Inferring Smooth Robot Control: Monte Carlo Posterior Policy Iteration with Gaussian Processes

Anonymous Author(s)
Affiliation
Address
email

Abstract: Monte Carlo methods have become increasingly relevant for control of non-differentiable systems, approximate dynamics models, and learning from data. These methods scale to high-dimensional spaces and are effective at the non-convex optimization often seen in robot learning. We look at sample-based methods from the perspective of inference-based control, specifically posterior policy iteration. From this perspective, we highlight how Gaussian noise priors produce rough control actions that are unsuitable for physical robot deployment. Considering smoother Gaussian process priors, as used in episodic reinforcement learning and motion planning, we demonstrate how smoother model predictive control can be achieved using online sequential inference. This inference is realized through an efficient factorization of the action distribution, and novel means of optimizing the likelihood temperature for to improve importance sampling accuracy. We evaluate this approach on several high-dimensional robot control tasks, matching the sample efficiency of prior heuristic methods while also ensuring smoothness. Simulation results can be seen at monte-carlo-ppi.github.io.

Keywords: Approximate inference, policy search, model predictive control

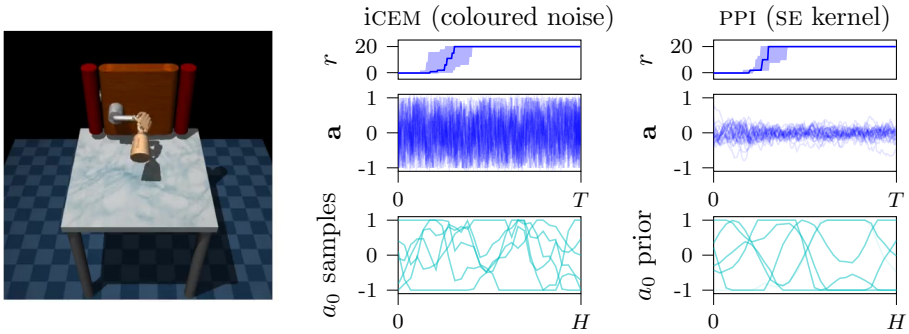

Figure 1: High-dimensional, contact-rich tasks such as manipulation (left) can be performed effectively using sample-based model predictive control. While prior work uses correlated actuator noise to improve sample-efficiency and exploration, these methods do not preserve the smoothness in the downstream actuation $\boldsymbol{a}$, resulting in aggressive control (center). We use smooth Gaussian process priors to infer posterior actions (right), which preserves smoothness while maintaining performance and sample efficiency, as both are using only 32 samples. Rewards r show quartiles over 25 seeds.

1 Introduction

Learning robot control requires optimization to be performed on sampled transitions of the environment [1]. Monte Carlo methods [2] provide a principled means to approach such algorithms, bridging black-box optimization and approximate inference techniques. These methods have been adopted extensively by the community for their impressive simulated [3, 4, 5, 6] and real-world [7, 8, 9, 10, 11, 12, 13, 14, 15] robot learning results. Their appeal is partly due to no required differentiability of the dynamics or objective, making them amenable to any simulation environment with minimal adjustments. Moreover, their stochastic nature also avoids issues with local minima that occur with gradient-based solvers [16, 17]. Finally, while Monte Carlo sampling is expensive, shooting

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

methods can be effectively parallelized across processes, and the advent of simulations on GPUs also provide a means of acceleration [18, 13]. However, some aspects of black-box optimization are open to criticism. Sample-based solvers such as the cross-entropy method (CEM) [19] appear 'wasteful', ignoring computation by throwing away most samples, while others maintain high-entropy search distributions to avoid premature convergence [18]. Moreover, many design decisions and hyperparameters are heuristic in nature, which is undesirable from both the user- and research perspective when interpreting, tuning or advancing these methods.

In this work, we consider Monte Carlo optimal control through the broader perspective of inference-based control [20, 21, 22, 23, 24, 25, 26, 27], where optimization is achieved through importance sampling [28]. This approach covers settings such as policy search [29], motion planning [8, 30] and model predictive control (MPC) [18]. From this view point, we highlight two key design decisions: the likelihood temperature and the distribution over action sequences. An adaptive temperature scheme is crucial for controlling the optimization behavior across objectives and distributions, but in many methods this aspect is ignored or opaque. Moreover, correlated action sequences are equally crucial for performing effective exploration and control in practical settings. *Smoothness*, arising from such correlations, is an aspect of human motion [31]. Smooth priors have taken many forms across domains, such as movement primitives [32], smoothed- [11, 12] or coloured noise [4]. We use Gaussian processes [33] as action priors and show how they can be scaled to high-dimensional action spaces through factorization of the covariance. Evaluating on simulated robotic systems, we reproduce prior results on policy search while transferring these ideas to MPC, matching prior performance w.r.t. sample efficiency while ensuring smooth actuation.

Contribution First, we present a perspective of episodic inference-based control based on Gibbs posteriors. Using this view, we then present novel algorithm variants that incorporate the approximate inference error due to importance sampling. Thirdly, we demonstrate how richer Gaussian process priors can be incorporated into model predictive control using online sequential inference, which achieves greater smoothness and sample efficiency than standard white noise priors.

2 Monte Carlo Methods for Optimal Control

We consider the standard (stochastic) optimal setting in discrete-time, with states $\boldsymbol{s} \in \mathbb{R}^{d_s}$, actions $\boldsymbol{a} \in \mathbb{R}^{d_a}$, and sequences $\boldsymbol{A} := \{\boldsymbol{a}_1, \dots, \boldsymbol{a}_T\}$. Optimization is framed as maximizing a reward $r : \mathbb{R}^{d_s} \times \mathbb{R}^{d_a} \to \mathbb{R}$ under stochastic dynamics $p(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{a}_t)$ and initial state distribution $p(\boldsymbol{s}_1)$,

$$\max_{\boldsymbol{a}_1, \boldsymbol{a}_2, \dots} \mathbb{E}\left[\sum_{t=1}^{T} r(\boldsymbol{s}_t, \boldsymbol{a}_t)\right] \quad \text{s.t.} \quad \boldsymbol{s}_{t+1} \sim p(\cdot \mid \boldsymbol{s}_t, \boldsymbol{a}_t), \quad \boldsymbol{s}_1 \sim p(\boldsymbol{s}_1). \tag{1}$$

This work focuses on the episodic setting, performs optimization after evaluating the current solution over a finite time horizon T. We frequently use the return R, where $R(\boldsymbol{S}, \boldsymbol{A}) = \sum_{t=1}^{T} r(\boldsymbol{s}_t, \boldsymbol{a}_t)$.

