Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Variance-Reduced Model Predictive Path Integral via Quadratic Model Approximation

Fabian Schramm[1], Franki Nguimatsia Tiofack[1], Nicolas Perrin-Gilbert[2], Marc Toussaint[3] and Justin Carpentier[1]

[1]Inria and DI-ENS, PSL Research University [2]Sorbonne University [3]TU Berlin
Email: fabian.schramm@inria.fr

***Abstract*—Sampling-based controllers, such as Model Predictive Path Integral (MPPI) methods, offer substantial flexibility but often suffer from high variance and low sample efficiency. To address these challenges, we introduce a hybrid variance-reduced MPPI framework that integrates a prior model into the sampling process. Our key insight is to decompose the objective function into a known approximate model and a residual term. Since the residual captures only the discrepancy between the model and the objective, it typically exhibits a smaller magnitude and lower variance than the original objective. Although this principle applies to general modeling choices, we demonstrate that adopting a quadratic approximation enables the derivation of a closed-form, model-guided prior that effectively concentrates samples in informative regions. Crucially, the framework is agnostic to the source of geometric information, allowing the quadratic model to be constructed from exact derivatives, structural approximations (e.g., Gauss- or Quasi-Newton), or gradient-free randomized smoothing. We validate the approach on standard optimization benchmarks, a nonlinear, underactuated cart-pole control task, and a contact-rich manipulation problem with non-smooth dynamics. Across these domains, we achieve faster convergence and superior performance in low-sample regimes compared to standard MPPI. These results suggest that the method can make sample-based control strategies more practical in scenarios where obtaining samples is expensive or limited.**

I. Introduction

Formulating control, estimation, and motion planning as optimization problems has become a standard paradigm in automatic control and robotics. Traditionally, these problems are posed in their primal form: one seeks a single optimal configuration or trajectory x^* that minimizes a cost function $f(x)$. Standard approaches to solving this problem, such as gradient descent or Newton's method, rely on local first- or second-order information. While efficient, these methods assume f is smooth and can converge to local minima when f is non-convex. To overcome these limitations, an alternative viewpoint elevates the problem to the *space of probability measures*. Here, the objective is to minimize the expected cost under a distribution μ. This measure-theoretic formulation unifies several powerful frameworks in global optimization, stochastic control, and sampling-based motion planning [30, 18].

Within this distributional approach, two complementary families of methods have emerged. The first relies on convex relaxations of the infinite-dimensional measure optimization problem. Lasserre's Moment–Sum-of-Squares (SOS) hierarchy [19], and its algebraic counterpart by Parrilo [24], provides a systematic sequence of semidefinite programming (SDP) relaxations whose solutions converge to the global optimum under mild regularity assumptions. These methods offer strong theoretical guarantees but remain limited in practice as the size of the SDPs grows rapidly with the dimensionality and polynomial degree of the underlying problem. Recent extensions, such as KernelSOS [28, 11], address these limitations by replacing polynomial bases with reproducing kernels, enabling the treatment of non-polynomial objectives and infinite-dimensional function spaces. While such approaches significantly broaden the expressive power of SOS relaxations, they still require solving large-scale SDPs, which makes real-time use in high-frequency control loops impractical.

The second family originates from information-theoretic or KL-regularized formulations of stochastic optimal control. Here, the optimal control distribution is expressed as a Boltzmann distribution over trajectories, obtained by introducing an

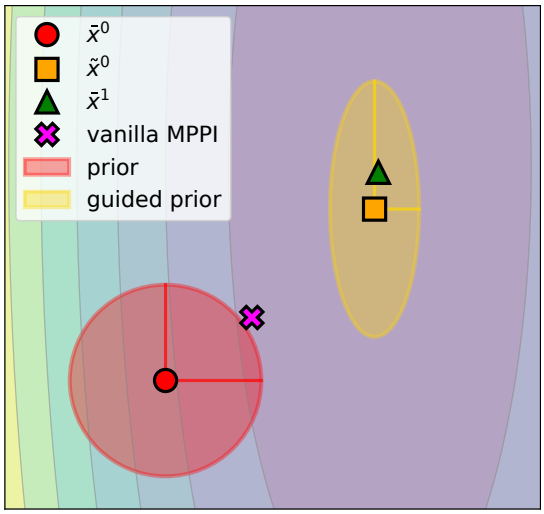


Fig. 1: **Illustration of covariance adaptation via Newton-like approximation.** Model-guided MPPI and vanilla MPPI start at the same state (circle) and use 100 samples from an isotropic Gaussian prior. The cross indicates the vanilla MPPI update. Our method exploits the gradient and Hessian at $\bar{x}^0$ to construct a quadratic model. This leads to a guided prior centered at $\tilde{x}^0$ (rectangle), concentrating the sampling distribution (yellow) along the valley and enabling an update $\bar{x}^1$ (triangle) by sampling the residual toward the optimum.

entropy or KL penalty on the control [17, 31, 30]. This reformulation naturally leads to sampling-based algorithms, most notably the Model Predictive Path Integral (MPPI) [37, 38] control algorithm. MPPI offers compelling practical advantages: it is derivative-free, accommodates non-smooth costs (e.g., contact events), and parallelizes efficiently on modern hardware accelerators. However, unlike convex relaxation methods, MPPI approximates the optimal distribution via Monte Carlo sampling, which introduces a central challenge: high estimator variance. Standard isotropic sampling fails to adapt to landscape geometry, resulting in poor sample efficiency and slow convergence in high-dimensional spaces. While recent work has shown that incorporating curvature information by reconstructing local Jacobians and performing Gauss–Newton updates can partially mitigate these effects [15], such approaches remain in a high-sample regime. Consequently, state-of-the-art implementations often rely on thousands of parallel rollouts to achieve stable performance.

Variance reduction is a well-studied topic in broader stochastic optimization. CMA-ES [13, 12] and the Cross-Entropy Method [27] adapt covariance matrices based on elite samples, while classical techniques like importance sampling [26], control variates [10], or baseline subtraction in reinforcement learning [39, 25], explicitly target variance reduction in Monte-Carlo estimators. Similar ideas appear in large-scale stochastic optimization, where variance-reduced gradient methods and curvature-aware updates are known to accelerate convergence [4].

Across domains, incorporating structural information consistently improves the efficiency of noisy estimators. This suggests that similar ideas may also benefit sampling-based control. Yet, unlike many large-scale optimization settings, real-time control imposes strict computational budgets, forcing algorithms to operate in a low-sample regime. These observations raise an important question: *Can we retain the flexibility of sampling-based control while achieving faster convergence with significantly fewer samples?*

In this work, we introduce a hybrid approach that guides sampling using a local model of the objective function, thereby accelerating the overall convergence of MPPI-based methods. Our main contributions are as follows: (i) we derive a variance-reduced MPPI framework by decomposing the objective into a known model and a residual term, (ii) we propose a specific instantiation using a quadratic model approximation, deriving a closed-form, model-guided proposal distribution, (iii) we show that this quadratic model can be computed via either exact derivatives, structural approximations, or a gradient-free randomized smoothing scheme, and (iv) we validate the approach numerically, showing superior convergence in low-sample regimes compared to standard baselines.

