2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

SA-MPPI: Sensitivity-Aware Model Predictive Path Integral Control for Robust and Agile Quadrotor Flight

Fuqiang Gu[1], Xu Lu[1], Huidong Liu*[1], Jiangshan Ai[1], Xianlei Long[1], Tao Jiang[2], Chao Chen[1], Zhao Huang[3]

***Abstract*— Reliable quadrotor control in dynamic environments remains challenging due to external disturbances and internal uncertainties. While Model Predictive Path Integral (MPPI) control enables agile maneuvers through sampling-based optimization, its performance often degrades under such unmodeled uncertainties, leading to brittle and unsafe behavior. To address this, we propose SA-MPPI, a novel robust MPC framework that integrates asynchronous guidance with a novel open-loop sensitivity metric. The asynchronous module leverages a slower auxiliary controller to generate an informed sampling distribution, improving convergence without introducing latency. The sensitivity metric penalizes high-variance trajectories under sampled disturbances via nested Monte Carlo rollouts, embedding robustness directly into the optimization. Extensive simulations and real-world quadrotor experiments demonstrate that SA-MPPI outperforms adaptive baselines, reducing tracking errors by up to 47% under significant wind disturbances while achieving over 2× higher computational efficiency. These results highlight SA-MPPI's ability to deliver low-latency, safe, and predictable control in uncertain, dynamic environments.**

I. Introduction

Quadrotors have become indispensable for tasks such as emergency response, package delivery, and industrial inspection due to their agility, versatility, and low cost [1]–[3]. In these applications, reliability is not only essential for mission success but also fundamental to operational safety. However, reliable quadrotor control remains challenging, as the dynamics are subject to considerable uncertainties. Such uncertainties may arise from external disturbances (e.g., turbulent wind or rapidly changing environments) or internal variations (e.g., battery aging) [4]. These factors significantly complicate the dynamics and pose severe challenges for robust and safe control.

Model predictive control (MPC) has emerged as a leading solution for uncertainty-aware quadrotor control. Existing approaches can be broadly categorized as deterministic, stochastic, or learning-based. Deterministic MPC enforces strict safety constraints and offers guarantees under nominal conditions but often fails under unmodeled disturbances [5]. Stochastic MPC explicitly incorporates uncertainty using probabilistic models, chance constraints, or Conditional Value-at-Risk (CVaR) formulations [6], [7], improving robustness but typically incurring high computational cost. Learning-based methods leverage reinforcement learning or neural networks to adapt to complex uncertainties [8], but they require extensive training data and often generalize poorly to unseen environments.

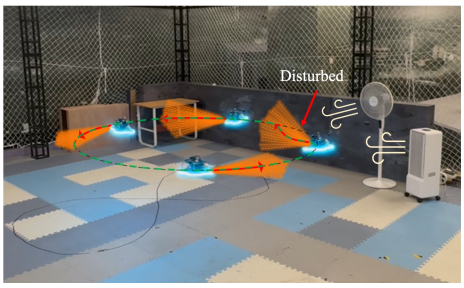


Fig. 1: Trajectory of a real-world flight experiment. The proposed SA-MPPI controller enables accurate circular trajectory tracking under significant wind disturbance from a fan, demonstrating robust and precise control in challenging environments.

Despite these advances, most MPC variants remain reactive, compensating for disturbances only after they occur, rather than proactively accounting for them in planning. Risk-sensitive extensions, such as CVaR-based methods [7], [9], attempt to address this limitation, but they often suffer from over-conservatism and heavy computational demands, limiting their agility in dynamic missions. This motivates the need for a proactive, computationally efficient framework that balances robustness, safety, and real-time performance.

To address this challenge, we propose SA-MPPI, a robust MPC framework that integrates an asynchronous guidance architecture with a novel sensitivity-aware cost metric. Unlike standard MPPI, SA-MPPI proactively penalizes control sequences with high cost variance under disturbances, ensuring trajectories that are safe, consistent, and predictable without sacrificing agility. Specifically, SA-MPPI operates in a receding-horizon manner: at each timestep, perturbed control sequences are sampled around a nominal guidance trajectory, their expected costs and sensitivities are evaluated via nested Monte Carlo rollouts, and a weighted optimal action is computed. The asynchronous guidance module

*This paper is supported by the National Natural Science Foundation of China (No. 42474027, No. 62403085), Graduate Research and Innovation Foundation of Chongqing, China (No. CYB240066), and the Xiaomi Young Talents Program. Corresponding author: Huidong Liu

[1]F. Gu, X. Lu, H. Liu, J. Ai, X. Long, and C. Chen are with the College of Computer Science, Chongqing University, Chongqing, China. e-mails: {gufq, lux, xianlei.long, cschaochen}@cqu.edu.cn, {liuhuidong, ajs}@stu.cqu.edu.cn

[2]T. Jiang is with the School of Automation, Chongqing University, Chongqing, China. e-mail: jiangtao_1992@cqu.edu.cn

[3]Z. Huang is with the School of Computer Sciences, Northumbria University, United Kingdom. e-mail: zhao.huang@northumbria.ac.uk

leverages a slower auxiliary controller to refine the nominal trajectory in the background, thereby improving convergence without introducing real-time latency. As shown in Fig. 1, SA-MPPI takes the current state and reference trajectory as inputs and produces an optimal control action, ensuring robust and accurate trajectory tracking in the presence of strong wind disturbances from a fan.

The main contributions of this work are:

1) We propose an asynchronous architecture that decouples the high-frequency MPPI control loop from a slower auxiliary controller, enabling high-quality sampling priors that accelerate convergence while preserving real-time responsiveness.
2) We introduce a sampling-compatible open-loop sensitivity metric that quantifies robustness as cost variance under disturbance uncertainty, steering optimization toward safe, consistent, and predictable control policies.
3) We demonstrate the effectiveness of SA-MPPI through extensive simulations and challenging real-world quadrotor experiments. Results show significant gains in robustness, tracking accuracy, and control smoothness compared to adaptive baselines.

II. Related Works

Safe and agile quadrotor control is hindered by uncertainties from unmodeled dynamics and external disturbances. While MPC offers a principled framework, its computational burden limits real-time use in highly dynamic tasks. MPPI control alleviates this via sampling-based optimization but remains brittle, as open-loop predictions are sensitive to disturbances. These challenges have motivated two main research directions: reactive and proactive methods.