2.1 Variational Optimization with Gibbs Posteriors

The optimization outlined above is amenable to gradient-based solvers such as stochastic differential dynamic programming [34]. However, to aid optimization through exploration and regularization, we can consider optimizing a *belief* over action sequences $q \in \mathcal{Q}$. More generally, this is referred to as variational optimization, $\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) \leq \mathbb{E}_{\boldsymbol{x} \sim q(\cdot)}[f(\boldsymbol{x})]$, when optimizing $\boldsymbol{x}$ given an objective f [35]. This work concerns when $\boldsymbol{x}$ represents an open-look action sequence and f in an episodic return. Inferring an action sequence from data is known as input estimation, and Bayesian inference can be performed using message passing of the appropriate probabilistic graphical model, capturing the sequential structure of the problem and necessary priors [27]. If the measurement log-likelihood is replaced with the control objective, this inference computation can be shown to have precise dualities with dynamic programming-based optimal control [36]. While this switch in objective provides a powerful suite of inference tools for efficient computation, it requires treating the control objective as a Markovian log-likelihood, which is not the case for episodic objectives. The Gibbs likelihood is a more general treatment of the objective-as-likelihood (Definition 1) [37, 38].

Definition 1 *(Gibbs likelihoods and posteriors) For a loss f and prior $p(\boldsymbol{x})$, the Gibbs posterior q_α for parameter $\boldsymbol{x}$ is derived by constructing the Gibbs likelihood $\exp(-\alpha\, f(\boldsymbol{x}))$ from the loss,*

$$q_\alpha(\boldsymbol{x}) = \frac{1}{Z_\alpha} \exp(-\alpha\, f(\boldsymbol{x}))\, p(\boldsymbol{x}), \quad Z_\alpha = \int \exp(-\alpha\, f(\boldsymbol{x}))\, p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}, \quad \alpha \geq 0. \tag{2}$$

This posterior minimizes the following objective

$$q_\alpha = \arg\min_{q \in \mathcal{Q}} \mathbb{E}_{\boldsymbol{x} \sim q(\cdot)}[f(\boldsymbol{x})] + \tfrac{1}{\alpha}\mathbb{D}_{\mathrm{KL}}[q(\boldsymbol{x}) \mid\mid p(\boldsymbol{x})]. \tag{3}$$

This objective appears in PAC-Bayes methods [37], mirror descent methods [39] and Bayesian inference as the evidence lower bound objective when $f(\boldsymbol{x})$ is a negative log-likelihood [38].

Augmenting the variational optimization objective with prior regularization (Equation 3), we obtain an expression of the optimal belief in the action sequence (Equation 2). The parameter α has a range of meanings, depending on context. In PAC-Bayes it is the dataset size, in mirror descent it is an update step size and in risk-sensitive control it is the sensitivity [40, 41]. Example 1 in Appendix A examines a tractable example of this update, demonstrating its soundness for optimization while reinforcing the importance of α for optimization regularization.

2.2 Posterior Policy Iteration

The optimal control problem (Equation 1) is ambiguous regarding whether the action sequence or state-action trajectory is the optimization variable. Applying the Gibbs posterior to the optimal control setting recovers Rawlik et al.'s posterior policy iteration [40], which can be implemented using the joint distribution or policy. We consider the following joint state-action distribution, that factorizes in the following Markovian fashion $p(\boldsymbol{S}, \boldsymbol{A}) = p(\boldsymbol{s}_1) \prod_{t=1}^{T} p(\boldsymbol{s}_{t+1} \mid \boldsymbol{s}_t, \boldsymbol{a}_t)\, p(\boldsymbol{a}_t \mid \boldsymbol{s}_t)$. Posterior policy iteration updates the state-action distribution through the policy, constructing a Gibbs likelihood from the reward, as the dynamics and initial state distribution are constant.

Definition 2 *(Posterior policy iterations (PPI) [42]) As the initial distribution and dynamics are shared by the prior and posterior joint state-action distribution, the joint Gibbs posterior $q_\alpha(\boldsymbol{S}, \boldsymbol{A}) \propto \exp(\alpha R(\boldsymbol{S}, \boldsymbol{A}))\, p(\boldsymbol{S}, \boldsymbol{A})$ can be alternatively expressed using the policy posterior update $q_\alpha(\boldsymbol{A} \mid \boldsymbol{S}) \propto \exp(\alpha R(\boldsymbol{S}, \boldsymbol{A}))\, p(\boldsymbol{A} \mid \boldsymbol{S})$.*

Using this update, the key decisions are choosing $p(\boldsymbol{A}|\boldsymbol{S})$, α and the inference approximation. If p and q_α are Gaussian, then PPI involves iterative refinement of the distribution. In the Monte Carlo setting, q_α takes the from of an importance-weighted empirical distribution. To apply iteratively, p is updated using the M-projection, following the objective (Equation 3), i.e. a weighted maximum likelihood fit of the policy parameters. This approach is an stochastic approximate expectation maximization (SAEM) method [43]. One question with PPI methods is how best to set the inverse temperature α during optimization, as it has a strong influence on the posterior.

3 Posterior Policy Constraints

The Gibbs posterior in Definition 2 has been adopted widely in control, albeit from a range of different perspectives, such as Bayesian smoothing [23], solutions to the Feynman-Kac equation [44], maximum entropy [26], mirror descent [45] and entropy-regularized reinforcement learning [46]. The latter provides an effective means of deriving α for stochastic optimization (Definition 3), using the constrained optimization view of entropy-regularized optimal control.

Definition 3 *(Episodic relative entropy policy search (eREPS) [29]) Maximize the expected return, subject to a KL bound ϵ on the policy update,*

$$\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{s}_{t+1} \sim p(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t), \boldsymbol{a}_t \sim q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{s}_t), \boldsymbol{s}_1 \sim p(\cdot)}[R(\boldsymbol{s}_t, \boldsymbol{a}_t)] \quad \textit{s.t.} \quad \mathbb{D}_{\mathrm{KL}}[q_{\boldsymbol{\theta}}(\boldsymbol{A}|\boldsymbol{S}) \mid\mid p(\boldsymbol{A}|\boldsymbol{S})] \leq \epsilon.$$

The posterior policy takes the form $q_{\boldsymbol{\theta}}(\boldsymbol{A}|\boldsymbol{S}) \propto \exp(\alpha R)\, p(\boldsymbol{A}|\boldsymbol{S})$, where α is obtained by minimizing the empirical dual $\mathcal{G}(\cdot)$ using N samples,

$$\min_\alpha \mathcal{G}(\alpha) = \tfrac{\epsilon}{\alpha} + \tfrac{1}{\alpha} \log \int p(\boldsymbol{S}, \boldsymbol{A}) \exp(\alpha R(\boldsymbol{S}, \boldsymbol{A}))\, \mathrm{d}\boldsymbol{S}\, \mathrm{d}\boldsymbol{A} \approx \tfrac{\epsilon}{\alpha} + \tfrac{1}{\alpha} \log \tfrac{1}{N} \textstyle\sum_{n=1}^{N} \exp(\alpha R_n).$$

Despite the optimization formulation, we interpret the update as a pseudo-posterior, but with the temperature obtained through the KL constraint. To reinforce this connection, we reverse the objective and constraint and adopt an equality constraint,

$$\min_{\boldsymbol{\theta}} \mathbb{D}_{\mathrm{KL}}[q_{\boldsymbol{\theta}}(\boldsymbol{A}|\boldsymbol{S}) \mid\mid p(\boldsymbol{A}|\boldsymbol{S})] \quad \text{s.t.} \quad \mathbb{E}_{\boldsymbol{s}_{t+1} \sim p(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t),\, \boldsymbol{a}_t \sim q_{\boldsymbol{\theta}}(\cdot|\boldsymbol{s}_t),\, \boldsymbol{s}_1 \sim p(\cdot)}[\textstyle\sum_t r(\boldsymbol{s}_t, \boldsymbol{a}_t)] = R^*.$$

This is now a maximum entropy problem, in the form minimizing a relative entropy (Lemma 1) [47].

