Sample Efficient Path Integral Control under Uncertainty

Yunpeng Pan, Evangelos A. Theodorou, and Michail Kontitsis

Autonomous Control and Decision Systems Laboratory
Institute for Robotics and Intelligent Machines
School of Aerospace Engineering
Georgia Institute of Technology, Atlanta, GA 30332
{ypan37,evangelos.theodorou,kontitsis}@gatech.edu

Abstract

We present a data-driven stochastic optimal control framework that is derived using the path integral (PI) control approach. We find iterative control laws analytically without a priori policy parameterization based on probabilistic representation of the learned dynamics model. The proposed algorithm operates in a forward-backward sweep manner which differentiate it from other PI-related methods that perform forward sampling to find open-loop optimal controls. Our method uses significantly less sampled data to find analytic control laws compared to other approaches within the PI control family that rely on extensive sampling from given dynamics models or trials on physical systems in a model-free fashion. In addition, the learned controllers can be generalized to new tasks without re-sampling based on the compositionality theory for the linearly-solvable optimal control framework. We provide experimental results on three different systems and comparisons with state-of-the-art model-based methods to demonstrate the efficiency and generalizability of the proposed framework.

1 Introduction

Stochastic optimal control (SOC) is a general and powerful framework with applications in many areas of science and engineering. However, despite the broad applicability, solving SOC problems remains challenging for systems in high-dimensional continuous state action spaces. Various function approximation approaches to optimal control are available [1, 2] but usually sensitive to model uncertainty. Over the last decade, SOC based on exponential transformation of the value function has demonstrated remarkable applicability in solving real world control and planning problems. In control theory the exponential transformation of the value function was introduced [3, 4] over 3 decades ago. In the recent decade it has been explored in terms of path integral interpretations and theoretical generalizations [5, 6, 7, 8], discrete time formulations [9], and scalable RL/control algorithms [10, 11, 12, 13, 14]. The resulting stochastic optimal control frameworks are known as Path Integral (PI) control for continuous time, Kullback Leibler (KL) control for discrete time, or more generally Linearly Solvable Optimal Control [9, 15].

One of the most attractive characteristics of PI control is that optimal control problems can be solved with forward sampling of Stochastic Differential Equations (SDEs). While the process of sampling with SDEs is more scalable than numerically solving partial differential equations, it still suffers from the curse of dimensionality when performed in a naive fashion. One way to circumvent this problem is to parameterize policies [10, 11, 14] and then perform optimization with sampling. However, in this case one has to impose the structure of the policy a-priori, therefore restrict the possible optimal control solutions within the assumed parameterization. In addition, the optimized policy

parameters can not be generalized to new tasks. In general, model-free PI policy search approaches require a large number of samples from trials performed on real systems. The issue of sample inefficiency further restricts the applicability of PI control on physical systems with unknown dynamics.

Motivated by the aforementioned limitations, in this paper we introduce a sample efficient, model-based approach to PI control. Different from existing PI control approaches, our method combines the benefits of PI control theory [5, 6, 7] and probabilistic model-based reinforcement learning methodologies [16, 17]. The main characteristics of the our approach are summarized as follows

- It extends the PI control theory [5, 6, 7] to the case of uncertain systems. The structural constraint is enforced between the control cost and uncertainty of the learned dynamics, which can be viewed as a generalization of previous work [5, 6, 7].

- Different from parameterized PI controllers [10, 11, 14, 8], we find analytic control law without any policy parameterization.

- Rather than keeping a fixed control cost weight [5, 6, 7, 10, 18], or ignoring the constraint between control authority and noise level [11], in this work the control cost is adapted based on the explicit uncertainty of the learned dynamics model.

- The algorithm operates in a different manner compared to existing PI-related methods [5, 6, 7, 10, 18, 11, 12, 14, 8]. More precisely it consists of successive forward belief propagation and backward computation of optimal control law.

- The proposed model-based approach is significantly more sample efficient than sampling-based PI control [5, 6, 7, 18]. In RL setting our method is comparable to the state-of-the-art RL methods [17, 19] in terms of sample and computational efficiency.

- Thanks to the linearity of the backward Chapman-Kolmogorov PDE, the learned controllers can be generalized to new tasks without re-sampling by constructing composite controllers. In contrast, most policy search and trajectory optimization methods [10, 11, 14, 17, 19, 20, 21, 22] find policy parameters that can not be generalized.