The remainder of this paper is organized as follows. Section II formalizes the optimization problem and the standard MPPI derivation. Section III details the derivation of the quadratic model approximation. Section IV presents the numerical results, and Section V concludes the paper.

II. Preliminaries

We consider optimization problems of the form:

$$x^* \in \arg\min_{x\in\Omega} f(x), \tag{1}$$

where $f : \Omega \to \mathbb{R}$ denotes the objective and $\Omega \subseteq \mathbb{R}$ represents the feasible set.

From a measure-theoretic perspective, the problem (1) can be reformulated as an equivalent problem over measures [19]. The goal is to find a probability distribution μ^*, minimizing the expected loss over distributions whose support lies in Ω:

$$\mu^* \in \arg\min_{\substack{\mu \\ \text{supp}(\mu)\subseteq\Omega}} \mathcal{J}(\mu) \quad \text{with} \quad \mathcal{J}(\mu) = \int_\Omega f(x)\mu(x)\,dx. \tag{2}$$

Depending on the setup, the objective function f can be convex, smooth, or arbitrarily complicated (non-convex, non-smooth). For instance, if f is strictly convex with a unique global minimum, working on (2) will converge to the optimal solution, which in the limit is a Dirac $\mu^*(x) \to \delta(x - x^*)$ concentrating the distribution mass entirely on the global optimal solution x^*.

While Problem (2) theoretically allows for finding global optima [19], solving it over the infinite-dimensional space of measures is generally intractable. Instead of a direct global solution, a common iterative approach is to perform sequential updates that penalize changes to the current distribution. At each step k, we seek a new distribution μ^{k+1} that minimizes the original objective $\mathcal{J}(\mu)$ in (2) subject to a proximity penalization to stay close to the previous estimate μ^k:

$$\mu^{k+1} = \arg\min_{\mu} \Big(\mathcal{J}(\mu) + \lambda \cdot \mathrm{KL}(\mu \,||\, \mu^k) \Big). \tag{3}$$

The analytical solution to this problem yields the closed-form Boltzmann update:

$$\mu^*(x) \propto \mu^k(x) \cdot \exp\left(-\tfrac{1}{\lambda} f(x)\right). \tag{4}$$

While μ^* is the optimal update, it is in general complex and difficult to sample from directly.

To achieve tractability, a common strategy is to constrain μ to a parametric family of distributions, typically a Gaussian $p_\theta = \mathcal{N}(\bar{x}, \Sigma)$, where $\bar{x}$ denotes the mean and Σ the covariance. The objective then becomes to identify the finite-dimensional parameters θ such that p_θ best approximates the optimal Boltzmann distribution μ^* in (4). This particular instantiation of sequential KL control with a Gaussian parametrization underlies sampling-based algorithms such as Model Predictive Path Integral (MPPI) control [30, 37], which have gained significant traction in the robotics community in recent years [16, 35, 7, 40, 41, 29]. These approaches typically rely solely on zeroth-order information (function evaluations). While such derivative-free formulations naturally handle non-smooth dynamics, they do not exploit available gradient or curvature information. As a result, these sampling-based methods often exhibit high variance and low sample efficiency in complex cost landscapes, in contrast to classical optimization

techniques — such as Newton-based methods — that can achieve second-order convergence rates near a local optimum.

In the following, we extend this framework by embedding geometric information into the MPPI formulation, enabling the sampling process to exploit local curvature and thereby achieving faster, more reliable convergence.

III. Methodology

Sampling-based control methods like MPPI offer substantial flexibility but often suffer from high variance and low sample efficiency. To address these limitations, we propose a hybrid MPPI approach that incorporates geometric information into the sampling process via a local approximation of the objective function. The key insight is to decompose the objective into an approximate model term and a residual term. The resulting model-based information biases the sampling distribution toward informative regions, reducing the variance of the underlying gradient estimators and thereby accelerating convergence relative to the classic MPPI updates. The remainder of this section details the formulation, the construction of the approximation, and stability mechanisms.

A. Model-guided distribution update

Consider the problem of finding the optimal distribution $p^*(x)$ at iteration $k+1$, given a prior estimate $p_{\theta^k}(x)$ obtained at iteration k. Following 4, the information-theoretic optimal update takes the Boltzmann form:

$$p^*(x) \propto p_{\theta^k}(x) \exp\left(-f(x)/\lambda\right). \tag{5}$$

At iteration k, we decompose the objective function f into a model m^k and a residual term r^k, such that $f(x) = m^k(x) + r^k(x)$. Here, m^k acts as a control variate intended to capture the dominant landscape geometry, while r^k accounts for unmodeled discrepancies. Substituting this decomposition into (5) yields the following factorization:

$$p^*(x) \propto \underbrace{\left[p_{\theta^k}(x) \exp\left(-m^k(x)/\lambda\right)\right]}_{\text{model-guided prior } \tilde{p}_{\theta^k}(x)} \cdot \exp\left(-r^k(x)/\lambda\right). \tag{6}$$

Equation (6) reveals a fundamental separation of concerns. The term in brackets, called model-guided prior, combines the previous prior with the explicit model to form a new intermediate distribution, which we denote as the *model-guided prior* $\tilde{p}_{\theta^k}(x)$. Rather than sampling from an uninformative prior and weighing by the full complex objective f as done classically in MPPI-based approaches, we sample from the structurally informed $\tilde{p}_{\theta^k}$ and weight only by the residual r^k.

Importantly, this formulation is valid for *any* choice of model. This generality offers a powerful degree of freedom: we can design m^k such that the resulting model-guided prior $\tilde{p}_{\theta^k}$ admits a closed-form solution. By selecting a model structure compatible with the prior, we ensure that the new sampling distribution remains analytically tractable.

Algorithm 1: MPPI with Quadratic Model Approximation

Input: Initial mean $\bar{x}^0$ and covariance Σ^0, samples N, temperature λ

for $k \leftarrow 0, 1, 2, \ldots$ **do**

```
// 1. Construct quadratic model analytically
   or via RS using Eq. (12), (13)
```

Obtain gradient g^k and hessian H^k

```
// 2. Compute guided prior using Eq. (8), (9)
```

Compute guided covariance $\tilde{\Sigma}^k$ and mean $\tilde{x}^k$

```
// 3. Sample from guided prior & evaluate
```

for $i \leftarrow 1$ **to** N **do**

Sample $x^{(i)} \sim \mathcal{N}(\tilde{x}^k, \tilde{\Sigma}^k)$

Compute residual $r^k(x^{(i)})$

end

```
// 4. Update using Eq. (11)
```

Compute weights $\tilde{w}^{(i)}$ and update mean $\bar{x}^{k+1}$

end

B. Closed-form model-guided prior via quadratic model approximation

To derive an efficient closed-form algorithm, we instantiate the general framework using Gaussian approximations. We assume the current estimate is a Gaussian distribution $p_{\theta^k}(x) = \mathcal{N}(x \,|\, \bar{x}^k, \Sigma^k)$ parametrized by $\theta^k = \{\bar{x}^k, \Sigma^k\}$. We choose m^k to be a quadratic expansion of the cost around the current mean $\bar{x}^k$:

$$m^k(x) = f(\bar{x}^k) + (g^k)^\top (x - \bar{x}^k) + \frac{1}{2}(x - \bar{x}^k)^\top H^k (x - \bar{x}^k), \tag{7}$$

where g^k and H^k denote the gradient and Hessian parameters at the expansion point, see Sec. III-C for details. Since the product of a Gaussian PDF and the exponential of a quadratic function is itself a Gaussian, the model-guided prior $\tilde{p}_{\theta^k}$ derived in (6) remains in the Gaussian family. We can therefore solve for its parameters $\tilde{\theta}^k = \{\tilde{x}^k, \tilde{\Sigma}^k\}$ analytically. The guided prior $\tilde{p}_{\theta^k}(x) = \mathcal{N}(x \,|\, \tilde{x}^k, \tilde{\Sigma}^k)$ is characterized by the updated covariance:

$$\tilde{\Sigma}^k = \left((\Sigma^k)^{-1} + \tfrac{1}{\lambda} H^k\right)^{-1}. \tag{8}$$