Reactive approaches compensate for uncertainties after they manifest. Adaptive control, particularly $\mathcal{L}_1$ methods [10], [11], dynamically estimate and cancel disturbances. Hybrids such as $\mathcal{L}_1$-MPPI [12] leverage adaptive elements to align real dynamics with nominal models, improving tracking accuracy. However, such methods often introduce high-frequency chatter and remain reactive rather than anticipatory. Structured robustness methods, such as tube-based MPC [13], maintain a bounded tube around nominal trajectories, while safety-critical architectures employ control barrier functions (CBFs) [14], [15] to filter unsafe actions in real time. Although effective for guaranteeing safety, these methods typically impose conservatism or require solving additional constrained optimization problems online, both of which reduce quadrotor agility and scalability in complex missions.

Proactive methods embed uncertainty directly into the optimization problem. Risk-aware MPC frameworks penalize tail risks by incorporating measures such as CVaR [16]–[18], allowing the controller to hedge against worst-case outcomes. Sensitivity-aware MPC [19] evaluates how small parametric changes affect trajectories, biasing solutions toward more robust plans. These approaches anticipate disturbances rather than react to them, but their benefits come at the cost of heavy computation due to complex risk evaluations or large-scale sampling requirements, which often preclude real-time deployment on embedded systems.

In summary, reactive methods enhance safety but sacrifice agility, while proactive methods improve robustness but remain computationally intensive and overly conservative. Our proposed SA-MPPI addresses this gap by introducing a sampling-compatible sensitivity metric to proactively bias the controller toward stable solutions, while an asynchronous guidance architecture decouples real-time control from slower optimization processes. This combination enables robust, safe, and agile quadrotor control in uncertain environments without sacrificing efficiency.

III. Preliminaries

A. Problem Formulation

The challenge of quadrotor control lies in achieving precise trajectory tracking in dynamic environments subject to external disturbances, which demand robust and computationally efficient strategies. Consider a quadrotor with state $\mathbf{x}_t \in \mathbb{R}^n$ (e.g., position, velocity, orientation, body rates) and control inputs $\mathbf{u}_t \in \mathbb{R}^m$ (e.g., rotor thrusts) at time-step t. The system evolves according to nonlinear stochastic dynamics:

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

where $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}^n$ denotes the dynamics function, and $\mathbf{w}_t \in \mathbb{R}^d$ represents disturbances drawn from a distribution $\mathcal{D}$, typically modeled as zero-mean Gaussian, $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \Sigma_w)$, though adaptable to other distributions.

The objective is to compute an optimal control sequence $\mathbf{U}_t = \{\mathbf{u}_t, \mathbf{u}_{t+1}, \ldots, \mathbf{u}_{t+N-1}\} \in \mathbb{R}^{m\times N}$ over a planning horizon N that minimizes the expected trajectory-tracking cost under disturbances. The cost function is defined as:

$$J(\mathbf{x}_{t:t+N}, \mathbf{u}_{t:t+N-1}) = \sum_{k=t}^{t+N-1} c(\mathbf{x}_k, \mathbf{u}_k) + c_T(\mathbf{x}_{t+N}), \tag{2}$$

where $c(\mathbf{x}_k, \mathbf{u}_k) = \|\mathbf{x}_k - \mathbf{x}_{\text{ref},k}\|_Q^2 + \|\mathbf{u}_k\|_R^2$ denotes the stage cost, penalizing deviations from the reference trajectory $\mathbf{x}_{\text{ref},k}$ and control effort with weighting matrices $Q \succeq 0$ and $R \succ 0$, and $c_T(\mathbf{x}_{t+N}) = \|\mathbf{x}_{t+N} - \mathbf{x}_{\text{ref},t+N}\|_P^2$ denotes the terminal cost with $P \succeq 0$. The optimization problem is:

$$\mathbf{U}_t^* = \arg\min_{\mathbf{U}_t} \mathbb{E}_{\mathbf{w}_{t:t+N-1}\sim\mathcal{D}}[J(\mathbf{x}_{t:t+N}, \mathbf{U}_t)]. \tag{3}$$

Standard MPPI struggles with unmodeled disturbances, leading to brittle performance, while risk-aware methods (e.g., CVaR-based) incur high computational costs [6]. SA-MPPI addresses these limitations through an $\mathcal{L}_1$ disturbance observer, asynchronous guidance, and a sensitivity-aware cost metric, as detailed in Section IV.

B. Quadrotor Dynamics

The quadrotor is modeled with respect to an inertial world frame W and a body-fixed frame B, as illustrated in Fig. 2. The state is defined as $\mathbf{x}_t = [\mathbf{p}_W^\top, \mathbf{v}_W^\top, \mathbf{q}_{WB}^\top, \boldsymbol{\omega}_B^\top]^\top \in \mathbb{R}^{13}$, where $\mathbf{p}_W \in \mathbb{R}^3$ and $\mathbf{v}_W \in \mathbb{R}^3$ represent the position and

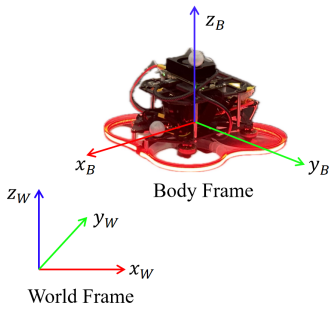


Fig. 2: The real quadrotor model and its position representation in the world frame (W) and body frame (B).

linear velocity in the world frame, $\mathbf{q}_{WB} \in \mathbb{R}^4$ denotes the unit quaternion representing the orientation from body to world frame, and $\boldsymbol{\omega}_B \in \mathbb{R}^3$ denotes the angular velocity in the body frame.

The dynamics of the quadrotor are given by:

$$\begin{aligned} \dot{\mathbf{p}}_W &= \mathbf{v}_W, \\ \dot{\mathbf{v}}_W &= \mathbf{g} + \frac{1}{m}\mathbf{R}(\mathbf{q}_{WB})[0,0,f_T]^\top, \\ \dot{\mathbf{q}}_{WB} &= \frac{1}{2}\mathbf{q}_{WB} \odot [0, \boldsymbol{\omega}_B]^\top, \\ \dot{\boldsymbol{\omega}}_B &= \mathbf{J}^{-1}\big(\boldsymbol{\tau}_B - \boldsymbol{\omega}_B \times \mathbf{J}\boldsymbol{\omega}_B\big), \end{aligned} \tag{4}$$

where $m > 0$ represents the quadrotor mass, $\mathbf{J} \in \mathbb{R}^{3\times3}$ represents the diagonal inertia matrix, $\mathbf{g} = [0,0,-g]^\top$ denotes gravitational acceleration, and $\mathbf{R}(\mathbf{q}_{WB}) \in SO(3)$ converts vectors from the body to the world frame. The symbol $\odot$ denotes quaternion multiplication. The control input is $\mathbf{u} = [f_T, \boldsymbol{\tau}_B^\top]^\top \in \mathbb{R}^4$, where f_T represents the total thrust along the body z-axis, and $\boldsymbol{\tau}_B \in \mathbb{R}^3$ represents the body torque vector.