Lemma 1 *(Minimum relative entropy [47]) Find q that minimizes the relative entropy from p, given a function $\boldsymbol{f}$ that describes random variable $\boldsymbol{x}$, $\min_q \mathbb{D}_{\mathrm{KL}}[q(\boldsymbol{x}) \mid\mid p(\boldsymbol{x})]$ s.t. $\mathbb{E}_{\boldsymbol{x}\sim q(\cdot)}[\boldsymbol{f}(\boldsymbol{x})] = \hat{\boldsymbol{f}}$. The solution to this problem is $q(\boldsymbol{x}) \propto \exp(-\boldsymbol{\lambda}^\top \boldsymbol{f}(\boldsymbol{x}))p(\boldsymbol{x})$, where $\boldsymbol{\lambda}$ is required to satisfy the expectation constraint. When $p(\boldsymbol{x})$ is uniform, $q(\boldsymbol{x})$ is a maximum entropy distribution.*

Considering the expectation constraint, in practice this estimator uses self-normalized importance sampling (SNIS) on episodic returns $R^{(n)}$, sampling policy prior p, where R^* is an hyperparameter,

$$\mathbb{E}_{\boldsymbol{s}_{t+1}\sim p(\cdot|\boldsymbol{s}_t,\boldsymbol{a}_t),\, \boldsymbol{a}_t\sim q(\cdot|\boldsymbol{s}_t),\, \boldsymbol{s}_1\sim p(\cdot)}[\textstyle\sum_t r(\boldsymbol{s}_t,\boldsymbol{a}_t)] \approx \sum_n w^{(n)}_{q/p} R^{(n)} = \frac{\sum_n R^{(n)} \exp(\alpha\, R^{(n)})}{\sum_n \exp(\alpha\, R^{(n)})} = R^*.$$

Regardless of R^*, this estimator is limited by inference accuracy. We capture this by applying a concentration inequality to this estimate (Theorem 1) [48]. This lower bound provides an objective for optimizing α, which balances policy improvement with approximate inference accuracy.

Theorem 1 *(Importance sampling estimator concentration inequality (Theorem 2, [48])) Let p and q be two probability densities such that $p \ll q$ and $d_2(p \mid\mid q) < +\infty$. Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ i.i.d. random variables sampled from q and $f : \mathcal{X} \to \mathbb{R}$ be a bounded function ($||f||_\infty < +\infty$). Then, for any $0 < \delta \leq 1$ and $N > 0$ with probability at least $1-\delta$:*

$$\mathbb{E}_{\boldsymbol{x}\sim p(\cdot)}[f(\boldsymbol{x})] \geq \tfrac{1}{N}\textstyle\sum_{i=1}^N w_{p/q}(\boldsymbol{x}_i) f(\boldsymbol{x}_i) - ||f||_\infty \sqrt{\frac{(1-\delta) d_2(p(\boldsymbol{x}) \mid\mid q(\boldsymbol{x}))}{\delta\, N}}. \tag{4}$$

The regularization term d_2 is the exponeniated Rény-2 divergence. While this is tractable for the multivariate Gaussian, it is otherwise not available in closed form. Fortunately, we can use the effective sample size (ESS) [49] as an approximation, as $\hat{N}_\alpha \approx N/d_2(p \mid\mid q_\alpha)$ (Lemma 2) [48, 50].

Lemma 2 *(Exponentiated Rény-2 divergence estimator [50]) The effective sample size diagnostic ($\hat{N}$) is an estimate of the exponentiated Rényi-2 divergence for finite samples, as $d_2(p||q)/N \approx \hat{N}^{-1}$,*

$$\lim_{N\to\infty} \frac{1}{N} \frac{(\sum_n w_{q/p}(\boldsymbol{x}_n))^2}{\sum_n w_{q/p}(\boldsymbol{x}_n)^2} = \int \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \mathrm{d}\boldsymbol{x} = \exp \mathbb{D}_2(p \mid\mid q)^{-1} = d_2(p \mid\mid q)^{-1}.$$

Combining Theorem 1 with our constraint, we maximize the unknown R^* using the IS lower bound to form an objective for α, incorporating the inference accuracy due to the sampling,

$$\max_\alpha R^*(\alpha,\delta) = \mathbb{E}_{q_\alpha/p}[R] - \mathcal{E}_R(\delta, \hat{N}_\alpha), \quad \mathcal{E}_R(\delta, \hat{N}_\alpha) = ||R||_\infty \sqrt{\frac{(1-\delta)}{\delta}} \frac{1}{\sqrt{\hat{N}_\alpha}}. \tag{5}$$

We refer to this approach as *lower-bound policy search* (LBPS). This objective combines the expected performance of q_α based on the IS estimate $\mathbb{E}_{q_\alpha/p}$, and a penalty term $\mathcal{E}_R$ based on the return and approximate inference accuracy. Treating p, N, $||R||_\infty$ as task specific hyperparameters, the only algorithm hyperparameter is the probability of the bound $\delta \in [0, 1)$. In practice, self-normalized importance sampling is used for control, as the normalizing constants of the Gibbs likelihoods are not available. While Metelli et al. also derive an SNIS lower bound [48], we found, as they did, that the IS lower bound with SNIS estimates work better in practice due to the conservatism of the SNIS bound. An interpretation of this approach is that the Rényi-2 regularization constrains the Gibbs posterior to one that can be estimated from the finite samples, as the divergence is used in evaluating IS sample complexity [51, 52]. Moreover, the role of the ESS for regularization is similar to the 'elite' samples in CEM. Connecting these two mechanisms as robust maximum estimators (Section A), we also propose *effective sample size policy search* (ESSPS), which optimizes α to achieve a desired ESS N^*, i.e. a Rényi-2 divergence bound, using the objective $\min_\alpha |\hat{N}_\alpha - N^*|$. Further details regarding PPI (Section A) and temperature selection methods (Table 1) are in the Appendix.