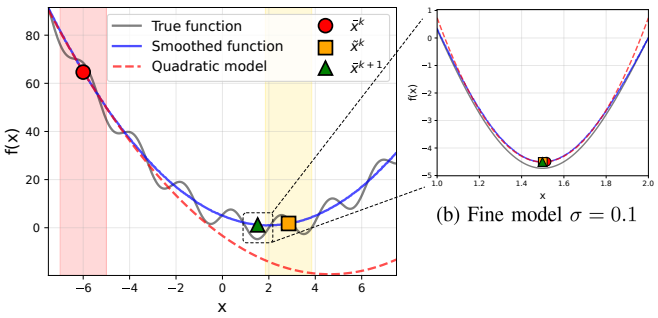
To guarantee a valid positive-definite covariance matrix $\tilde{\Sigma}^k$, appropriate regularization or convexification is applied to H^k whenever it is indefinite. Consequently, the mean is given by:

$$\tilde{x}^k = \tilde{\Sigma}^k \left((\Sigma^k)^{-1} \bar{x}^k + \tfrac{1}{\lambda}\left(H^k \bar{x}^k - g^k\right)\right). \tag{9}$$

The final optimal distribution is then obtained by reweighting this model-guided Gaussian $\tilde{p}_{\theta^k}$ with the residual:

$$p^*(x) \propto \underbrace{\mathcal{N}(x \,|\, \tilde{x}^k, \tilde{\Sigma}^k)}_{\tilde{p}_{\theta^k}(x)} \cdot \exp\left(-r^k(x)/\lambda\right). \tag{10}$$

Interestingly, this reformulation naturally lends itself to applying MPPI on the residual function r^k while sampling



(a) Coarse model $\sigma = 1.0$

(b) Fine model $\sigma = 0.1$

Fig. 2: **Comparison coarse vs fine model.** (a) A large smoothing kernel $\sigma = 1.0$ suppresses sinusoidal disturbances (grey), yielding a coarse approximation (blue) capturing global geometry. The resulting quadratic model (red) generates a proposal $\tilde{x}^1$ moving toward the global minimum despite local nonconvexities. (b) Near the optimum, reducing the scale to $\sigma = 0.1$ captures the local curvature for final convergence.

trajectories $\{x^{(i)}\}_{i=1}^N$ from the model-guided prior $\tilde{p}_{\theta^k}$. The update is now given by:

$$\bar{x}^{k+1} \approx \sum_{i=1}^{N} \tilde{w}^{(i)}\, x^{(i)}, \text{ with } \tilde{w}^{(i)} = \frac{\exp\left(-r^k(x^{(i)})/\lambda\right)}{\sum_{j=1}^{N} \exp\left(-r^k(x^{(j)})/\lambda\right)}. \tag{11}$$

Compared to vanilla MPPI, which samples directly from p_θ^k, this model-guided approach fundamentally alters the update mechanism. First, the analytical shift to $\tilde{p}_{\theta^k}$ integrates geometry information, blending the prior mean $\bar{x}^k$ with a Newton-like step derived from the model m^k. Second, the sampling process is reserved for exploring the residual error r^k around this new center, rather than exploring the full cost f. This procedure is summarized in Algorithm 1.

C. Quadratic model approximations

To implement the update rules above, we require g^k and H^k to construct the model (7). A strength of the proposed framework is that it decouples the sampling mechanism from the source of geometric information, allowing the practitioner to select the most appropriate approximation strategy based on the available computational budget and problem structure.

Analytical second-order approximations. If f is differentiable and smooth, g^k and H^k can be exact analytical derivatives. In this exact *local model*, the resulting guided prior $\tilde{p}_{\theta^k}$ acts as a precise local Newton step. This maximizes convergence speed near the optimum but risks guiding the sampling distribution into shallow local minima if the current estimate is far from the global solution.

Gauss-Newton approximations. Beyond exact derivatives, our framework accommodates strategies that exploit specific problem structures. For instance, many control objectives are formulated as non-linear least-squares problems: $f(x) = \frac{1}{2}\|R(x)\|^2$, where residual $R(x)$ stacks error terms such as state deviations and control penalties. In such cases, the Hessian can be approximated via the Gauss-Newton approximation as $H^k \approx J(x)^\top J(x)$, where $J(x)$ is the Jacobian of $R(x)$. This avoids computing the full Hessian while ensuring a positive-semidefinite approximation that seamlessly integrates into our quadratic model update. Recent work [15] can be viewed as an instantiation of this principle, in which curvature information is reconstructed from black-box residuals via randomized smoothing.

Quasi-Newton approximations. Similarly, when second-order information is prohibitively expensive or unavailable, one can employ Quasi-Newton methods such as BFGS or L-BFGS [23]. These techniques iteratively build a curvature estimate H^k from only a history of gradient updates, enabling our model-guided framework to be applied in settings where only first-order derivatives are available.

Randomized smoothing approximations. In the most general case where f is non-smooth or black-box, we employ Randomized Smoothing (RS) [3, 8, 1] to estimate the model parameters. This allows us to recover zeroth-order estimators for the gradient and Hessian using only function evaluations. While RS is a general framework compatible with various smoothing distributions, we use a Gaussian kernel in this work. This defines the smoothed surrogate as the Gaussian convolution $f_\sigma(x) = \mathbb{E}_{z\sim\mathcal{N}(0,\sigma^2 I)}[f(x+z)]$. By exploiting Stein's identity and utilizing a centered control variate to reduce variance, we obtain the following Monte Carlo estimators for the gradient g^k and Hessian H^k at the current mean $\bar{x}^k$:

$$g^k \approx \frac{1}{M\sigma^2} \sum_{j=1}^{M} \left(f(\bar{x}^k + z_j) - f(\bar{x}^k)\right) z_j, \tag{12}$$

$$H^k \approx \frac{1}{M\sigma^4} \sum_{j=1}^{M} \left(f(\bar{x}^k + z_j) - f(\bar{x}^k)\right) (z_j z_j^\top - \sigma^2 I), \tag{13}$$

where $\{z_j\}_{j=1}^M$ are samples drawn from $\mathcal{N}(0, \sigma^2 I)$.