C. Sampling-based Model Predictive Control

MPPI is a sampling-based method that approximates the optimal control by evaluating stochastic perturbations of a nominal sequence. At each timestep, K control sequences $\mathbf{U}^k = \{\mathbf{u}_t^k, \ldots, \mathbf{u}_{t+N-1}^k\}$, $k = 1, \ldots, K$, are sampled around $\bar{\mathbf{U}}_t$:

$$\mathbf{u}_t^k = \bar{\mathbf{u}}_t + \delta\mathbf{u}_t^k, \quad \delta\mathbf{u}_t^k \sim \mathcal{N}(\mathbf{0}, \Sigma_u), \tag{5}$$

where Σ_u represents the perturbation covariance. Each sequence is propagated through the system dynamics to obtain trajectories $\mathbf{X}^k = \{\mathbf{x}_1^k, \ldots, \mathbf{x}_T^k\}$, and the associated cost $J^k = J(\mathbf{X}^k, \mathbf{U}^k)$ is computed, then we can get the weight for each sequence:

$$w^k \leftarrow \frac{\exp(-\frac{1}{\lambda}J^k)}{\sum_{j=1}^K \exp(-\frac{1}{\lambda}J^j)}, \tag{6}$$

where $\lambda > 0$ represents the temperature parameter controlling exploration. The nominal sequence is updated by

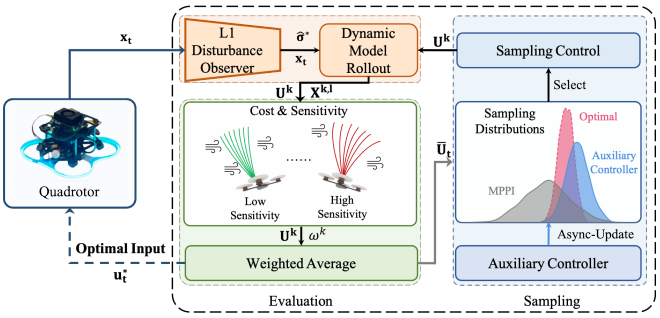


Fig. 3: SA-MPPI control framework. An $\mathcal{L}_1$ observer estimates disturbances from the current state ($\mathbf{x}_t$) to augment the dynamics model. An asynchronous auxiliary controller provides high-quality guidance to inform the sampling of control sequences ($\mathbf{U}^k$). A sensitivity-aware evaluation then uses nested rollouts to penalize sequences with high cost variance under disturbances, ensuring the selection of a robust optimal control input ($\mathbf{u}_t^*$).

shifting the cost-weighted averaging optimal sequence:

$$\begin{aligned} \bar{\mathbf{U}}_t &\leftarrow \{\mathbf{u}_{t+1}^*, \ldots, \mathbf{u}_{t+N-1}^*, \mathbf{0}\}, \\ \mathbf{U}_t^* &\leftarrow \sum_{k=1}^K w^k \mathbf{U}^k. \end{aligned} \tag{7}$$

Only the first control $\mathbf{u}_t^*$ is applied, and the procedure repeats in a receding-horizon fashion.

IV. Proposed Method: SA-MPPI

The SA-MPPI framework integrates three core modules: (1) an $\mathcal{L}_1$ disturbance observer for real-time model augmentation, (2) an asynchronous guidance module that leverages an auxiliary controller to generate an informed sampling distribution, and (3) a sensitivity-aware cost metric that penalizes sequences with high variance under disturbances. The overall architecture is illustrated in Fig. 3. In operation, the $\mathcal{L}_1$ observer estimates wind and other external disturbances for online model augmentation, while the asynchronous guidance module supplies high-quality reference trajectories at a reduced frequency. The sensitivity-aware evaluation then employs nested Monte Carlo rollouts to compute expected costs while explicitly penalizing variance, thereby favoring robust control sequences. This combination enables the quadrotor to reliably track aggressive trajectories with improved efficiency and resilience to wind disturbances.

A. Incorporation of $\mathcal{L}_1$ Disturbance Observer

To improve robustness against unmodeled dynamics, parametric uncertainties, and external disturbances (e.g., wind gusts), SA-MPPI augments the nominal dynamics (Eq. (4)) with an $\mathcal{L}_1$ disturbance observer. Disturbances are modeled as additive force $\boldsymbol{\delta}_f = [\delta_{fx}, \delta_{fy}, \delta_{fz}]^\top \in \mathbb{R}^3$ and torque $\boldsymbol{\delta}_\tau = [\delta_{\tau x}, \delta_{\tau y}, \delta_{\tau z}]^\top \in \mathbb{R}^3$, expressed in the body frame.

The modified dynamics are given by:

$$\begin{aligned}\dot{\mathbf{v}}_W &= \mathbf{g} + \frac{1}{m}\mathbf{R}(\mathbf{q}_{WB})\big([0,0,f_T]^\top + \boldsymbol{\delta}_f\big),\\ \dot{\boldsymbol{\omega}}_B &= \mathbf{J}^{-1}\Big(\boldsymbol{\tau}_B + \boldsymbol{\delta}_\tau - \boldsymbol{\omega}_B \times (\mathbf{J}\boldsymbol{\omega}_B)\Big),\end{aligned} \tag{8}$$

where the kinematic equations for $\dot{\mathbf{p}}_W$ and $\dot{\mathbf{q}}_{WB}$ remain as in Eq. (4).