4 Online Posterior Policy Iteration & Prior Design

Model predictive control [53] is a popular approximation of the optimal control problem, and requires optimizing a shorter horizon H in an online fashion. We consider the online realization of posterior policy iteration, computing the finite-horizon future marginal posteriors given a likelihood on a subset of marginals from the past. $\mathcal{O}$ is used to represent the Gibbs likelihood variable [26]

$$q_\alpha(\boldsymbol{a}_{t:t+H} \mid \mathcal{O}_{1:\tau}) = \textstyle\int q_\alpha(\boldsymbol{a}_{1:t+H} \mid \mathcal{O}_{1:\tau})\, \mathrm{d}\boldsymbol{a}_{1:t} \propto \int p(\mathcal{O}_{1:\tau} \mid \boldsymbol{a}_{1:\tau})\, p(\boldsymbol{a}_{1:t+H})\, \mathrm{d}\boldsymbol{a}_{1:t},$$

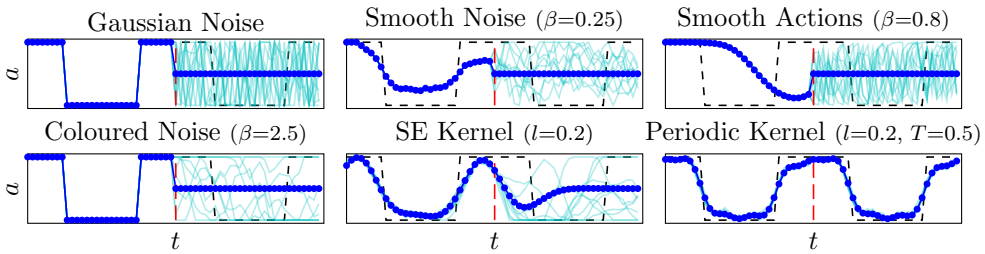

Figure 2: An important consideration for Monte Carlo control methods for robotics is optimizing smooth action sequence. This example shows a non-smooth optimal sequence - - -, which is undesirable to fit exactly. Prior methods struggle at providing both effective smooth solutions in the mean —— and action samples ——, as they ultimately fit the action distribution in an independent fashion. Using kernel-derived covariance function provides both. - - - denotes the optimization horizon, beyond which is exploratory actions derived from both the posterior and prior.

where $\tau \leq t + h$. As an analogy, this is equivalent to combining forecasting with state estimation, i.e. $p(\boldsymbol{x}_{1:t+H}|\boldsymbol{y}_{1:t})$ for states $\boldsymbol{x}$ and measurements $\boldsymbol{y}$. For arbitrary priors on the action space, this computation is tractable if working with Gaussian processes. In fact, a recurring aspect across several many posterior policy iteration-like approaches is the use of Gaussian process policies,

$$p(\boldsymbol{A} \mid \boldsymbol{S}) = \begin{cases} \prod_t \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), & \text{(Independent Gaussian noise, e.g. [18])}, \\ \mathcal{GP}(\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(t)), & \text{(Markovian Gaussian process, e.g. GPMP [30])}, \\ \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{w}}^\top \phi(t), \phi(t)^\top \boldsymbol{\Sigma}_{\boldsymbol{w}} \phi(t)), & \text{(Bayesian linear regression, e.g. ProMP [32])}, \\ \prod_t \mathcal{N}(\boldsymbol{k}_t + \boldsymbol{K}_t \boldsymbol{s}, \boldsymbol{\Sigma}_t), & \text{(time-varying linear Gaussian e.g. [40, 54, 41])}, \\ \mathcal{GP}(\boldsymbol{\mu}(\boldsymbol{s}), \boldsymbol{\Sigma}(\boldsymbol{s})), & \text{(non-parametric Gaussian process [55])}. \end{cases}$$

While Gaussian noise is simple, for robotics tasks, more sophisticated noise is required for safety and effective exploration. Prior work on Monte Carlo MPC has introduced similar priors [11, 12]. For action sampling, where $\boldsymbol{v}_t^{(n)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, $\beta \in [0, 1]$ and $\boldsymbol{\Sigma}_t = \boldsymbol{L}_t \boldsymbol{L}_t^\top$, actions are computed using

$$\boldsymbol{a}_t^{(n)} = \boldsymbol{\mu}_t + \boldsymbol{L}_t \boldsymbol{n}_t^{(n)}, \quad \boldsymbol{n}_t^{(n)} = \beta \boldsymbol{v}_t^{(n)} + (1-\beta)\boldsymbol{n}_{t-1}^{(n)}, \quad \text{or} \quad \boldsymbol{n}_t^{(n)} = \beta \boldsymbol{v}_t^{(n)} + \sqrt{(1-\beta^2)}\, \boldsymbol{n}_{t-1}^{(n)}.$$

However, in practice it is also implemented as $\boldsymbol{a}_t^{(n)} = \beta(\boldsymbol{\mu}_t + \boldsymbol{L}_t \boldsymbol{v}_t^{(n)}) + (1-\beta)\boldsymbol{a}_{t-1}^{(n)}$ [1]. While this approach directly smooths the actuation, it also introduces a lag, which may deteriorate performance. Other approaches have used colored noise for sampling the noise $\boldsymbol{n}$ [4]. Contrast these approaches to Gibbs sampling a multivariate Gaussian joint distribution with 1-step cross-correlations [57], which is $\boldsymbol{a}_{t|t-1}^{(n)} = \boldsymbol{\mu}_{t|t-1}^{(n)} + \boldsymbol{L}_{t|t-1} \boldsymbol{v}_t^{(n)}$, where $\boldsymbol{\mu}_{t|t-1}^{(n)} = \boldsymbol{\mu}_t + \boldsymbol{\Sigma}_{t,t-1}\boldsymbol{\Sigma}_{t-1}^{-1}(\boldsymbol{a}_{t-1}^{(n)} - \boldsymbol{\mu}_{t-1})$, and $\boldsymbol{\Sigma}_{t|t-1} = \boldsymbol{\Sigma}_t - \boldsymbol{\Sigma}_{t,t-1}\boldsymbol{\Sigma}_{t-1}^{-1}\boldsymbol{\Sigma}_{t,t-1}^\top$. The differences are subtle, but important. The initial proposed sampling scheme essentially adds correlated noise to the mean for exploration, but does not consider the smoothness of the mean itself. The practical implementation incorporates the previous action, but through exponential smoothing, which introduces a fixed lag that potentially degrades the quality of the mean action sequence. Correct sampling of the joint distribution has neither of these issues and naturally extends to correlations over several time steps. We do this in a general fashion by considering a continuous-time Gaussian process $\boldsymbol{a_t} \sim \mathcal{GP}(\boldsymbol{\mu}(\boldsymbol{t}), \boldsymbol{\Sigma}(\boldsymbol{t}))$ (see Section G, Appendix) and a discrete time sequence $[t_i, \ldots, t_j]$ as an input vector $\boldsymbol{t}$, i.e. $\boldsymbol{a_t} = \boldsymbol{a}_{t_i:t_j}$. Given a posterior action sequence $\boldsymbol{a}_{t_1:t_2|\mathcal{O}}$ for t_1 to t_2, we wish sample action sequences from t_3 to t_4 that incorporate this belief and the prior. A tractable means to achieve this is to obtain the Gaussian likelihood $\mathcal{N}(\boldsymbol{\mu}_{\mathcal{O}|t_1:t_2}, \boldsymbol{\Sigma}_{\mathcal{O}|t_1:t_2})$ for the given posterior. Using $\boldsymbol{K}_{t_1:t_2} = \boldsymbol{\Sigma}_{t_1:t_2,t_1:t_2}\boldsymbol{\Sigma}_{\mathcal{O}|t_1:t_2}^{-1}$,

$$\boldsymbol{\mu}_{t_1:t_2|\mathcal{O}} = \boldsymbol{\mu}_{t_1:t_2} + \boldsymbol{K}_{t_1:t_2}(\boldsymbol{\mu}_{\mathcal{O}|t_1:t_2} - \boldsymbol{\mu}_{t_1:t_2}), \quad \boldsymbol{\Sigma}_{t_1:t_2,t_1:t_2|\mathcal{O}} = \boldsymbol{\Sigma}_{t_1:t_2,t_1:t_2} - \boldsymbol{K}_{t_1:t_2}\boldsymbol{\Sigma}_{\mathcal{O}|t_1:t_2}\boldsymbol{K}_{t_1:t_2}^\top.$$