With a small noise scale ($\sigma \to 0$), the smoothed surrogate f_σ closely approximates the original objective f. The estimators (12), (13) capture fine-scale curvature information, behaving similarly to analytical derivatives. We refer to this as the *fine model* (Fig. 2b). As σ increases, RS aggregates objective information over larger neighborhoods around the current iterate. The gradient and Hessian are based on global structure rather than local details, allowing the algorithm to step over small obstacles that might trap a local optimizer. We refer to this as the *coarse model* (Fig. 2a). The induced quadratic approximation biases the guided prior toward regions that appear promising under wider exploratory variations, encouraging exploration while filtering out fine local structure.

D. Incremental update formulation

Although formulated above in terms of full iterations, the update mechanism can equivalently be expressed in terms of incremental steps. Consider a perturbation ϵ such that $x = \bar{x}^k + \epsilon$. A second-order expansion of the objective yields

$$f(\bar{x}^k + \epsilon) = f(\bar{x}^k) + (g^k)^\top \epsilon + \frac{1}{2}\epsilon^\top H^k \epsilon + r^k(\bar{x}^k + \epsilon), \tag{14}$$

where the quadratic terms define the local model m^k and r^k accounts for unmodeled effects. As in Sec. III-B, combining the quadratic model with the k-th Gaussian exploration prior $p(\epsilon) = \mathcal{N}(0, \Sigma_{0k})$ induces a guided Gaussian distribution

$$\tilde{p}_{\theta^k}(\epsilon) = \mathcal{N}(\epsilon \,|\, \tilde{\delta}^k, \tilde{\Sigma}^k), \tag{15}$$

with parameters given by

$$\tilde{\Sigma}^k = \left(\Sigma_{0k}^{-1} + \tfrac{1}{\lambda} H^k\right)^{-1}, \tag{16}$$

$$\tilde{\delta}^k = -(\lambda \Sigma_{0k}^{-1} + H^k)^{-1} g^k, \tag{17}$$

where $\tilde{\delta}^k$ represents the model-guided incremental update step mean. As before, we regularize H^k if necessary to ensure $\tilde{\Sigma}^k$ is positive-definite. The optimal update step distribution is then obtained by reweighting this guided Gaussian by the residual,

$$p^*(\epsilon) \propto \tilde{p}_{\theta^k}(\epsilon) \exp\left(-r^k(\bar{x}^k + \epsilon)/\lambda\right). \tag{18}$$

This is the direct update-step analogue of the iterate-level distribution in (10). Yet, this formulation choice exposes the prior covariance Σ_{0k} as a tunable parameter in the algorithm design, a feature that will be exploited in the next section III-E to adequately control the variance.

A natural strategy is to perform the parameter update along the mean of this optimal distribution:

$$\Delta \bar{x}^k = \mathbb{E}_{p^*}[\epsilon]. \tag{19}$$

Since p^* is available only up to a normalizing constant, we estimate this expectation via self-normalized importance sampling using $\tilde{p}_{\theta^k}$ as the proposal. We first express the expectation as a ratio:

$$\Delta \bar{x}^k = \frac{\mathbb{E}_{\tilde{p}_{\theta^k}}\left[\epsilon \, \exp\left(-r^k(\bar{x}^k + \epsilon)/\lambda\right)\right]}{\mathbb{E}_{\tilde{p}_{\theta^k}}\left[\exp(-r^k(\bar{x}^k + \epsilon)/\lambda)\right]}. \tag{20}$$

Drawing samples $\{\epsilon_i\}_{i=1}^N \sim \mathcal{N}(\tilde{\delta}^k, \tilde{\Sigma}^k)$ from the guided prior, we obtain the Monte Carlo approximation:

$$\Delta \bar{x}^k \approx \sum_{i=1}^{N} w_i \, \epsilon_i, \text{ with } w_i = \frac{\exp\left(-r^k(\bar{x}^k + \epsilon_i)/\lambda\right)}{\sum_{j=1}^{N} \exp(-r^k(\bar{x}^k + \epsilon_j)/\lambda)}. \tag{21}$$

Finally, the mean parameter is updated as:

$$\bar{x}^{k+1} \leftarrow \bar{x}^k + \Delta \bar{x}^k. \tag{22}$$

E. Stability and variance control of the model-guided prior

While the update rules in (8) and (16) directly incorporate curvature, they can lead to variance collapse since they can only decrease variance. In regions where the Hessian H^k has large positive eigenvalues, the resulting guided covariance $\tilde{\Sigma}^k$ may shrink excessively, hindering the exploration required to resolve the residual r^k. To mitigate this, we use regularization strategies that ensure stability and maintain a minimum exploration width.

Polyak/Juditsky averaging. To smooth the evolution of the sampling distribution and prevent abrupt jumps caused by noisy gradient or Hessian estimates, we apply exponential moving averages to the model-guided parameters. Instead of using the raw guided mean $\tilde{\delta}^k$ and covariance $\tilde{\Sigma}^k$ directly for the current step, we update the sampling parameters via an exponential moving average with smoothing factor $\alpha_\delta, \alpha_\Sigma \in [0, 1]$:

$$\bar{\delta}^k = \alpha_\delta \tilde{\delta}^k + (1 - \alpha_\delta)\bar{\delta}^{k-1}, \tag{23}$$

$$\bar{\Sigma}^k = \alpha_\Sigma \tilde{\Sigma}^k + (1 - \alpha_\Sigma)\bar{\Sigma}^{k-1}. \tag{24}$$

Variance control. While the above averaging acts as a temporal filter, it does not guarantee a lower bound on the variance. To strictly prevent collapse, we use a scaling strategy that regulates the distribution's magnitude while preserving its geometry. By applying a scalar gain, we maintain the anisotropic structure (eigenvectors and relative aspect ratios) derived from the Hessian, ensuring the sampling ellipsoid remains aligned with the landscape curvature even when inflated to a safe minimum volume.

We derive the required input noise σ_{0k}^2 by analyzing the covariance update in the principal directions of the local curvature. Starting from the update rule in Eq. (16) and assuming an isotropic prior covariance $\Sigma_{0k} = \sigma_{0k}^2 I$, the inverse of the updated covariance becomes:

$$(\tilde{\Sigma}^k)^{-1} = \frac{1}{\sigma_{0k}^2} I + \frac{1}{\lambda} H^k. \tag{25}$$

Since H^k is symmetric, we can analyze the update along its principal axes. Let κ_i denote the eigenvalues of H^k. The updated covariance $\tilde{\Sigma}^k$ shares same eigenbasis, with its eigenvalues $\tilde{\sigma}_i^2$ given by the reciprocal sum:

$$\frac{1}{\tilde{\sigma}_i^2} = \frac{1}{\sigma_{0k}^2} + \frac{\kappa_i}{\lambda} \quad \Rightarrow \quad \tilde{\sigma}_i^2 = \frac{\lambda \, \sigma_{0k}^2}{\lambda + \sigma_{0k}^2 \, \kappa_i}. \tag{26}$$

We see that the variance is most compressed in the direction of maximum positive curvature. To prevent excessive narrowing, we enforce that the variance in this "worst-case" direction, associated with the maximum eigenvalue $\kappa_{\max}(H^k)$, must be at least a target threshold σ_{target}^2:

$$\frac{\lambda \, \sigma_{0k}^2}{\lambda + \sigma_{0k}^2 \, \kappa_{\max}} \geq \sigma_{\text{target}}^2, \tag{27}$$

yielding the lower bound:

$$\sigma_{0k}^2 \geq \frac{\lambda \, \sigma_{\text{target}}^2}{\lambda - \sigma_{\text{target}}^2 \, \kappa_{\max}}. \tag{28}$$

This ensures that the sampling distribution maintains a desired width in the most constrained direction, thereby guaranteeing that variance in all other directions exceeds the threshold without distorting the curvature information.