Disturbances are categorized into matched components, $\boldsymbol{\sigma}_m = [\delta_{fz}, \delta_{\tau x}, \delta_{\tau y}, \delta_{\tau z}]^\top$, which directly influence the control channels, and unmatched components, $\boldsymbol{\sigma}_{um} = [\delta_{fx}, \delta_{fy}]^\top$, which act as lateral forces. Unlike conventional $\mathcal{L}_1$ adaptive control, which primarily mitigates matched uncertainties via feedback [20], SA-MPPI employs the $\mathcal{L}_1$ framework as a general-purpose disturbance observer. In this role, it provides real-time estimates of the combined disturbance vector $\hat{\boldsymbol{\sigma}}^* = \begin{bmatrix}\hat{\boldsymbol{\sigma}}_m^\top & \hat{\boldsymbol{\sigma}}_{um}^\top\end{bmatrix}^\top$, based on the current state $\mathbf{x}_t$ and control input $\mathbf{u}_t$. These disturbance estimates directly augment the dynamics used in trajectory rollouts, ensuring that the sampling-based optimization accounts for external perturbations in real time.

B. Asynchronous Guidance and Informed Sampling

To mitigate the slow convergence and control chatter associated with standard MPPI's purely random sampling [15], SA-MPPI integrates an asynchronous guidance module that supplies high-quality control sequences to bias exploration. Analogous to a navigation system suggesting optimal routes, this module directs sampling toward promising regions of the control space while preserving real-time responsiveness.

This is achieved by decoupling the high-frequency MPPI loop (e.g., 100 Hz) from a lower-frequency auxiliary controller (e.g., 20–50 Hz), leveraging a heterogeneous computing architecture. The massively parallel nature of MPPI's sampling and simulation rollouts makes it ideally suited for GPU execution. Concurrently, the auxiliary controller, such as a nonlinear MPC or trajectory optimizer which often relies on sequential computations, runs on the CPU. This parallel structure allows the two components to operate asynchronously, preventing the computationally demanding guidance calculation from introducing latency into the real-time control loop. In the background, the auxiliary controller utilizes a high-fidelity dynamics model to compute a guidance sequence $\mathbf{U}_{\text{aux}} \in \mathbb{R}^{m\times N}$.

The sampling module then generates K perturbed sequences:

$$\mathbf{U}^k \sim \begin{cases}\mathcal{N}(\mathbf{U}_{\text{aux}}, \Sigma_u) & \text{if } k \pmod 2 = 0,\\ \mathcal{N}(\bar{\mathbf{U}}_{t-1}, \Sigma_u) & \text{if } k \pmod 2 = 1,\end{cases} \tag{9}$$

where $k = 1, \ldots, K$, and $\Sigma_u \in \mathbb{R}^{m\times m}$ denoting the perturbation covariance. This informed sampling procedure biases exploration toward regions of higher expected performance, thereby reducing control chatter and accelerating convergence compared to serial sampling methods [21]. Within the SA-MPPI pipeline, $\mathbf{U}_{\text{aux}}$ is computed using the current state $\mathbf{x}_t$ and reference trajectory, and its influence propagates through Eq. (9). The resulting candidate sequences $\mathbf{U}^k$ are then evaluated under dynamics augmented by the disturbance estimate $\hat{\boldsymbol{\sigma}}^*$ (Section IV-A), ensuring both robustness and efficiency.

Algorithm 1: SA-MPPI Algorithm

Input: Current state $\mathbf{x}_t$, Previous nominal sequence $\bar{\mathbf{U}}_{t-1}$, Disturbance distribution $\mathcal{D}$, Number of samples K, Number of rollouts L, Horizon length N, Temperature parameter λ, Sensitivity weight γ, Reference trajectory $\mathbf{x}_{\text{ref},t:t+N}$
Output: Optimal control input $\mathbf{u}_t^*$, Updated nominal sequence $\bar{\mathbf{U}}_t$

```
Compute σ̂* using L1 observer;
for k = 1, ..., K do
    if k mod 2 = 0 then
        // Eq. (9)
        Get U_aux from auxiliary controller;
        Sample U^k = U_aux + δU^k, δU^k ~ N(0, Σ_u);
    else
        Sample U^k = Ū_{t-1} + δU^k, δU^k ~ N(0, Σ_u);
    for l = 1, ..., L do
        Simulate X^{k,l} using Eq. (10);
        Compute cost J^{k,l} = J(X^{k,l}, U^k);      // Eq. (11)
    Compute J̄^k, S^k, J^k_total;                    // Eq. (12), (13)
Update U*_t ← Σ_{k=1}^K w^k U^k, with w^k from Eq. (6);
Apply u*_t, shift Ū_t ← {u*_{t+1}, ..., u*_{t+N-1}, 0};
```

From a computational perspective, the asynchronous module substantially improves sample efficiency: vanilla MPPI often requires large K (e.g., thousands of trajectories) to achieve stable performance, whereas SA-MPPI achieves comparable or superior results with significantly fewer samples by concentrating rollouts in high-reward regions. This reduction in effective sampling complexity is crucial for real-time quadrotor control under stringent onboard compute constraints.

C. Sensitivity-Aware Cost Evaluation

To ensure robustness in uncertain environments, SA-MPPI incorporates a sensitivity-aware cost metric that favors control sequences with consistent performance under disturbances. In contrast, standard MPPI may select sequences that achieve low nominal cost but degrade significantly in the presence of unmodeled uncertainties [19]. SA-MPPI biases the search toward trajectories that remain reliable despite perturbations. While gradient-based MPC methods leverage sensitivity analysis for robustness, such approaches are incompatible with sampling-based formulations [19]. To address this gap, SA-MPPI introduces an open-loop sensitivity metric defined as the variance of trajectory costs under disturbances.

For each sampled control sequence $\mathbf{U}^k$ (Section IV-B), L nested Monte Carlo rollouts simulate trajectories subject to disturbances $\mathbf{w}^l \sim \mathcal{D}$, $l = 1, \ldots, L$, using the augmented dynamics:

$$\mathbf{x}_{t+i+1}^{k,l} = f(\mathbf{x}_{t+i}^{k,l}, \mathbf{u}_{t+i}^k, \hat{\boldsymbol{\sigma}}^* + \mathbf{w}_{t+i}^l), \quad \mathbf{x}_t^{k,l} = \mathbf{x}_t, \tag{10}$$

where $\hat{\boldsymbol{\sigma}}^*$ denotes the disturbance estimate from the $\mathcal{L}_1$ observer (Section IV-A), and $\mathbf{w}_{t+i}^l$ models residual uncertainty. The cost of each rollout is

$$J^{k,l} = J(\mathbf{X}^{k,l}, \mathbf{U}^k), \quad \mathbf{X}^{k,l} = \{\mathbf{x}_{t+1}^{k,l}, \ldots, \mathbf{x}_{t+N}^{k,l}\}. \tag{11}$$

with $J(\cdot)$ defined in Eq. (2). The expected cost and sensitivity are computed as:

$$\bar{J}^k = \frac{1}{L}\sum_{l=1}^{L} J^{k,l}, \quad S^k = \frac{1}{L-1}\sum_{l=1}^{L}(J^{k,l} - \bar{J}^k)^2. \quad (12)$$

The sensitivity S^k measures variability in performance, with smaller values indicating more robust sequences. The total cost is then defined as

$$J^k_{\text{total}} = \bar{J}^k + \gamma S^k, \quad (13)$$

where $\gamma \geq 0$ balances nominal performance ($\bar{J}^k$) against robustness (S^k).

