, Abhishek Cauligi, Andrew Bylard, and Marco Pavone. Gusto: Guaranteed sequential trajectory optimization via sequential convex programming. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6741–6747, 2019.
- [6] Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Information-theoretic model predictive control: Theory and applications to autonomous driving. *IEEE Transactions on Robotics*, 34(6):1603–1622, 2018.
- [7] Ihab S Mohamed, Kai Yin, and Lantao Liu. Autonomous navigation of agvs in unknown cluttered environments: log-mpipi control strategy. *IEEE Robotics and Automation Letters*, 7(4):10240–10247, 2022.

- [8] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1352–1361. PMLR, 06–11 Aug 2017.
- [9] Alexander Lambert and Byron Boots. Entropy regularized motion planning via stein variational inference, 2021.
- [10] Alexander Lambert, Adam Fishman, Dieter Fox, Byron Boots, and Fabio Ramos. Stein variational model predictive control. *arXiv preprint arXiv:2011.07641*, 2020.
- [11] Yuichiro Aoyama, Peter Lehmann, and Evangelos A. Theodorou. Second-Order Stein Variational Dynamic Optimization, October 2024. arXiv:2409.04644 [math].
- [12] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm, 2019.
- [13] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Ann. Oper. Res.*, 37(1–4):357–373, January 1992.
- [14] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, March 2006.
- [15] Philip E. Gill, Walter Murray, and Michael A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002.
- [16] D. Q. Mayne. A second-order gradient method of optimizing nonlinear discrete time systems. *International Journal of Control*, 3:85–95, 1966.
- [17] W. Li and E. Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics*. Setubal, Portugal, 2004.
- [18] Oswin So, Ziyi Wang, and Evangelos A. Theodorou. Maximum entropy differential dynamic programming, 2022.
- [19] Yuichiro Aoyama and Evangelos A. Theodorou. Generalized maximum entropy differential dynamic programming, 2024.
- [20] Thomas Power and Dmitry Berenson. Constrained stein variational trajectory optimization, 2024.
- [21] Muhammad Kazim, JunGee Hong, Min-Gyeom Kim, and Kwang-Ki K Kim. Recent advances in path integral control for trajectory optimization: An overview in theoretical and algorithmic perspectives. *Annual Reviews in Control*, 57:100931, 2024.
- [22] Reuven Rubinfeld. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1:127–190, 1999.
- [23] Bogdan Vlahov, Jason Gibson, David D. Fan, Patrick Spieler, Aliakbar Agha-mohammadi, and Evangelos A. Theodorou. Low frequency sampling in model predictive path integral control. *IEEE Robotics and Automation Letters*, 9(5):4543–4550, May 2024.
- [24] Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In Jens Kober, Fabio Ramos, and Claire Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 1049–1065. PMLR, 16–18 Nov 2021.
- [25] Kohei Honda, Naoki Akai, Kosuke Suzuki, Mizuho Aoki, Hirohito Hosogaya, Hiroyuki Okuda, and Tatsuya Suzuki. Stein variational guided model predictive path integral control: Proposal and experiments with fast maneuvering vehicles, 2024.
- [26] O. Goktug Poyrazoglu, Yukang Cao, and Volkan Isler. C-uniform trajectory sampling for fast motion planning, 2024.
- [27] O. Goktug Poyrazoglu, Rahul Moorthy, Yukang Cao, William Chastek, and Volkan Isler. An unsupervised c-uniform trajectory sampler with applications to model predictive path integral control, 2025.
- [28] Ihab S. Mohamed, Junhong Xu, Gaurav S Sukhatme, and Lantao Liu. Towards efficient mppi trajectory generation with unscented guidance: U-mpipi control strategy, 2024.
- [29] Ran Ding and Anthony Mullaht. Empirical squared hellinger distance estimator and generalizations to a family of α -divergence estimators. *Entropy*, 25(4):612, 2023.
- [30] Ziyi Wang, Oswin So, Jason Gibson, Bogdan Vlahov, Manan S. Gandhi, Guan-Hong Liu, and Evangelos A. Theodorou. Variational inference mpc using tsallis divergence, 2021.