IV. Experiments

We investigate how incorporating a quadratic model into MPPI affects convergence speed, solution quality, and robustness in low-sample regimes. In particular, we study how the *locality* and *fidelity* of the quadratic model approximation m^k influence performance. To isolate specific algorithmic properties, we consider problems with increasing difficulty. We begin

Function	Model-Guided MPPI (ours)	Vanilla MPPI	CMA-ES
Rosenbrock	6.9 ± 2.5 —	21.4 ± 7.3 —	12.0 ± 1.3 —
Styblinski–Tang	2.1 ± 0.4 —	9.2 ± 0.8 (14)	7.8 ± 2.9 —
Rastrigin	3.0 ± 1.1 —	6.9 ± 6.4 —	5.5 ± 4.6 —
Ackley	3.3 ± 1.6 (1)	3.8 ± 2.1 —	3.5 ± 1.4 —

TABLE I: Mean iterations $\pm$ std (failures) over 100 seeds.

with static optimization benchmarks to validate the core optimization mechanism. Then, we examine continuous control in the context of cart-pole control, a smooth but underactuated, nonlinear dynamical system, exploiting analytical derivatives and assessing the performance of Gauss-Newton and Quasi-Newton approximations. Finally, we aim to evaluate robustness to non-smooth dynamics (Single-Finger Sphere Manipulation) in contact-rich scenarios using randomized smoothing.

As baselines, we consider (i) vanilla MPPI, which samples from a fixed Gaussian prior and evaluates trajectories using the objective f, without exploiting any explicit local structure, and (ii) CMA-ES, implemented using the `pycma` Python library [14], a state-of-the-art derivative-free optimizer that adapts a full covariance matrix. We compare this against several model-guided MPPI variants that adopt the incremental update formulation and differ only in how the quadratic approximation in (7) is constructed.

Our experiments highlight three different key effects of model guidance: (i) accelerated and more accurate convergence when reliable analytical information is available, (ii) increased robustness in low-sample regimes, where purely sampling-based methods suffer from weight degeneracy, and (iii) the complementary roles of coarse and fine quadratic models in capturing global structure versus local precision.

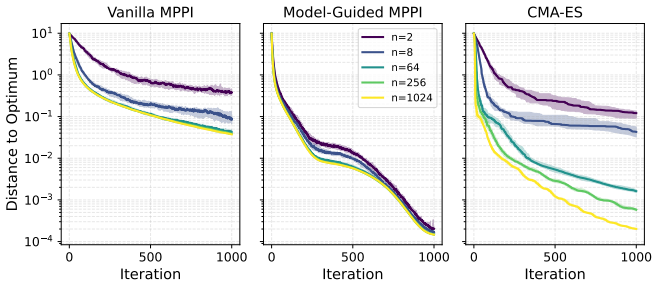
A. Illustration of quadratic model guidance

We first show the mechanism on a 2D narrow-valley objective illustrated in Fig. 1. This visualizes how second-order information reshapes the sampling distribution to align with the cost geometry. By effectively performing a Newton-like update for the distribution mean, Model-Guided MPPI concentrates samples in high-probability regions.

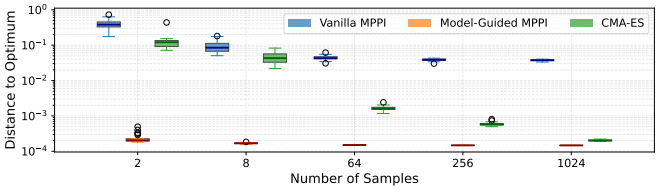
To evaluate sample quality, we use the effective sample size (ESS) [22], a standard diagnostic for importance sampling that quantifies the weight degeneracy induced by the sampling distribution. Given normalized importance weights w_i, the ESS is defined as:

$$w_i = \frac{\exp(-f_i/\lambda)}{\sum_j \exp(-f_j/\lambda)}, \quad \text{ESS} = \frac{1}{\sum_i w_i^2}. \tag{29}$$

The ESS measures the number of effectively contributing samples; larger values indicate lower variance in the importance weights. In the context of MPPI, a higher ESS corresponds to more informative samples. Using the same parameters ($N = 100$, $\lambda = 0.1$, $\sigma = 0.2$) in Fig. 1, vanilla MPPI collapses to an ESS of 1.1, indicating severe weight degeneracy. In contrast, our method maintains an ESS of 23.4, demonstrating efficient coverage of the relevant state space.



(a) **Convergence speed.** Median distance to the Newton-optimal solution over iterations. Shaded areas indicate the interquartile range (IQR) over 20 seeds.



(b) **Variance analysis in low-sample regime.** Box-plot of the final distance to the optimum after 1000 iterations. The log-scale highlights the variance and error increase for baselines when $N < 64$.

Fig. 3: **Cart-pole swing-up results**. Comparison of convergence behavior and variance across sample budgets $N \in \{2, \ldots, 1024\}$ and 20 random seeds.

B. Performance with analytical gradients

We first validate the method in a setting where exact gradient and Hessian information are available. Here, the quadratic model m^k accurately captures local curvature for smooth functions, allowing us to isolate and analyze the benefits of the update rule from errors in gradient estimation.

Static benchmarks. Tab. I reports the number of iterations required by each method to reach the global optimum within a specified infinity-norm distance using 100 samples. Initializations were set to $\bar{x}^0 = (0, 0)$ (Rosenbrock, Styblinski–Tang), $(2.0, 2.0)$ (Ackley), and $(1.9, 1.7)$ (Rastrigin). The results confirm the efficacy of the model guidance: our framework consistently converges faster than the baselines and, crucially, exhibits lower variance across the 100 random seeds. For instance, on the Rastrigin benchmark, we achieve a tight convergence distribution (3.0 ± 1.1 iterations), whereas Vanilla MPPI (6.9 ± 6.4) and CMA-ES (5.5 ± 4.6) display higher volatility, reflecting their sensitivity to the stochasticity of the sampling process.

Continuous control. To assess our method on a continuous control problem, we consider a nonlinear, underactuated cart-pole swing-up task with a horizon length of 2.5 s and a time step of 50 ms. The objective is formulated as a finite-horizon trajectory optimization problem over a sequence of continuous controls. We leverage open-source tools to implement the cart-pole dynamics and cost function using the Pinocchio library [6, 5], and CasADi [2]. This provides access to exact

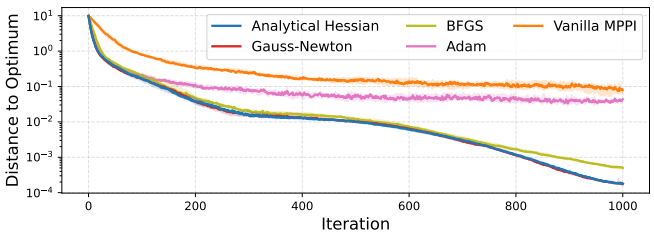


Fig. 4: **Convergence comparison of Hessian approximations on the cart-pole swing-up task.** We compare dense curvature models (Analytical, Gauss-Newton, BFGS), against a diagonal approximation (Adam) and the isotropic baseline (Vanilla MPPI). The median distance to the reference optimum is plotted across 10 random seeds, with shaded areas indicating the interquartile range (IQR).