The optimal control sequence is obtained by substituting J^k_{total} into Eq. 6 and subsequently solving Eq. 7. Only the first control input $\mathbf{u}^*_t$ is then applied, and the sequence is shifted for the next iteration, ensuring robustness in a receding-horizon fashion.

The SA-MPPI framework thus unifies disturbance estimation, asynchronous guidance, and sensitivity-aware cost evaluation into a cohesive pipeline for robust quadrotor control, as summarized in Algorithm 1.

V. Experiments and Results

This section presents a comprehensive evaluation of the proposed SA-MPPI control framework, with a focus on its capability to achieve safe and robust quadrotor operation under unknown external disturbances. The performance of SA-MPPI was benchmarked against multiple baselines in both simulation and real-world experiments, demonstrating consistent improvements in tracking accuracy, robustness, and safety.

A. Experimental Setup

An extensive set of simulations was conducted to assess the performance of SA-MPPI. All simulations were implemented in a custom Python framework and executed on a Linux workstation equipped with an Intel i7-10700KF CPU, 16 GB RAM, and an NVIDIA RTX 3060 GPU. The low-level control loop was maintained at 1000 Hz, ensuring real-time responsiveness.

1) Quadrotor Model and Parameters: The motor dynamics were modeled as a first-order system, with the mapping from normalized motor command t to thrust T represented by a second-order polynomial:

$$T = c_2 t^2 + c_1 t + c_0. \quad (14)$$

The nominal quadrotor parameters used throughout the simulations are summarized in Table I.

2) Baselines and Controller Configuration: To provide a fair and comprehensive comparison, we evaluated SA-MPPI against four controllers: a standard MPPI controller [22], an $\mathcal{L}_1$-MPPI controller [12], and an $\mathcal{L}_1$-MPC controller [4].

All MPPI-based algorithms, including the proposed SA-MPPI, were configured with the same core hyperparameters: a temperature parameter of $\lambda = 1 \times 10^{-4}$, a prediction horizon of $T_f = 1.0$ s, and $N = 10$ prediction steps. To ensure robust comparisons of accuracy and computational efficiency, the baseline MPPI and $\mathcal{L}_1$-MPPI controllers were assigned $K = 5000$ trajectory samples, representing their best-performance setting. In contrast, SA-MPPI exploits its asynchronous guidance architecture to achieve superior performance with significantly fewer samples, requiring only $K = 100$ outer rollouts and $L = 20$ inner rollouts (2000 total dynamics evaluations).

For the sensitivity evaluation, the disturbance covariance matrix is set to $\Sigma_w = \text{diag}[0.1, 0.01, 0.01, 0.01, 0.1, 0.1]$. The auxiliary controller was implemented as a standard nonlinear MPC with a 2.0 s horizon over 20 discretization steps, solved using the acados toolkit [23]. The MPPI framework and its nested sampling structure were implemented in CUDA for efficient parallelization on the GPU.

3) Metrics: Controller performance was evaluated using two metrics: *Mean Absolute Error (MAE)*, which measures the average deviation between the reference and executed trajectories, and *Root Mean Square Error (RMSE)*, which captures both bias and variance, highlighting larger deviations.

TABLE I: Quadrotor Parameters

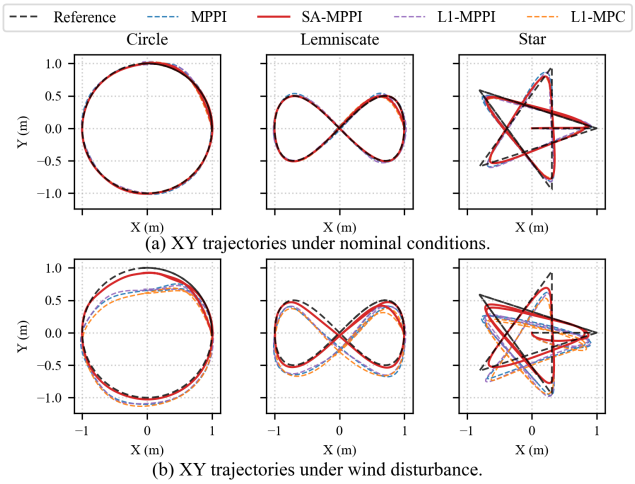
Parameter	Value	Unit
Mass	0.772	kg
Inertia	$\text{diag}([2.5, 2.1, 4.3] \times 10^{-3})$	kg·m^2
Drag Coefficients	$[0.1, 0.1, 0.1]$	N/(m/s)
Motor Time Constant	0.033	s
Arm Length	0.08	m
Torque Constant (κ)	0.016	N·m/N
Thrust Map(c_2, c_1, c_0)	$[1 \times 10^{-6}, 8.0, 0.0]$	–
Min Motor Thrust	0.0	N
Max Motor Thrust	8.0	N

TABLE II: Quantitative results of MAE ($\pm$ std) for trajectory tracking performance under nominal conditions,

Method	Circle	Lemniscate	Star
MPPI	0.077 $\pm$ 0.046	0.079 $\pm$ 0.043	0.179 $\pm$ 0.075
$\mathcal{L}_1$-MPPI	0.079 $\pm$ 0.042	0.079 $\pm$ 0.041	0.173 $\pm$ 0.074
$\mathcal{L}_1$-MPC	0.073 $\pm$ 0.046	0.076 $\pm$ 0.045	0.171 $\pm$ 0.043
SA-MPPI(Ours)	**0.066 $\pm$ 0.041**	**0.069 $\pm$ 0.040**	**0.167 $\pm$ 0.043**