We introduce $\boldsymbol{\nu}$ and $\boldsymbol{\Lambda}$ to capture the unknown terms involving $\boldsymbol{a}_{\mathcal{O}|t_1:t_2}$ w.r.t. $\boldsymbol{a}_{t_1:t_2}$ and $\boldsymbol{a}_{t_1:t_2|\mathcal{O}}$,

$$\begin{aligned} \boldsymbol{\nu}_{t_1:t_2} &= \boldsymbol{\Sigma}_{\mathcal{O}|t_1:t_2}^{-1}(\boldsymbol{\mu}_{\mathcal{O}|t_1:t_2} - \boldsymbol{\mu}_{t_1:t_2}) = \boldsymbol{\Sigma}_{t_1:t_2,t_1:t_2}^{-1}(\boldsymbol{\mu}_{t_1:t_2|\mathcal{O}} - \boldsymbol{\mu}_{t_1:t_2}), \\ \boldsymbol{\Lambda}_{t_1:t_2} &= \boldsymbol{\Sigma}_{\mathcal{O}|t_1:t_2,t_1:t_2}^{-1} = \boldsymbol{\Sigma}_{t_1:t_2,t_1:t_2}^{-1}(\boldsymbol{\Sigma}_{t_1:t_2,t_1:t_2} - \boldsymbol{\Sigma}_{t_1:t_2,t_1:t_2|\mathcal{O}})\boldsymbol{\Sigma}_{t_1:t_2,t_1:t_2}^{-1}. \end{aligned}$$

When extrapolating to the new time sequence, the distribution is

$$\begin{aligned} \boldsymbol{\mu}_{t_3:t_4|\mathcal{O}} &= \boldsymbol{\mu}_{t_3:t_4} + \boldsymbol{\Sigma}_{t_3:t_4,t_1:t_2}\boldsymbol{\nu}_{t_1:t_2}, \\ \boldsymbol{\Sigma}_{t_3:t_4,t_3:t_4|\mathcal{O}} &= \boldsymbol{\Sigma}_{t_3:t_4,t_3:t_4} - \boldsymbol{\Sigma}_{t_3:t_4,t_1:t_2}\boldsymbol{\Lambda}_{t_1:t_2}\boldsymbol{\Sigma}_{t_3:t_4,t_1:t_2}^\top. \end{aligned} \tag{6}$$

[1] See the source code for Nagabandi et al. [11] and `MBRL-lib` [56].

For a stationary kernel, fixed planning horizon and fixed control frequency, the prior $\boldsymbol{\Sigma}^{-1}_{t:t+H,t:t+H}$ is constant and computed at initialization to avoid repeated inversion. Figure 2 demonstrates how this update lets us combine our prior with previous posterior in a principled fashion. Moreover, its continuous-time construction means that the time resolution can be updated, not just the time window, for planning at different timescales [30].

Compared to the independence assumption, modeling correlations between actions introduces complexity. The full covariance over (flattened) time and action has a complexity $\mathcal{O}(T^3 d_a^3)$, which is infeasible to work with. Assuming independence between actions, a GP per action has a complexity of $\mathcal{O}(T^3 d_a)$, requiring d_a GPs to be fit, which is not desirable for online methods, i.e. MPC. To avoid the linear scaling w.r.t. d_a, we propose using the *matrix Normal distribution* (Definition 4) for scalability, as it is parameterized into single T and d_a-dimensional covariances,

Definition 4 *(Matrix Normal Distribution (MaVN) [58]) For a random matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, it follows a matrix normal distribution $\boldsymbol{X} \sim \mathcal{MN}(\boldsymbol{M}, \boldsymbol{K}, \boldsymbol{\Sigma})$, where $\boldsymbol{M} \in \mathbb{R}^{n \times p}, \boldsymbol{K} \in S_+^n$ and $\boldsymbol{\Sigma} \in S_+^p$, if and only if $vec(\boldsymbol{X}) \sim \mathcal{N}(vec(\boldsymbol{M}), \boldsymbol{\Sigma} \otimes \boldsymbol{K})$, where $\otimes$ denotes the Kronecker product.*

Using the Kronecker-structured covariance provides a useful decomposition of the time-based covariance $\boldsymbol{K}$, that defines correlations between time steps, and an action covariance $\boldsymbol{\Sigma}$ that captures correlations between actions. Typically we assume actions are independent, but cross-correlations could be learned from experience for richer coordination. While this Kronecker structure does not fully capture the correlations between time and actions, the structure is very useful for MPC on robotic systems, where the actions space could be very high but the planning horizon is sufficiently small for covariance estimation using a reasonable number of Monte Carlo rollouts.

Feature Approximations Despite the matrix Normal factorization, computing the correlations between actions still requires a dense $H \times H$ covariance matrix $\boldsymbol{K}$ for planning horizon H. To sparsify this quantity, we consider kernel approximations, such as the canonical basis functions $\sum_n k(\cdot, \boldsymbol{x}_n)$ and *spectral* approximations using random features $\sum_n \phi_n(\cdot)$ [59], for a Bayesian linear model $\boldsymbol{\phi_t}^\top \boldsymbol{W}$. Focusing on the squared exponential (SE) kernel, this results in radial basis function (RBF) and random Fourier features (RFF) respectively. Interestingly, RBF features are closely related to probabilistic movement primitives, used extensively in policy search for robotics [32]. For one-dimensional inputs, RFFs are effectively approximated applying Gauss-Hermite quadrature [60] to the random weights [61]. RBF features and RFFs approximate w.r.t. time and frequency respectively, and could be combined [62]. Using these continuous-time features, the optimization is now abstracted from planning horizon and control frequency, providing much greater flexibility. Secondly, due to the features, a factorized weight covariance approximation does not sacrifice smoothness. Moreover, the moment updates described above are not needed, as only $\boldsymbol{\phi_t}$ is updated.

5 Related Work

Inference-based control. Posterior policy iteration was proposed by Rawlik et al. [40] and covers prior methods developed from Bayesian smoothing [23, 36], expectation maximization [22, 63], entropy regularization [46, 9] and path integral [64] perspectives. For MPC specifically, the path integral-based MPPI was popularized [18], with alternative formulations based on mirror descent [45] and variational inference [5, 65, 66]. PIPC [25] models the optimal state-action distribution as a sparse Gaussian process and uses linearization for approximate inference. The same approach is used for Gaussian process motion planning [30], which are also optimized using sampling [8]. Gaussian quadrature is also used for inference-based MPC [27]. Concurrent work uses the ESS for a temperature adjusting heuristic for MPPI [15], and also combines policy search with MPC using PPI techniques [67]. See Section B for a more in-depth discussion on these related works.

Gaussian processes for control. This work adopts GPs for correlated action priors. This is distinct from prior work which uses GPs to approximate dynamics or value functions, e.g. [68, 69, 70, 71].