2 Iterative Path Integral Control for a Class of Uncertain Systems

2.1 Problem formulation

We consider a nonlinear stochastic system described by the following differential equation

$$d\mathbf{x} = \big(\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\big)dt + \mathbf{B}d\boldsymbol{\omega}, \tag{1}$$

with state $\mathbf{x} \in \mathbb{R}^n$, control $\mathbf{u} \in \mathbb{R}^m$, and standard Brownian motion noise $\boldsymbol{\omega} \in \mathbb{R}^p$ with variance $\boldsymbol{\Sigma}_\omega$. $\mathbf{f}(\mathbf{x})$ is the unknown drift term (passive dynamics), $\mathbf{G}(\mathbf{x}) \in \mathbb{R}^{n \times m}$ is the control matrix and $\mathbf{B} \in \mathbb{R}^{n \times p}$ is the diffusion matrix. Given some previous control $\mathbf{u}^{old}$, we seek the optimal control correction term $\delta\mathbf{u}$ such that the total control $\mathbf{u} = \mathbf{u}^{old} + \delta\mathbf{u}$. The original system can be rewritten as

$$d\mathbf{x} = \big(\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})(\mathbf{u}^{old} + \delta\mathbf{u})\big)dt + \mathbf{B}d\boldsymbol{\omega} = \underbrace{\big(\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}^{old}\big)}_{\tilde{\mathbf{f}}(\mathbf{x}, \mathbf{u}^{old})} dt + \mathbf{G}(\mathbf{x})\delta\mathbf{u}dt + \mathbf{B}d\boldsymbol{\omega}.$$

In this work we assume the dynamics based on the previous control can be represented by Gaussian processes (GP) such that

$$\mathbf{f}_{\mathbb{GP}}(\mathbf{x}) = \tilde{\mathbf{f}}(\mathbf{x}, \mathbf{u}^{old})dt + \mathbf{B}d\boldsymbol{\omega}, \tag{2}$$

where $\mathbf{f}_{\mathbb{GP}}$ is the GP representation of the biased drift term under the previous control. Now the original physical system (1) becomes the following system with uncertainty

$$d\mathbf{x} = \mathbf{f}_{\mathbb{GP}}dt + \mathbf{G}\delta\mathbf{u}dt, \qquad \mathbf{f}_{\mathbb{GP}} \sim \mathcal{GP}(\boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f). \tag{3}$$

For the GP model we use a standard prior of zero mean and covariance function $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_s^2 \exp(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}\mathbf{W}(\mathbf{x}_i - \mathbf{x}_j)) + \mathbf{B}\boldsymbol{\Sigma}_\omega\mathbf{B}^{\mathrm{T}}$, with $\sigma_s, \mathbf{W}$ the hyper-parameters. Samples over $\mathbf{f}_{\mathbb{GP}}$ can be drawn using an independent and identically distributed Gaussian variable Ω such that,

$$\tilde{\mathbf{f}}_{\mathbb{GP}} = \mu_f + \mathbf{L}_f\Omega \tag{4}$$

2

where $\mathbf{L}_f$ is obtained using Cholesky factorization such that $\boldsymbol{\Sigma}_f = \mathbf{L}_f \mathbf{L}_f^{\mathrm{T}}$. Notice that generally Ω is an infinite dimensional vector and we can use the same sample during learning [23]. Without loss of generality we assume Ω to be the standard zero-mean Brownian motion. For the rest of the paper we use simplified notations with subscripts indicating the time step. The discrete-time representation of the system is $\mathbf{x}_{t+\mathrm{d}t} = \mathbf{x}_t + \boldsymbol{\mu}_{ft} + \mathbf{G}_t \delta \mathbf{u}_t \mathrm{d}t + \mathbf{L}_{ft}\Omega_t \sqrt{\mathrm{d}t}$, and the conditional probability of $\mathbf{x}_{t+\mathrm{d}t}$ given $\mathbf{x}_t$ and $\delta \mathbf{u}_t$ is a Gaussian $\mathrm{p}\big(\mathbf{x}_{t+\mathrm{d}t}|\mathbf{x}_t, \delta \mathbf{u}_t\big) \sim \mathcal{N}\big(\boldsymbol{\mu}_{t+\mathrm{d}t}, \boldsymbol{\Sigma}_{t+\mathrm{d}t}\big)$, where $\boldsymbol{\mu}_{t+\mathrm{d}t} = \mathbf{x}_t + \boldsymbol{\mu}_{ft} + \mathbf{G}_t \delta \mathbf{u}_t$ and $\boldsymbol{\Sigma}_{t+\mathrm{d}t} = \boldsymbol{\Sigma}_{ft}$. In this paper we consider a finite-horizon stochastic optimal control problem