B. Results and Analysis

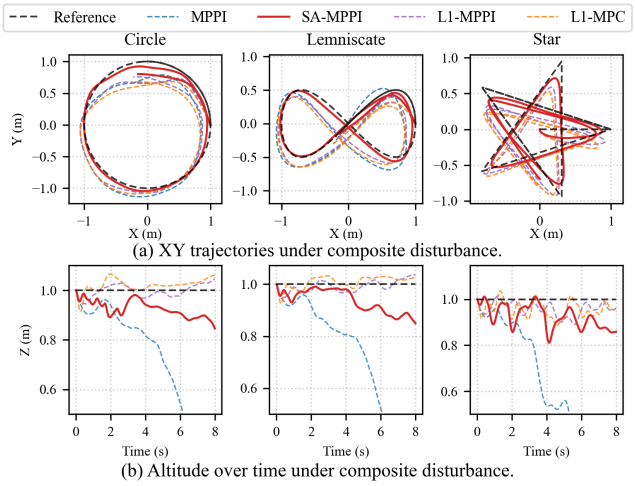
1) Nominal Condition Evaluation: We first evaluated tracking performance under nominal conditions, where only actuator noise and minor process noise were present. As shown in Fig. 4(a) and summarized in Table II, all controllers achieved satisfactory tracking accuracy in the absence of significant external disturbances. Nevertheless, SA-MPPI consistently outperformed the baselines. On the challenging star trajectory, SA-MPPI achieved a mean tracking error of 0.167 m, compared to 0.179 m for standard MPPI and 0.171 m for $\mathcal{L}_1$-MPC. This improvement is attributed to the asynchronous guidance mechanism, which generates smoother nominal trajectories for the MPPI sampler, thereby mitigating the effects of minor system uncertainties and enhancing stability.



(a) XY trajectories under nominal conditions.

(b) XY trajectories under wind disturbance.

Fig. 4: XY trajectory tracking comparison between SA-MPPI and baseline controllers. (a) Results under nominal conditions. (b) Results under wind disturbance for three reference trajectories.



(a) XY trajectories under composite disturbance.

(b) Altitude over time under composite disturbance.

Fig. 5: Controller performance under composite disturbance affecting lateral and vertical dynamics. (a) XY trajectories. (b) Altitude over time for Circle, Lemniscate, and Star references. SA-MPPI (solid red) effectively rejects disturbances, maintaining trajectory and altitude, while MPPI (dashed blue) suffers significant altitude loss.

2) Disturbance Robustness Evaluation: To assess robustness, we evaluated all controllers in two challenging scenarios: a persistent wind disturbance, and a composite disturbance combining wind with an unmodeled constant force along the body's z-axis. The quantitative results in Table III indicate that the performance gap between SA-MPPI and the baselines widens as disturbance complexity increases.

This trend is most pronounced in the composite scenario, illustrated in Fig. 5. On the aggressive star trajectory, baseline methods degraded significantly; standard MPPI failed to maintain stability and ultimately crashed. In contrast, SA-MPPI successfully completed the trajectory with a mean error of only 0.185 m–a 16% improvement over the next-best method, $\mathcal{L}_1$-MPPI (0.219 m)–while entirely avoiding catastrophic failure. This robustness stems from SA-MPPI's proactive planning: by penalizing control sequences sensitive to disturbance uncertainty, the controller preemptively selects stable, low-risk actions rather than relying on reactive corrections.

TABLE III: Quantitative results of MAE ($\pm$ std) for trajectory tracking performance of all controllers on three trajectories under wind and composite disturbances.

Trajectory	Method	Wind	Composite
Circle	MPPI	0.187 $\pm$ 0.089	0.364 $\pm$ 0.172
	$\mathcal{L}_1$-MPPI	0.185 $\pm$ 0.087	0.197 $\pm$ 0.072
	$\mathcal{L}_1$-MPC	0.221 $\pm$ 0.095	0.235 $\pm$ 0.084
	SA-MPPI(Ours)	**0.099** $\pm$ **0.051**	**0.137** $\pm$ **0.056**
Lemniscate	MPPI	0.170 $\pm$ 0.045	0.361 $\pm$ 0.197
	$\mathcal{L}_1$-MPPI	0.162 $\pm$ 0.047	0.174 $\pm$ 0.042
	$\mathcal{L}_1$-MPC	0.201 $\pm$ 0.042	0.212 $\pm$ 0.047
	SA-MPPI(Ours)	**0.089** $\pm$ **0.049**	**0.116** $\pm$ **0.058**
Star	MPPI	0.237 $\pm$ 0.083	*crash*
	$\mathcal{L}_1$-MPPI	0.233 $\pm$ 0.078	0.219 $\pm$ 0.062
	$\mathcal{L}_1$-MPC	0.260 $\pm$ 0.069	0.248 $\pm$ 0.060
	SA-MPPI(Ours)	**0.172** $\pm$ **0.052**	**0.185** $\pm$ **0.053**

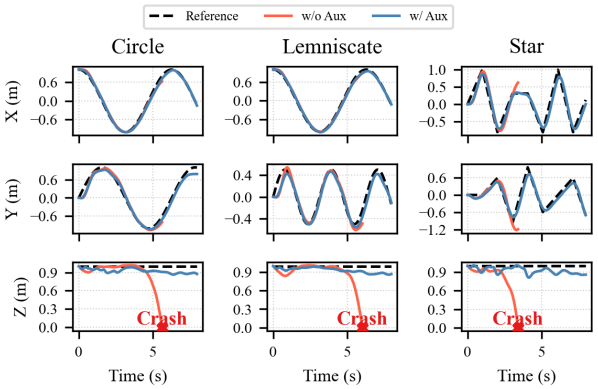


Fig. 6: Ablation of the asynchronous guidance, showing the stable SA-MPPI trajectory versus the catastrophic failure when guidance is disabled.

3) Computational Efficiency: We evaluated all controllers on an NVIDIA Jetson Orin NX to assess their computational efficiency, with results summarized in Fig. 8. Our proposed SA-MPPI achieves a mean computation time of 2.86 ms, representing a 2.35$\times$ speedup (57.5% reduction) compared to standard MPPI (6.73 ms) and a similar 2.36$\times$ speedup (57.6% reduction) over $\mathcal{L}_1$-MPPI (6.74 ms). SA-MPPI also improves upon $\mathcal{L}_1$-MPC (3.21 ms) by 10.9%. These gains stem from the asynchronous guidance architecture, which decouples the high-frequency MPPI loop from the computationally demanding auxiliary controller. By eliminating the latency bottleneck of serial informed methods and reducing the sampling requirement through improved guidance quality, SA-MPPI ensures real-time feasibility for agile quadrotor control on resource-constrained embedded systems.