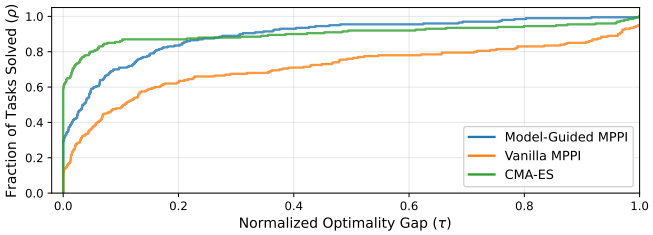


Fig. 5: **Performance profile on Single-Finger Sphere Manipulation.** Comparison across 200 randomized tasks.

analytical expressions for the trajectory cost, its gradient, and Hessian with respect to the control sequence through automatic differentiation. As a reference for the true optimum over the given horizon and discretization, we compute a locally optimal control sequence using a Newton solver with a line search.

We evaluate convergence behavior across a wide range of sampling budgets $N \in \{2, \dots, 1024\}$ in Fig. 3. In the high-sample regime ($N = 1024$), both CMA-ES and Model-Guided MPPI converge rapidly to the optimum ($< 10^{-4}$ error), whereas Vanilla MPPI requires significantly more iterations. However, the baselines are sensitive as the number of samples is reduced. Vanilla MPPI exhibits increasingly slow convergence and high variance, while CMA-ES becomes unstable below $N = 64$. In contrast, our method demonstrates sample invariance by leveraging an analytical quadratic model to effectively decouple performance from the sampling budget. As shown in Fig. 3a, the convergence trajectories from $N = 2$ to 1024 collapse onto a single curve. Fig. 3b further quantifies this robustness: while the baselines suffer from large variance in the low-sample regime, Model-Guided MPPI maintains negligible variance and consistent high-precision convergence even with as few as $N = 2$ samples. This confirms that when reliable second-order information is available, the quadratic model successfully guides optimization, reducing reliance on massive parallel rollouts.

Hessian approximations. While the previous results leverage exact analytical Hessians, we now investigate the impact of approximation quality. Following the strategies in Sec. III-C, we compare the exact local model against Gauss-Newton (GN), iterative BFGS approximations, and an Adam-based approximation (interpreting the second moment as a diagonal Hessian), alongside a vanilla MPPI baseline on the same cart-pole task using 8 samples. As shown in Fig. 4, the fidelity of the curvature information is critical. While the Adam-based model-guidance improves upon the baseline, its diagonal approximation proves insufficient to achieve high precision. In contrast, variants that capture dense curvature information, such as exact, GN, or BFGS, achieve high-precision convergence. Notably, the GN approximation exploits the least-squares structure of the cost and closely matches the analytical Hessian. The BFGS variant, despite relying solely on iterative gradient updates, maintains a similar trajectory for most of the optimization, with only a minor performance gap emerging as it approaches the higher-precision regime ($< 10^{-3}$).

C. Coarse and fine quadratic model approximations

We next evaluate how the locality of the quadratic model, controlled by the smoothing scale, affects performance on nonconvex objectives. To build intuition, we consider a one-dimensional illustrative example in Fig. 2, in which the convex objective is corrupted by sinusoidal noise. Here, the trade-off becomes visible: a coarse model (large σ) filters out high-frequency perturbations to recover the dominant global geometry. This prevents the optimizer from being trapped in local minima and guides exploration toward the correct basin of attraction. Once the iterate is in the vicinity of the optimum, a fine model (small σ) is used to resolve accurate local curvature for high-precision convergence.

This intuition extends to higher-dimensional challenges, such as the Rastrigin function with initial guess $(1.9, 1.7)$. By starting with a large noise kernel ($\sigma = 1.5$), Model-Guided MPPI discovers the global basin of attraction around $(0, 0)$. Transitioning to a finer kernel ($\sigma = 0.1$), then employs a Newton-like update step to achieve high-precision optimization in just 3 iterations with 100 samples.

D. Randomized smoothing on non-smooth dynamics

To demonstrate the versatility of our framework, we apply it to a challenging contact-rich manipulation task. Contact dynamics introduce discontinuities that make standard analytical gradients uninformative. In this setting, we use randomized smoothing to extract quadratic model-guidance for a black-box physics simulator.

Benchmark task. We use a single-finger and sphere manipulation scenario as a minimal proxy for contact-rich manipulation. The system comprises two spheres in a static environment with walls, as shown in Fig. 6. While geometrically simple, this task captures the core complexity of manipulation planning: non-smooth dynamics and underactuation. The actuated ”finger” (blue sphere) must navigate to a target while manipulating a passive object

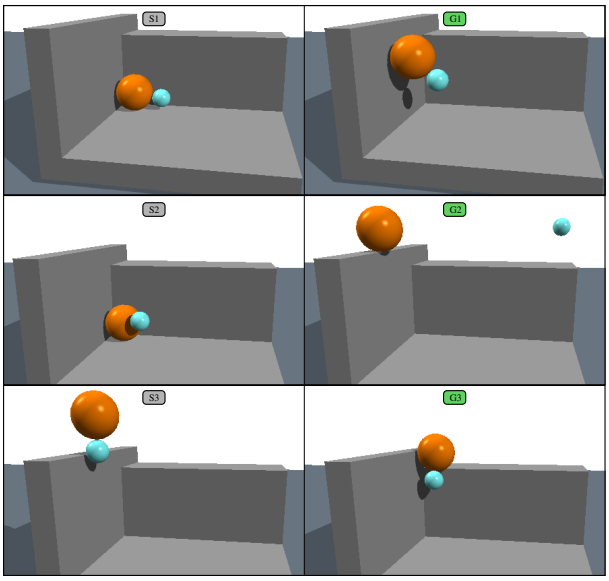


Fig. 6: **Single-Finger Sphere Manipulation.** Three different starting states (left) and corresponding goal states (right) obtained via NLP sampling illustrate varying degrees of task difficulty. The task requires the actuated finger (blue) to navigate from its start to a target configuration, while simultaneously manipulating the passive object (orange) from its initial position to a specified goal pose.

(orange sphere), requiring the optimizer to manage frequent contact-making and breaking to transfer momentum.

Experimental setup. This task is implemented within the `robotic` library [33], a framework for manipulation planning and constrained optimization which interfaces with the `MuJoCo` physics engine [32]. The simulation handles system dynamics, including friction and contact impulses at a timestep of 1ms. We parameterize the robot's motion as a spline trajectory defined by 4 control points over a fixed horizon of 1.0s, where the optimization variables are the spline control points.