$$J(\mathbf{x}_0) = \mathbb{E}\Big[q(\mathbf{x}_T) + \int_{t=0}^T \mathcal{L}(\mathbf{x}_t, \delta \mathbf{u}_t)\mathrm{d}t\Big],$$

where the immediate cost is defined as $\mathcal{L}(\mathbf{x}_t, \mathbf{u}_t) = q(\mathbf{x}_t) + \frac{1}{2}\delta \mathbf{u}_t^{\mathrm{T}} \mathbf{R}_t \delta \mathbf{u}_t$, and $q(\mathbf{x}_t) = (\mathbf{x}_t - \mathbf{x}_t^d)^{\mathrm{T}} \mathbf{Q}(\mathbf{x}_t - \mathbf{x}_t^d)$ is a quadratic cost function where $\mathbf{x}_t^d$ is the desired state. $\mathbf{R}_t = \mathbf{R}(\mathbf{x}_t)$ is a state-dependent positive definite weight matrix. Next we show the linearized Hamilton-Jacobi-Bellman equation for this class of optimal control problems.

2.2 Linearized Hamilton-Jacobi-Bellman equation for uncertain dynamics

At each iteration the goal is to find the optimal control update $\delta \mathbf{u}_t$ that minimizes the value function

$$V(\mathbf{x}_t, t) = \min_{\delta \mathbf{u}_t} \mathbb{E}\Big[\mathcal{L}(\mathbf{x}_t, \delta \mathbf{u}_t) + V(\mathbf{x}_t + \mathrm{d}\mathbf{x}_t, t + \mathrm{d}t)|\mathbf{x}_t\Big]. \tag{5}$$

The above equation is the Bellman equation. By taking the limit of $\mathrm{d}t \to 0$ we obtain the Hamilton-Jacobi-Bellman (HJB) equation (detailed derivation is skipped):

$$-\partial_t V_t = \min_{\delta \mathbf{u}_t}(q_t + \frac{1}{2}\delta \mathbf{u}_t^{\mathrm{T}} \mathbf{R}_t \delta \mathbf{u}_t + (\boldsymbol{\mu}_{ft} + \mathbf{G}_t \delta \mathbf{u}_t)^{\mathrm{T}} \nabla_{\mathbf{x}} V_t + \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_{ft} \nabla_{\mathbf{x}\mathbf{x}} V_t)).$$

To find the optimal control update, we take gradient of the above expression (inside the parentheses) with respect to $\delta \mathbf{u}_t$ and set to 0. This yields $\delta \mathbf{u}_t = -\mathbf{R}_t^{-1} \mathbf{G}_t^{\mathrm{T}} \nabla_{\mathbf{x}} V_t$. Inserting this expression into the HJB equation yields the following nonlinear and second order PDE

$$-\partial_t V_t = q_t + (\nabla_{\mathbf{x}} V_t)^{\mathrm{T}} \boldsymbol{\mu}_{ft} - \frac{1}{2}(\nabla_{\mathbf{x}} V_t)^{\mathrm{T}} \mathbf{G}_t \mathbf{R}^{-1} \mathbf{G}_t^{\mathrm{T}} \nabla_{\mathbf{x}} V_t + \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_{ft} \nabla_{\mathbf{x}\mathbf{x}} V_t). \tag{6}$$