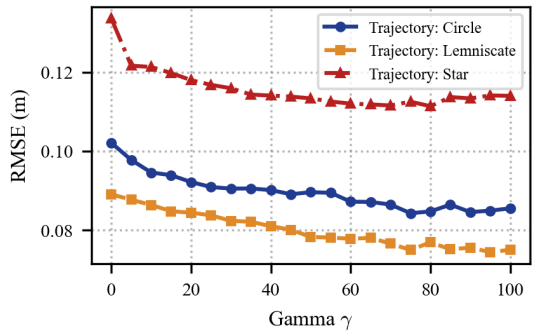


Fig. 7: Impact of the sensitivity penalty γ on RMSE under composite disturbance. Higher γ penalizes high-variance control sequences, reducing tracking error across trajectories and highlighting the importance of sensitivity for robust disturbance mitigation.

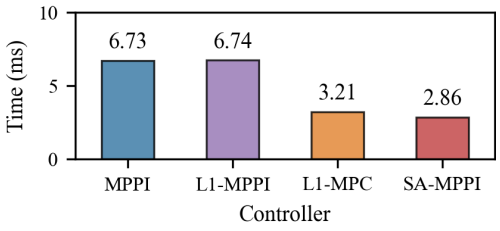


Fig. 8: Average computation time per control step on an NVIDIA Jetson Orin NX platform.

C. Ablation Experiments

To evaluate the contribution of SA-MPPI's core components, we conducted ablation studies focusing on the asynchronous auxiliary guidance and the sensitivity penalty γ.

1) Effect of Asynchronous Guidance: We first examined the role of the asynchronous guidance by comparing the full framework with a variant that excluded it. As shown in Fig. 6, removing the auxiliary guidance drastically degraded performance, resulting in unstable behavior, large tracking deviations, and frequent failures on complex trajectories. On the star trajectory, the RMS error increased by more than 200%. These results confirm that auxiliary guidance is essential for steering the MPPI sampler toward stable, high-performance control regions.

2) Effect of Sensitivity Penalty: Next, we analyzed the influence of the sensitivity penalty γ. The results in Fig. 7 indicate that penalizing cost variance (i.e., increasing γ from zero) is critical for achieving robustness, as it substantially reduced tracking errors under disturbance. This highlights the importance of explicitly accounting for sensitivity in the optimization, which enables SA-MPPI to balance robustness against performance.

D. Real-World Validation

We validated the SA-MPPI controller on a custom quadrotor (mass 772 g) equipped with an NVIDIA Jetson Orin NX for onboard computation. The optimized CUDA implementation achieves an average inference time of 3 ms, supporting a high-level control frequency of 100 Hz.

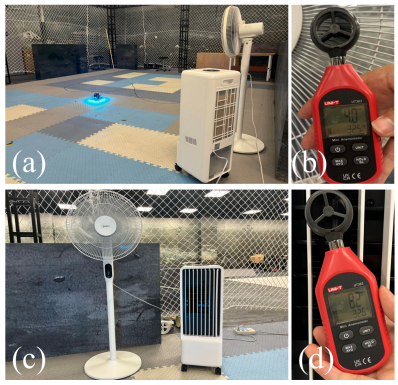


Fig. 9: Experimental setup for real-world quadrotor flights under wind disturbances. (a) Overview of the flight arena with motion capture system. (b) Wind speed measurement from fan (approx. 4.0 m/s). (c) Sources of disturbances: fan and cooler. (d) Wind speed measurement from cooler (approx. 8.2 m/s).

The resulting commands are sent to a PX4-based low-level controller via Micro Air Vehicle link (MAVLink), with state estimation provided by a 200 Hz Nokov motion capture system. A 30 Hz low-pass filter is applied to angular velocity commands to reduce sampling-induced chatter.

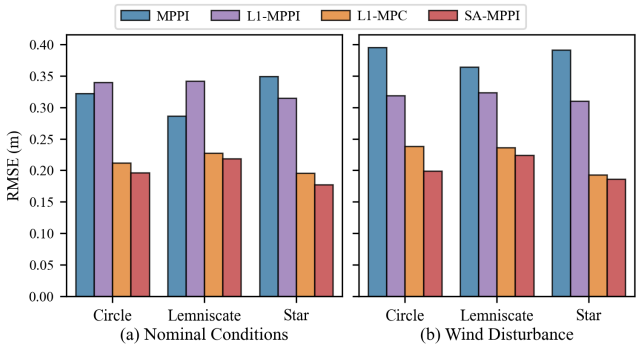
The experimental setup is shown in Fig. 9. We evaluated performance in two scenarios: a) Nominal, without disturbances, and b) Wind, where a fan and portable cooler generated a consistent airflow of ∼4.1 m/s. In both cases, the quadrotor tracked circle, lemniscate, and star trajectories used in simulation.

Results in Fig. 10 demonstrate that SA-MPPI consistently achieves the lowest RMSE, with a minimal 0.186 m on the aggressive star trajectory under wind. Standard MPPI exhibits the highest errors, while $\mathcal{L}_1$-MPC performs competitively but is generally outperformed by SA-MPPI, confirming its superior accuracy and robustness.

Analysis of rotor thrusts (Fig. 11) reveals distinct control behaviors. Standard MPPI (Fig. 11(a)) shows high-frequency, large-amplitude oscillations, indicating reactive disturbance rejection that compromises efficiency and stresses hardware. In contrast, SA-MPPI (Fig. 11(b)) produces smooth, coordinated thrusts, demonstrating proactive disturbance rejection and resulting in more stable, efficient flight.