Evaluation protocol. We generate 200 diverse problem instances with challenging contact configurations via NLP sampling [34] and compare methods using performance profiles of the normalized optimality gap after a fixed budget of 30 iterations. We enforce a fixed planning budget of 64 trajectory samples per iteration for all methods. For our model-guided approach, we use randomized smoothing with 128 auxiliary samples to estimate the quadratic model approximation. We treat this estimation step as a surrogate for an analytical gradient oracle. It represents the cost of obtaining model information in a non-differentiable setting, while the planning update itself respects the same sample constraint as the baselines.

Metric and results. Since the global optimum is unknown, we define the best solution f^* as the minimum cost found by any method for a given task. The normalized optimality gap is then defined as $\tau = (f_{\text{final}} - f^*)/(f_{\text{init}} - f^*)$. A value of $\tau = 0$ indicates that the method matched the best-performing method, while $\tau = 1$ implies no improvement.

The performance profile, depicted in Fig. 5, reports the fraction of tasks where a method achieves a solution within a threshold τ of the best performance. Model-Guided MPPI consistently achieves low normalized optimality gaps across a large fraction of problem instances, indicating reliable convergence, whereas Vanilla MPPI exhibits higher variability. CMA-ES demonstrates competitive performance in most instances but shows slightly reduced consistency across the full task distribution. While this benchmark serves as a first step, the results confirm that the proposed formulation can handle the hybrid dynamics inherent to rich-contact manipulation scenarios. This suggests a promising path toward applying variance-reduced MPPI to full-scale dexterous manipulation tasks in future work.

TABLE II: Per-iteration timings $\pm$ std over 10 seeds using N samples and percentages denote share of total time. Static benchmark (SB) uses analytical derivatives, cart-pole (CP) uses AD-based exact derivatives and non-smooth single-finger (SF) manipulation uses randomized smoothing.

Task - (N)	Model Construction [ms]		Sampling r^k or f [ms]	
SB - (100)	0.2 ± 0.0	(12%)	1.4 ± 0.2	(88%)
CP - (2)	3.7 ± 0.5	(75%)	1.2 ± 1.8	(25%)
CP - (8)	—	(70%)	1.5 ± 3.5	(30%)
CP - (64)	—	(52%)	3.6 ± 2.1	(48%)
CP - (256)	—	(26%)	10.8 ± 3.0	(74%)
CP - (1024)	—	(9%)	39.5 ± 4.5	(91%)
SF - (64)	1412.1 ± 156.6	(68%)	706.4 ± 78.1	(32%)

E. Computational analysis

Finally, we examine how model construction and residual sampling contribute to the per-iteration cost across our three setups. Vanilla MPPI incurs the same sampling cost but evaluates the full objective f rather than the residual r^k, so the overhead of our method is entirely concentrated in model construction, whose magnitude depends on the chosen approximation strategy.

Tab. II reports per-iteration timings, measured on an Intel Core i7-11850H. (i) Static benchmarks (SB): model construction is negligible (12% vs. 88%) since derivatives are available in closed form. The timings are averaged over the four functions. (ii) Cart-pole (CP): using AD-based exact derivatives, model construction cost is constant and independent of the sample budget N, while residual sampling grows linearly. The model share consequently decreases from 75% at $N = 2$ to 9% at $N = 1024$. We use AD here to validate the variance-reduction principle under an exact model, not for wall-clock efficiency. (iii) Single-finger (SF): model construction dominates (68% vs. 32%), since we use 128 auxiliary samples (twice the planning budget) for randomized smoothing.

Overall, the method is most attractive when the quadratic model is already available or cheap relative to rollouts. When it must be estimated from scratch, the trade-off becomes a

balance between the additional estimation cost and the gain in sample efficiency. The framework remains agnostic to this choice, supporting cheaper structured approximations such as Gauss–Newton or BFGS that closely track the exact model (Fig. 4) at a fraction of the cost.

V. Discussion and Conclusion

In this work, we introduced a variance-reduced, accelerated extension of MPPI based on a model–residual decomposition of the objective. By factoring the Boltzmann update into a model-guided prior and a residual correction, the method injects structural information into the sampling process while retaining the flexibility of sampling-based control. When instantiated with a quadratic model, this decomposition yields a closed-form Gaussian guided prior whose mean and covariance encode first- and second-order information, resulting in a Newton-like update at the distribution level. A key strength is that the framework is agnostic to the source of curvature information, seamlessly accommodating analytical Hessians, structural approximations (e.g., Gauss- or Quasi-Newton), and gradient-free randomized smoothing. Across benchmark optimization problems, nonlinear, underactuated, and non-smooth control tasks, the proposed method demonstrates faster convergence, a higher effective sample size, and greater robustness in low-sample regimes compared to vanilla MPPI and CMA-ES. These results show that exploiting model structure can improve the efficiency of sampling-based control under tight computational budgets.

While our method improves sample efficiency, it introduces some trade-offs. First, constructing the quadratic model incurs an additional computational cost (see Tab. II). The magnitude of this overhead depends on the approximation strategy. With the development of fast differentiable physics engines [36, 9, 20, 21], obtaining first-order gradients is becoming increasingly negligible, effectively mitigating the cost of analytical or Gauss-Newton approximations. In non-differentiable settings that require randomized smoothing, the overhead is distinct. While these evaluations are fully parallelizable, they incur a per-iteration computational cost that exceeds that of Vanilla MPPI. Second, the impact of the quality of the quadratic approximation on performance can be quantified through the residual. When the model m^k accurately captures the local geometry, the residual r^k has smaller magnitude and lower variance than f, the importance weights $\exp(-r^k/\lambda)$ are more uniform, and the effective sample size (ESS, Eq. (29)) increases. This is the regime in which our method provides the largest benefit. Conversely, when the model is less informative (poor curvature estimates or mismatched smoothing scale), r^k becomes larger and more variable, the weights concentrate, and ESS drops. The sampler then corrects for a poor proposal rather than exploiting a good one, and performance degrades toward vanilla MPPI. As an illustration, in Fig. 2, a too local model would guide the proposal toward the nearest valley, leading to limited benefits or oscillatory behavior. An appropriately coarse model instead smooths out local irregularities and steers iterations toward the correct basin before switching to a finer model for high-precision convergence.

We will focus on reducing computational overheads and expanding the applicability of our model-guided MPPI. We plan to investigate adaptive smoothing strategies in which the noise scale σ and temperature λ are adapted online based on the residual variance, thereby removing the need for manual heuristic tuning. Furthermore, we aim to integrate learned models using data-driven priors to guide efficient sampling and to scale this method to high-dimensional, contact-rich domains such as dexterous manipulation, where the cost of exploration is prohibitively high.

Acknowledgments

This work has received support from the French government, managed by the National Research Agency, under the France 2030 program with the references Organic Robotics Program (PEPR O2R), "PR[AI]RIE-PSAI" (ANR-23-IACL-0008) and RODEO (ANR-24-CE23-5886). The European Union also supported this work through the ARTIFACT project (GA no.101165695) and the AGIMUS project (GA no.101070165). The Paris Île-de-France Région also supported this work in the frame of the DIM AI4IDF. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the funding agencies.

References

[1] Jacob Abernethy, Chansoo Lee, and Ambuj Tewari. *Perturbation Techniques in Online Learning and Optimization*, pages 233–264. 12 2016. ISBN 9780262337939. doi: 10.7551/mitpress/10761.003.0009.