In order to solve the above PDE we use the exponential transformation of the value function $V_t = -\lambda \log \Psi_t$, where $\Psi_t = \Psi(\mathbf{x}_t)$ is called the *desirability* of $\mathbf{x}_t$. The corresponding partial derivatives can be found as $\partial_t V_t = -\frac{\lambda}{\Psi_t}\partial_t \Psi_t$, $\nabla_{\mathbf{x}} V_t = -\frac{\lambda}{\Psi_t}\nabla_{\mathbf{x}} \Psi_t$ and $\nabla_{\mathbf{x}\mathbf{x}} V_t = \frac{\lambda}{\Psi_t^2}\nabla_{\mathbf{x}} \Psi_t \nabla_{\mathbf{x}} \Psi_t^{\mathrm{T}} - \frac{\lambda}{\Psi_t}\nabla_{\mathbf{x}\mathbf{x}} \Psi_t$. Inserting these terms to (6) results in

$$\frac{\lambda}{\Psi_t}\partial_t \Psi_t = q_t - \frac{\lambda}{\Psi_t}(\nabla_{\mathbf{x}} \Psi_t)^{\mathrm{T}} \boldsymbol{\mu}_{ft} - \frac{\lambda^2}{2\Psi_t^2}(\nabla_{\mathbf{x}} \Psi_t)^{\mathrm{T}} \mathbf{G}_t \mathbf{R}_t^{-1} \mathbf{G}_t^{\mathrm{T}} \nabla_{\mathbf{x}} \Psi_t + \frac{\lambda}{2\Psi_t^2}\mathrm{Tr}((\nabla_{\mathbf{x}} \Psi_t)^{\mathrm{T}} \boldsymbol{\Sigma}_{ft} \nabla_{\mathbf{x}} \Psi_t) - \frac{\lambda}{2\Psi_t}\mathrm{Tr}(\nabla_{\mathbf{x}\mathbf{x}} \Psi_t \boldsymbol{\Sigma}_{ft}).$$

The quadratic terms $\nabla_{\mathbf{x}} \Psi_t$ will cancel out under the assumption of $\lambda \mathbf{G}_t \mathbf{R}_t^{-1} \mathbf{G}_t^{\mathrm{T}} = \boldsymbol{\Sigma}_{ft}$. This constraint is different from existing works in path integral control [5, 6, 7, 10, 18, 8] where the constraint is enforced between the additive noise covariance and control authority, more precisely $\lambda \mathbf{G}_t \mathbf{R}_t^{-1} \mathbf{G}_t^{\mathrm{T}} = \mathbf{B}\boldsymbol{\Sigma}_{\omega}\mathbf{B}^{\mathrm{T}}$. The new constraint enables an adaptive update of control cost weight based on explicit uncertainty of the learned dynamics. In contrast, most existing works use a fixed control cost weight [5, 6, 7, 10, 18, 12, 14, 8]. This condition also leads to more exploration (more aggressive control) under high uncertainty and less exploration with more certain dynamics. In addition, unlike existing works in PI control, here we do not need to assume the noise to act in the same subspace of control. Given the aforementioned assumption, the above PDE is simplified as

$$\partial_t \Psi_t = \frac{1}{\lambda}q_t \Psi_t - \boldsymbol{\mu}_{ft}^{\mathrm{T}} \nabla_{\mathbf{x}} \Psi_t - \frac{1}{2}\mathrm{Tr}(\nabla_{\mathbf{x}\mathbf{x}} \Psi_t \boldsymbol{\Sigma}_{ft}), \tag{7}$$

subject to the terminal condition $\Psi_T = \exp(-\frac{1}{\lambda}q_T)$. The resulting Chapman-Kolmogorov PDE (7) is linear. In general, solving (7) analytically is intractable for nonlinear systems and cost functions. We apply the Feynman-Kac formula which gives a probabilistic representation of the solution of the linear PDE (7)

$$\Psi_t = \lim_{\mathrm{d}t \to 0} \int \mathrm{p}(\tau_t|\mathbf{x}_t) \exp\Big(-\frac{1}{\lambda}(\sum_{j=t}^T q_j \mathrm{d}t)\Big)\Psi_T \mathrm{d}\tau_t, \tag{8}$$