Policy design and regularization. Smooth actuation is important in robot learning for safety and exploration. It has been proposed for Monte Carlo MPC [11, 12, 4], and more broadly incorporated using augmented objectives, parameter sampling and policy design, e.g. [72, 73, 74].

Stochastic search. Probabilistic interpretations of black-box optimization algorithms are well established [75, 76, 77], however prior work did not connect the ESS and elite samples. CEM and extensions have also been adopted widely as a solver for MPC [3, 4, 6].

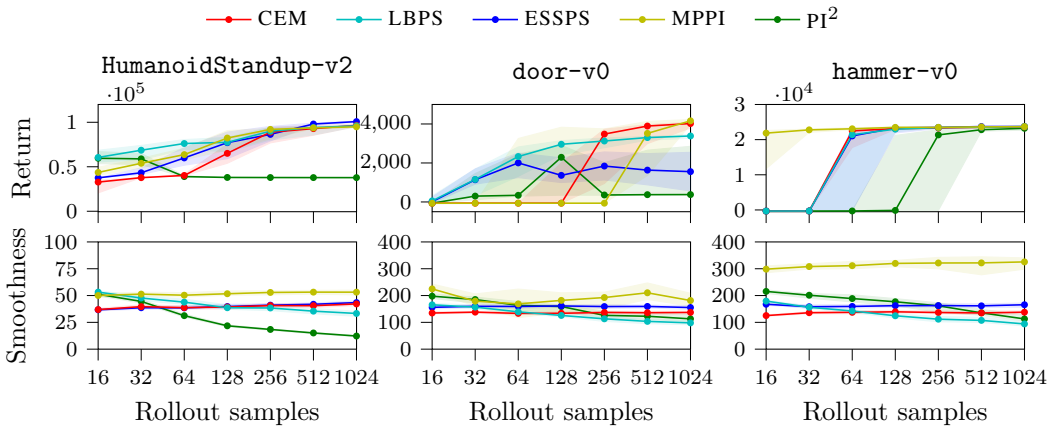

Figure 3: Monte Carlo MPC with white noise policies. Displaying quartiles over 50 seeds. White noise priors require a large number of samples for good performance, across methods and tasks.

6 Experimental Results

For the empirical results, we assess the effectiveness of both the Gibbs posterior methods and policy design across various settings. Black-box optimization compares the posteriors alone, while policy search compares LBPS and ESSPS against prior eREPS results. For MPC, we evaluate online PPI using LBPS and ESSPS with white noise and SE kernel priors, comparing against previous approaches on high-dimensional, contact-rich tasks, with a focus on sample efficiency and control smoothness.

6.1 Black-box Optimization

To understand the behaviour of the proposed posteriors as solvers, the performance results of LBPS and ESSPS on a range of standard black-box optimization functions over a range of hyperparameters, with eREPS and CEM as baselines (Appendix D.1). The results show that ESS is an useful characterization for these methods, as each solver exhibits consistent ESS for a given hyperparameter value. However, the uniform weights used by CEM maintain entropy longer than ESSPS, which can lead to better optima, so the ESS is not the only diagnostic required to capture the behavior of these solvers.

6.2 Policy Search

For policy search, as LBPS and ESSPS are closely related to eREPS, we repeat the experiment from prior work performing the 'ball in a cup' task using a Barret WAM [14], which has been shown to transfer to the physical system [78, 14, 79], as a benchmark task. Moreover, we replace ProMPs with Matrix normal RBF and RFF policies. From the kernel perspective, this feature approximation is motivated by the large ($T \simeq 1000$) task horizon. The results in Appendix D.2 confirm that these solvers are all capable of solving the task, while RBF and RFF features perform equally well.

6.3 Model Predictive Control

We evaluate online PPI across a range of high-dimensional robotic control tasks in `MuJoCo` [80], including `HumanoidStandup-v2` in `Gym` [81] and `door-v0`, `hammer-v0` from `mj_envs`, using the Adroit hand [82]. To measure smoothness, we adopt the FFT-based score $\frac{2}{Nf_s}\sum_{i=1}^{N} a_i f_i$ [72], with sampling frequency f_s and N resolvable frequencies f with amplitudes a. We compute the Euclidean norm of the action sequence over time and apply the smoothness measure to this 1D signal. For the evaluation, we focus on a low computational budget, with 1 or 2 iterations per timestep. To assess the impact of approximate inference, we assess performance over an logarithmic range of sample rollouts, following prior work [4]. Details may be found in Appendix E.2.

The white noise baseline. Figure 3 shows online PPI with white noise priors using LBPS and ESSPS, with MPPI, CEM and PI^2 baselines. While each solver performs comparably for 1024 rollouts, the low sample regime shows greater performance variance. While MPPI seems particularly effective, Figure 11 shows that its average ESS is particularly low, $\simeq 1$ for many cases. Combined with the fixed variances, this suggests optimization is closer to greedy random search than importance sampling. The poor `door-v0` performance of ESSPS is due to slow opening, rather than task failure.

Policy design for smooth control. Figure 4 shows online PPI with smooth priors. LBPS, ESSPS and MPPI use the SE kernel, with iCEM [4] and MPPI with smooth actions and noise as a baseline. The

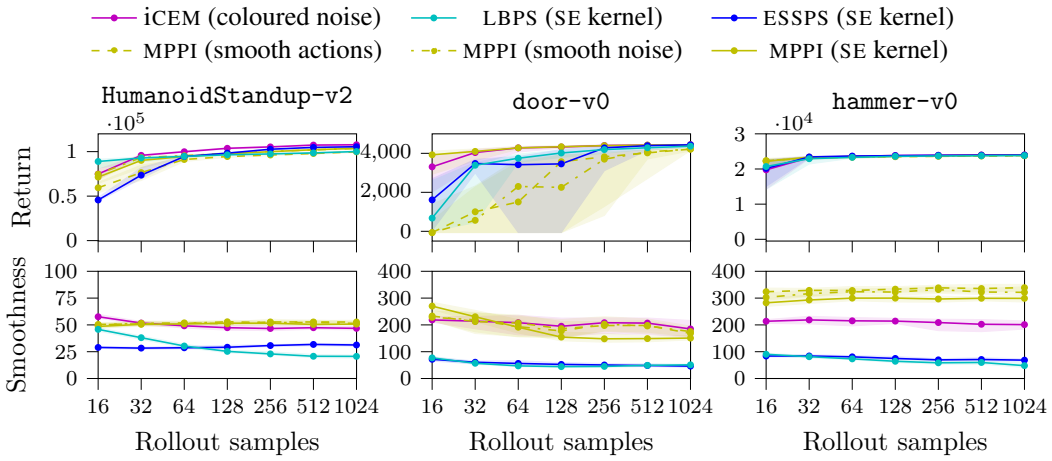

Figure 4: Monte Carlo MPC with smoothed noise policies. Displaying quartiles over 50 seeds. Compared to white noise priors, smooth action priors improve sample efficiency dramatically, but only PPI methods (LBPS, ESSPS) preserve this smoothness in the downstream control.

smooth action distributions greatly improves performance across models, tasks and sample sizes, due to effective exploration. As desired, the richer SE kernel provides much greater smoothness, by a factor of 2 compared to baselines, with limited impact to performance. It is unsurprising that the smoothness bias reduces performance if optimal behavior is non-smooth, as illustrated in Figure 2. Figure 1 and Appendix D.3.2 depicts the actuation these smoothness measures corresponds to.