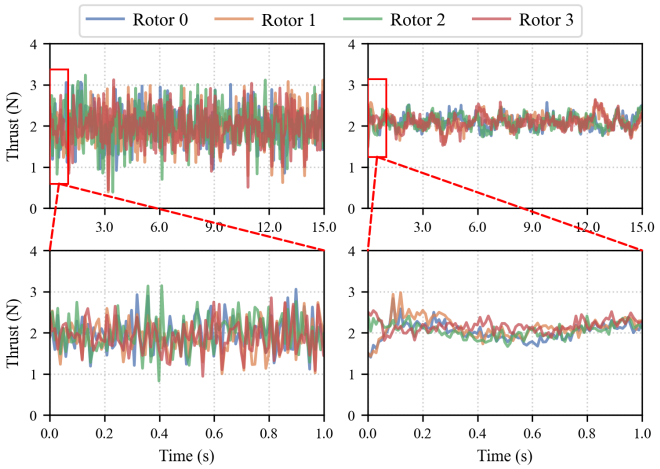
VI. CONCLUSION

We presented SA-MPPI, a novel sampling-based control framework that addresses the limitations of traditional MPPI under persistent uncertainties. The approach combines two key innovations: an asynchronous architecture that constructs an informed sampling distribution without introducing latency, and a sensitivity-aware cost metric that penalizes high-variance sequences to embed robustness directly into the optimization. This integration yields predictable trajectories, mitigates control chatter, and enables low-latency,



(a) Nominal Conditions (b) Wind Disturbance

Fig. 10: Quantitative tracking performance under (a) nominal conditions and (b) wind disturbance. The bars represent the Position RMSE (m) for four controllers across three distinct trajectories. Lower values indicate higher tracking accuracy.



(a) MPPI Rotor Forces (b) SA-MPPI Rotor Forces

Fig. 11: Rotor thrust commands for (a) standard MPPI and (b) SA-MPPI while tracking the star trajectory under wind disturbance.

agile quadrotor control in dynamic environments. Extensive simulations and hardware experiments demonstrate that SA-MPPI consistently outperforms baselines in both tracking accuracy and computational efficiency. Future work will explore extending SA-MPPI with learning-based models to address more complex unmodeled dynamics.

REFERENCES

[1] M. Abitha and A. Saleem, "Quadrotor modeling approaches and trajectory tracking control algorithms: A review," *International Journal of Robotics and Control Systems*, vol. 4, no. 1, pp. 401–426, 2024.

[2] H. Liu, X. Long, Y. Li, J. Yan, M. Li, C. Chen, F. Gu, H. Pu, and J. Luo, "Adaptive multi-uav cooperative path planning based on novel rotation artificial potential fields," *Knowledge-Based Systems*, vol. 317, p. 113429, 2025.

[3] F. Gu, J. Ai, X. Lu, X. Long, Y. Li, T. Jiang, C. Chen, and H. Liu, "Heteroscedastic bayesian optimization-based dynamic pid tuning for accurate and robust uav trajectory tracking," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025, pp. 17 428–17 435.

[4] D. Hanover, P. Foehn, S. Sun, E. Kaufmann, and D. Scaramuzza, "Performance, precision, and payloads: Adaptive nonlinear mpc for quadrotors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 690–697, 2021.

[5] J. Frey, Y. Gao, F. Messerer, A. Lahr, M. Zeilinger, and M. Diehl, "Efficient zero-order robust optimization for real-time model predictive control with acados," in *2024 European Control Conference (ECC)*. IEEE, 2024, pp. 3470–3475.

[6] J. Yin, P. Tsiotras, and K. Berntorp, "Chance-constrained information-theoretic stochastic model predictive control with safety shielding," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*. IEEE, 2024, pp. 653–658.

[7] J. Yin, Z. Zhang, and P. Tsiotras, "Risk-aware model predictive path integral control using conditional value-at-risk," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. London, United Kingdom: IEEE, May 2023, p. 7937–7943.

[8] K. Huang, R. Rana, A. Spitzer, G. Shi, and B. Boots, "Datt: Deep adaptive trajectory tracking for quadrotor control," *arXiv preprint arXiv:2310.09053*, 2023.

[9] S. Safaoui and T. H. Summers, "Distributionally robust cvar-based safety filtering for motion planning in uncertain environments," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 103–109.

[10] N. Hovakimyan and C. Cao, *L1 adaptive control theory: Guaranteed robustness with fast adaptation*. SIAM, 2010.

[11] Z. Wu, S. Cheng, P. Zhao, A. Gahlawat, K. A. Ackerman, A. Lakshmanan, C. Yang, J. Yu, and N. Hovakimyan, "L1quad: L1 adaptive augmentation of geometric control for agile quadrotors with performance guarantees," *IEEE Transactions on Control Systems Technology*, vol. 33, no. 2, pp. 597–612, 2025.

[12] J. Pravitra, K. A. Ackerman, C. Cao, N. Hovakimyan, and E. A. Theodorou, " 1-adaptive mppi architecture for robust and agile control of multirotors," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7661–7666.

[13] G. Williams, B. Goldfain, P. Drews, K. Saigol, J. M. Rehg, and E. A. Theodorou, "Robust sampling based model predictive control with sparse objective information." in *Robotics: Science and Systems*, vol. 14, 2018, p. 2018.

[14] 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.

[15] J. Yin, C. Dawson, C. Fan, and P. Tsiotras, "Shield model predictive path integral: A computationally efficient robust mpc method using control barrier functions," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7106–7113, 2023.

[16] A. Hakobyan, G. C. Kim, and I. Yang, "Risk-aware motion planning and control using cvar-constrained optimization," *IEEE Robotics and Automation letters*, vol. 4, no. 4, pp. 3924–3931, 2019.

[17] X. Cai, S. Ancha, L. Sharma, P. R. Osteen, B. Bucher, S. Phillips, J. Wang, M. Everett, N. Roy, and J. P. How, "EVORA: Deep Evidential Traversability Learning for Risk-Aware Off-Road Autonomy," *IEEE Transactions on Robotics*, vol. 40, pp. 3756–3777, 2024.

[18] X. Cai, M. Everett, J. Fink, and J. P. How, "Risk-Aware Off-Road Navigation via a Learned Speed Distribution Map," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2931–2937.

[19] T. Belvedere, M. Cognetti, G. Oriolo, and P. R. Giordano, "Sensitivity-aware model predictive control for robots with parametric uncertainty," *IEEE Transactions on Robotics*, 2025.

[20] K. Y. Chee, T. C. Silva, M. A. Hsieh, and G. J. Pappas, "Enhancing sample efficiency and uncertainty compensation in learning-based model predictive control for aerial robots," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 9435–9441.

[21] E. Trevisan and J. Alonso-Mora, "Biased-mppi: Informing sampling-based model predictive control by fusing ancillary controllers," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5871–5878, 2024.

[22] M. Minařík, R. Pěnička, V. Vonásek, and M. Saska, "Model predictive path integral control for agile unmanned aerial vehicles," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 13 144–13 151.

[23] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados – a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, 2021.