[2] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1): 1–36, 2019. doi: 10.1007/s12532-018-0139-4.

[3] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9508–9519, 2020.

[4] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *ArXiv*, abs/1606.04838, 2016.

[5] Justin Carpentier, Florian Valenza, Nicolas Mansard, et al. Pinocchio: fast forward and inverse dynamics for poly-articulated systems. https://stack-of-tasks.github.io/pinocchio, 2015–2021.

[6] Justin Carpentier, Guilhem Saurel, Gabriele Buondonno, Joseph Mirabel, Florent Lamiraux, Olivier Stasse, and Nicolas Mansard. The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives. In *IEEE International Symposium on System Integrations (SII)*, 2019.

[7] Pietro Noah Crestaz, Ludovic De Matteis, Elliot Chane-Sane, Nicolas Mansard, and Andrea Del Prete. Td-cd-mppi: Temporal-difference constraint-discounted model predictive path integral control. *IEEE Robotics and Automation Letters*, 11(1):498–505, 2026. doi: 10.1109/LRA.2025.3632612.

[8] John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012. doi: 10.1137/110831659.

[9] Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax - a differentiable physics engine for large scale rigid body simulation. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/d1f491a404d6854880943e5c3cd9ca25-Paper-round1.pdf.

[10] Paul Glasserman. Monte carlo methods in financial engineering. 2003.

[11] Antoine Groudiev, Fabian Schramm, Éloïse Berthier, Justin Carpentier, and Frederike Dümbgen. Sampling-based global optimal control and estimation via semidefinite programming, 2025. URL https://arxiv.org/abs/2507.17572.

[12] Nikolaus Hansen. *The CMA Evolution Strategy: A Comparing Review*, pages 75–102. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-32494-2. doi: 10.1007/3-540-32494-1_4.

[13] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[14] Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. CMA-ES/pycma on Github. Zenodo, DOI:10.5281/zenodo.2559634, February 2019.

[15] Hannes Homburger, Katrin Baumgärtner, Moritz Diehl, and Johannes Reuter. Gauss-newton accelerated mppi control, 2026.

[16] Taylor Howell, Nimrod Gileadi, Saran Tunyasuvunakool, Kevin Zakka, Tom Erez, and Yuval Tassa. Predictive sampling: Real-time behaviour synthesis with mujoco, 2022. URL https://arxiv.org/abs/2212.00541.

[17] Hilbert J. Kappen. Linear theory for control of nonlinear stochastic systems. *Physical review letters*, 95 20:200201, 2004. URL https://api.semanticscholar.org/CorpusID:5774276.

[18] 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. ISSN 1367-5788. doi: https://doi.org/10.1016/j.arcontrol.2023.100931. URL https://www.sciencedirect.com/science/article/pii/S1367578823000950.

[19] Jean B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[20] Quentin Le Lidec, Igor Kalevatykh, Ivan Laptev, Cordelia Schmid, and Justin Carpentier. Differentiable simulation for physical system identification. *IEEE Robotics Autom. Lett.*, 6(2):3413–3420, 2021. doi: 10.1109/LRA.2021.3062323. URL https://doi.org/10.1109/LRA.2021.3062323.

[21] Quentin Le Lidec, Louis Montaut, Yann de Mont-Marin, Fabian Schramm, and Justin Carpentier. End-to-end and highly-efficient differentiable simulation for robotics, 2025. URL https://arxiv.org/abs/2409.07107.

[22] Luca Martino, Víctor Elvira, and Francisco Louzada. Effective sample size for importance sampling based on discrepancy measures. *Signal Processing*, 131:386–401, 2017. ISSN 0165-1684. doi: https://doi.org/10.1016/j.sigpro.2016.08.025.

[23] Jorge Nocedal and Stephen J. Wright. Numerical optimization. In *Fundamental Statistical Inference*, 2018. URL https://api.semanticscholar.org/CorpusID:189864167.

[24] Pablo A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, 2003.

[25] Jan Peters and Stefan Schaal. Policy gradient methods for robotics. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2219–2225, 2006. URL https://api.semanticscholar.org/CorpusID:613022.

[26] Reuven Y. Rubinstein. Simulation and the monte carlo method. In *Wiley series in probability and mathematical statistics*, 1981. URL https://api.semanticscholar.org/CorpusID:39230485.

[27] Reuven Y. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology And Computing In Applied Probability*, 1:127–190, 1999. URL https://api.semanticscholar.org/CorpusID:18838459.

[28] Alessandro Rudi, Ulysse Marteau-Ferey, and Francis Bach. Finding global minima via kernel approximations: Finding global minima via kernel approximations. *Math. Program.*, 209(1):703–784, April 2024. ISSN 0025-5610. doi: 10.1007/s10107-024-02081-4. URL https://doi.org/10.1007/s10107-024-02081-4.

[29] Fabian Schramm, Pierre Fabre, Nicolas Perrin-Gilbert, and Justin Carpentier. Reference-free sampling-based model predictive control. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, Vienna, Austria, 2026. URL https://arxiv.org/abs/2511.19204.

[30] Evangelos A. Theodorou and Emanuel Todorov. Relative entropy and free energy dualities: Connections to path integral and kl control. *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 1466–1473, 2012.

[31] Emanuel Todorov. Linearly-solvable markov decision problems. In *Neural Information Processing Systems*, 2006.

[32] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

[33] Marc Toussaint. Robotic: Robotic control interface & manipulation planning library. https://github.com/MarcToussaint/robotic.

[34] Marc Toussaint, Cornelius V. Braun, and Joaquim Ortiz de Haro. NLP Sampling: Combining MCMC and NLP Methods for Diverse Constrained Sampling. *ArXiv*, abs/2407.03035, 2024.

[35] Giulio Turrisi, Valerio Modugno, Lorenzo Amatucci, Dimitrios Kanoulas, and Claudio Semini. On the Benefits of GPU Sample-Based Stochastic Predictive Controllers for Legged Locomotion. *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13757–13764, 2024. URL https://api.semanticscholar.org/CorpusID:268513407.

[36] Keenon Werling, Dalton Omens, Jeongseok Lee, Ioannis Exarchos, and C. Karen Liu. Fast and feature-complete differentiable physics engine for articulated rigid bodies with contact constraints. In Dylan A. Shell, Marc Toussaint, and M. Ani Hsieh, editors, *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021. doi: 10.15607/RSS.2021.XVII.034. URL https://doi.org/10.15607/RSS.2021.XVII.034.

[37] Grady Williams, Andrew Aldrich, and Evangelos A. Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance Control and Dynamics*, 40:344–357, 2017.

[38] 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*, 34:1603–1622, 2017. URL https://api.semanticscholar.org/CorpusID:8719932.

[39] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 2004.

[40] Haoru Xue, Chaoyi Pan, Zeji Yi, Guannan Qu, and Guanya Shi. Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4974–4981, 2025. doi: 10.1109/ICRA55743.2025.11127320.

[41] Yifan Zhai, Rudolf Reiter, and Davide Scaramuzza. Pamppi: Perception-aware model predictive path integral control for quadrotor navigation in unknown environments. *ArXiv*, abs/2509.14978, 2025. URL https://api.semanticscholar.org/CorpusID:281394267.