Comparing kernel- and feature-based policies Figure 13 replaces the SE kernel with RBF and RFF features, while keeping the lengthscale. These policies perform worse given fewer samples, but are comparable with sufficient samples. We attribute this to the compounding errors of kernel approximation and fewer effective samples. In contrast to the policy search task, RFFs are superior.

Learning priors from data A benefit of using GP priors is the ability to optimize hyperparameters from expert demonstrations through the likelihood or a divergence. Moreover, the matrix normal distribution is useful for analyzing high-dimensional action sequences, as it decomposes temporal- and action correlations into viewable covariance matrices. Section D.3.3 shows the matrix normal distributions of expert demonstrations of the tasks, obtained through human and RL experts. Results show that, surprisingly, the demonstrations are rougher than the smoothness achievable with MPC. We attribute this to control artifacts from demonstration collection, and the use of Gaussian noise by RL agents. Applying the same methodology to the demonstrations of the smooth MPC agents proposed here extracts the expected action correlations across tasks. This analysis also raises the question of whether smoothness is an inductive bias we enforce for practicality, or a phenomena we expect to arise from optimality. If the latter, it may be that the simulated environments or objectives considered are lacking components that encourage smoothness, such as energy efficiency.

7 Conclusion

We present a broad perspective on episodic posterior policy iteration method for robotics and new methods for the Monte Carlo setting, based on regularizing the IS approximations. By considering vector-valued Gaussian processes for action priors, we have demonstrated how sample-efficient MPC can be performed as online inference and with greater control over actuator smoothness, connecting Monte Carlo MPC to prior work on policy search. This approach was validated on a set of high-dimensional MPC tasks closely matching baseline performance while preserving smoothness.

Limitations Much of the prior work is motivated by simplicity, minimizing hyperparameter tuning and numerical procedures such as matrix inversion [64]. In contrast, the contributions of this work introduces complexity, i.e. online temperature optimization and the use of dense covariance matrices in order to perform more sophisticated approximate inference. While this additional complexity has an impact on execution time (Table 2, Appendix), we hope the sample-efficiency when combined with accelerations such as GPU-integration should produce real-time algorithms [13].

Future work will consider model- and step-based control, which requires the additional consideration of dynamics and value function approximation respectively [5, 55, 83] when optimizing α.

References

[1] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

[2] A. B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[3] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, 2018.

[4] C. Pinneri, S. Sawant, S. Blaes, J. Achterhold, J. Stueckler, M. Rolinek, and G. Martius. Sample-efficient cross-entropy method for real-time planning. In *Conference on Robot Learning*, 2020.

[5] M. Okada and T. Taniguchi. Variational inference MPC for Bayesian model-based reinforcement learning. In *Conference on Robot Learning*, 2019.

[6] M. Lutter, L. Hasenclever, A. Byravan, G. Dulac-Arnold, P. Trochim, N. Heess, J. Merel, and Y. Tassa. Learning dynamics models for model predictive agents. *arXiv preprint arXiv:2109.14311*, 2021.

[7] F. Stulp, E. Theodorou, J. Buchli, and S. Schaal. Learning to grasp under uncertainty. In *2011 IEEE International Conference on Robotics and Automation*, 2011.

[8] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal. STOMP: Stochastic trajectory optimization for motion planning. In *2011 IEEE International Conference on Robotics and Automation*, 2011.

[9] C. Daniel, G. Neumann, O. Kroemer, J. Peters, et al. Hierarchical relative entropy policy search. *Journal of Machine Learning Research*, 2016.

[10] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou. Information-theoretic model predictive control: Theory and applications to autonomous driving. *IEEE Transactions on Robotics*, 2018.

[11] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, 2019.

[12] Y. Yang, K. Caluwaerts, A. Iscen, T. Zhang, J. Tan, and V. Sindhwani. Data-efficient reinforcement learning for legged robots. In *Conference on Robot Learning*, 2019.

[13] M. Bhardwaj, B. Sundaralingam, A. Mousavian, N. D. Ratliff, D. Fox, F. Ramos, and B. Boots. STORM: An integrated framework for fast joint-space model predictive control for reactive manipulation. In *Conference on Robot Learning*, 2022.

[14] M. Lutter, J. Silberbauer, J. Watson, and J. Peters. Differentiable physics models for real-world offline model-based reinforcement learning. In *IEEE International Conference on Robotics and Automation*, 2021.

[15] J. Carius, R. Ranftl, F. Farshidian, and M. Hutter. Constrained stochastic optimal control with learned importance sampling: A path integral approach. *The International Journal of Robotics Research*, 2022.

[16] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 2014.

[17] A. Abdolmaleki, R. Lioutikov, J. R. Peters, N. Lau, L. Pualo Reis, and G. Neumann. Model-based relative entropy stochastic search. In *Advances in Neural Information Processing Systems*, 2015.

[18] G. Williams, A. Aldrich, and E. A. Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 2017.

[19] R. Y. Rubinstein and D. P. Kroese. *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte Carlo Simulation*. 2004.

[20] P. Dayan and G. E. Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 1997.

[21] M. Toussaint and A. Storkey. Probabilistic inference for solving discrete and continuous state Markov decision processes. In *International Conference of Machine Learning*, 2006.

[22] J. Peters and S. Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *International Conference on Machine Learning*, 2007.

[23] M. Toussaint. Robot trajectory optimization using approximate inference. In *International Conference on Machine Learning*, 2009.

[24] H. J. Kappen, V. Gómez, and M. Opper. Optimal control as a graphical model inference problem. In *International Conference on Automated Planning and Scheduling*, 2013.

[25] M. Mukadam, C. Cheng, X. Yan, and B. Boots. Approximately optimal continuous-time motion planning and control via probabilistic inference. In *IEEE International Conference on Robotics and Automation*, 2017.

[26] S. Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

[27] J. Watson, H. Abdulsamad, R. Findeisen, and J. Peters. Efficient stochastic optimal control through approximate Bayesian input inference. *arXiv preprint arXiv:2105.07693*, 2021.

[28] H. J. Kappen and H. C. Ruiz. Adaptive importance sampling for control and inference. *Journal of Statistical Physics*, 2016.

[29] M. P. Deisenroth, G. Neumann, J. Peters, et al. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2013.

[30] M. Mukadam, J. Dong, X. Yan, F. Dellaert, and B. Boots. Continuous-time Gaussian process motion planning via probabilistic inference. *The International Journal of Robotics Research*, 2018.

[31] T. Flash and N. Hogan. The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience*, 1985.

[32] A. Paraschos, C. Daniel, J. Peters, G. Neumann, et al. Probabilistic movement primitives. In *Advances in Neural Information Processing Systems*, 2013.

[33] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[34] E. Theodorou, Y. Tassa, and E. Todorov. Stochastic differential dynamic programming. In *American Control Conference*, 2010.

[35] J. Staines and D. Barber. Variational optimization. *arXiv preprint arXiv:1212.4507*, 2012.

[36] J. Watson, H. Abdulsamad, and J. Peters. Stochastic optimal control as approximate input inference. In *Conference on Robot Learning*, 2019.

[37] B. Guedj. A primer on PAC-Bayesian learning. In *Proceedings of the second congress of the French Mathematical Society*, 2019.

[38] P. Alquier, J. Ridgway, and N. Chopin. On the properties of variational approximations of Gibbs posteriors. *Journal of Machine Learning Research*, 2016.

[39] B. Dai, N. He, H. Dai, and L. Song. Provable bayesian inference via particle mirror descent. In *International Conference on Artificial Intelligence and Statistics*, 2016.

[40] K. Rawlik, M. Toussaint, and S. Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. In *Robotics: Science and Systems*, 2012.

[41] J. Watson and J. Peters. Advancing trajectory optimization with approximate inference: Exploration, covariance control and adaptive risk. In *American Control Conference*, 2021.

[42] K. C. Rawlik. *On probabilistic inference approaches to stochastic optimal control*. PhD thesis, The University of Edinburgh, 2013.

[43] C. Wirth, J. Fürnkranz, and G. Neumann. Model-free preference-based reinforcement learning. In *Conference on Artificial Intelligence*, 2016.

[44] S. Satoh, H. J. Kappen, and M. Saeki. An iterative method for nonlinear stochastic optimal control based on path integrals. *IEEE Transactions on Automatic Control*, 2017.

[45] N. Wagener, C. an Cheng, J. Sacks, and B. Boots. An online learning approach to model predictive control. In *Robotics: Science and Systems*, 2019.

[46] J. Peters, K. Mülling, and Y. Altün. Relative entropy policy search. In *Conference on Artificial Intelligence*, 2010.

[47] J. van Campenhout and T. Cover. Maximum entropy and conditional probability. *IEEE Transactions on Information Theory*, 1981.

[48] A. M. Metelli, M. Papini, N. Montali, and M. Restelli. Importance sampling techniques for policy optimization. *Journal of Machine Learning Research*, 2020.

[49] A. Kong. A note on importance sampling using standardized weights. *University of Chicago, Dept. of Statistics, Technical Report*, 1992.

[50] C. Cortes, Y. Mansour, and M. Mohri. Learning bounds for importance weighting. In *Advances in Neural Information Processing systems*, 2010.

[51] S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 2017.

[52] J. Hernández-González and J. Cerquides. A robust solution to variational importance sampling of minimum variance. *Entropy*, 2020.

[53] A. Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 2016.

[54] V. Gómez, H. J. Kappen, J. Peters, and G. Neumann. Policy search for path integral control. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2014.

[55] H. van Hoof, G. Neumann, and J. Peters. Non-parametric policy search with limited information loss. *Journal of Machine Learning Research*, 2017.

[56] L. Pineda, B. Amos, A. Zhang, N. O. Lambert, and R. Calandra. MBRL-lib: A modular library for model-based reinforcement learning. *Arxiv*, 2021.

[57] A. Doucet. A note on efficient conditional simulation of Gaussian distributions. *Departments of Computer Science and Statistics, University of British Columbia*, 2010.

[58] A. P. Dawid. Some matrix-variate distribution theory: notational considerations and a Bayesian application. *Biometrika*, 1981.

[59] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, 2007.

[60] F. B. Hildebrand. *Introduction to numerical analysis*. Courier Corporation, 1987.

[61] M. Mutny and A. Krause. Efficient high-dimensional Bayesian optimization with additivity and quadrature Fourier features. In *Advances in Neural Information Processing Systems*, 2018.

[62] J. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *International Conference on Machine Learning*, 2020.

[63] J. Kober and J. Peters. Policy search for motor primitives in robotics. In *Advances in Neural Information Processing Systems*, 2009.

[64] E. Theodorou, J. Buchli, and S. Schaal. A generalized path integral control approach to reinforcement learning. *The Journal of Machine Learning Research*, 2010.

[65] A. Lambert, A. Fishman, D. Fox, B. Boots, and F. Ramos. Stein variational model predictive control. In *Conference on Robot Learning*, 2020.

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

[67] Y. Song and D. Scaramuzza. Policy search for model predictive control with application to agile drone flight. *IEEE Transactions on Robotics*, 2022.

[68] F. Berkenkamp and A. P. Schoellig. Safe and robust learning control with Gaussian processes. In *European Control Conference*, 2015.

[69] J. Kocijan, R. Murray-Smith, C. Rasmussen, and A. Girard. Gaussian process model based predictive control. In *American Control Conference*, 2004.

[70] M. Deisenroth, C. Rasmussen, and J. Peters. Gaussian process dynamic programming. *Neurocomputing*, 2009.

[71] M. Maiworm, D. Limon, J. Maria Manzano, and R. Findeisen. Stability of Gaussian process learning based output feedback model predictive control. In *IFAC Conference on Nonlinear Model Predictive Control*, 2018.

[72] S. Mysore, B. Mabsout, R. Mancuso, and K. Saenko. Regularizing action policies for smooth control with reinforcement learning. In *IEEE International Conference on Robotics and Automation*, 2021.

[73] A. Raffin, J. Kober, and F. Stulp. Smooth exploration for robotic reinforcement learning. In *Conference on Robot Learning*, 2022.

[74] D. Korenkevych, A. R. Mahmood, G. Vasan, and J. Bergstra. Autoregressive policies for continuous control deep reinforcement learning. In *International Joint Conference on Artificial Intelligence*, 2019.

[75] F. Stulp and O. Sigaud. Path integral policy improvement with covariance matrix adaptation. In *International Conference on Machine Learning*, 2012.

[76] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *Journal of Machine Learning Research*, 2017.

[77] A. Abdolmaleki, B. Price, N. Lau, L. P. Reis, and G. Neumann. Deriving and improving CMA-ES with information geometric trust regions. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 2017.

[78] P. Klink, H. Abdulsamad, B. Belousov, C. D'Eramo, J. Peters, and J. Pajarinen. A probabilistic interpretation of self-paced learning with applications to reinforcement learning. *Journal of Machine Learning Research*, 2021.

[79] F. Muratore, C. Eilers, M. Gienger, and J. Peters. Data-efficient domain randomization with Bayesian optimization. *IEEE Robotics and Automation Letters*, 2021.

[80] E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *IEEE International Conference on Intelligent Robots and Systems*, 2012.

[81] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym, 2016.

[82] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Robotics: Science and Systems*, 2018.

[83] H. Furuta, T. Kozuno, T. Matsushima, Y. Matsuo, and S. S. Gu. Co-adaptation of algorithmic and implementational innovations in inference-based deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

[84] O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 2007.

[85] K. Asadi and M. L. Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, 2017.

[86] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou. Information-theoretic MPC for model-based reinforcement learning. In *IEEE International Conference on Robotics and Automation*, 2017.

[87] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 2020.

[88] P. Dutilleul. The MLE algorithm for the matrix normal distribution. *Journal of statistical computation and simulation*, 1999.

[89] L. Peng, H. Qian, Z. Shen, C. Zhang, and F. Li. Generative actor-critic: An off-policy algorithm using the push-forward model. *arXiv preprint arXiv:2105.03733*, 2021.

[90] S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[91] S. Särkkä and A. Solin. *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019.