3

where τ_t is the state trajectory from time t to T. The optimal control is obtained as

$$\mathbf{G}_t \delta \hat{\mathbf{u}}_t = -\mathbf{G}_t \mathbf{R}_t^{-1} \mathbf{G}_t^{\mathrm{T}} (\nabla_{\mathbf{x}} V_t) = \lambda \mathbf{G}_t \mathbf{R}_t^{-1} \mathbf{G}_t^{\mathrm{T}} \Big(\frac{\nabla_{\mathbf{x}} \Psi_t}{\Psi_t} \Big) = \mathbf{\Sigma}_{ft} \Big(\frac{\nabla_{\mathbf{x}} \Psi_t}{\Psi_t} \Big)$$

$$\Longrightarrow \hat{\mathbf{u}}_t = \mathbf{u}_t^{old} + \delta \hat{\mathbf{u}}_t = \mathbf{u}_t^{old} + \mathbf{G}_t^{-1} \mathbf{\Sigma}_{ft} \Big(\frac{\nabla_{\mathbf{x}} \Psi_t}{\Psi_t} \Big). \tag{9}$$

Rather than computing $\nabla_{\mathbf{x}} \Psi_t$ and Ψ_t, the optimal control $\hat{\mathbf{u}}_t$ can be approximated based on path costs of sampled trajectories. Next we briefly review some of the existing approaches.

2.3 Related works

According to the path integral control theory [5, 6, 7, 10, 18, 8], the stochastic optimal control problem becomes an approximation problem of a path integral (8). This problem can be solved by forward sampling of the uncontrolled ($\mathbf{u} = 0$) SDE (1). The optimal control $\hat{\mathbf{u}}_t$ is approximated based on path costs of sampled trajectories. Therefore the computation of optimal controls becomes a forward process. More precisely, when the control and noise act in the same subspace, the optimal control can be evaluated as the weighted average of the noise $\hat{\mathbf{u}}_t = \mathbb{E}_{\mathrm{p}(\tau_t | \mathbf{x}_t)} \big[\mathrm{d} \boldsymbol{\omega}_t \big]$, where the probability of a trajectory is $\mathrm{p}(\tau_t | \mathbf{x}_t) = \frac{\exp(-\frac{1}{\lambda} S(\tau_t | \mathbf{x}_t))}{\int \exp(-\frac{1}{\lambda} S(\tau_t | \mathbf{x}_t)) \mathrm{d}\tau}$, and $S(\tau_t | \mathbf{x}_t)$ is defined as the path cost computed by performing forward sampling. However, these approaches require a large amount of samples from a given dynamics model, or extensive trials on physical systems when applied in model-free reinforcement learning settings. In order to improve sample efficiency, a nonparametric approach was developed by representing the desirability Ψ_t in terms of linear operators in a reproducing kernel Hilbert space (RKHS) [12]. As a model-free approach, it allows sample re-use but relies on numerical methods to estimate the gradient of desirability, i.e., $\nabla_{\mathbf{x}} \Psi_t$, which can be computationally expensive. On the other hand, computing the analytic expressions of the path integral embedding is intractable and requires exact knowledge of the system dynamics. Furthermore, the control approximation is based on samples from the uncontrolled dynamics, which is usually not sufficient for highly nonlinear systems.

Another class of PI-related method is based on policy parameterization. Notable approaches include PI^2 [10], PI^2-CMA [11], PI-REPS[14] and recently developed state-dependent PI[8]. The limitations of these methods are: 1) They do not take into account model uncertainty in the passive dynamics $\mathbf{f}(\mathbf{x})$. 2) The imposed policy parameterizations restrict optimal control solutions. 3) The optimized policy parameters can not be generalized to new tasks. A brief comparison of some of these methods can be found in Table 1. Motivated by the challenge of combining sample efficiency and generalizability, next we introduce a probabilistic model-based approach to compute the optimal control (9) analytically.

	PI [5, 6, 7], iterative PI [18]	PI^2[10], PI^2-CMA [11]	PI-REPS[14]	State feedback PI[8]	Our method
Structural constraint	$\lambda \mathbf{G}_t \mathbf{R}_t^{-1} \mathbf{G}_t^{\mathrm{T}} = \mathbf{B} \mathbf{\Sigma}_\omega \mathbf{B}^{\mathrm{T}}$	same as PI	same as PI	same as PI	$\lambda \mathbf{G} \mathbf{R}^{-1} \mathbf{G}^{\mathrm{T}} = \mathbf{\Sigma}_f$
Dynamics model	model-based	model-free	model-based	model-based	model-based
Policy parameterization	No	Yes	Yes	Yes	No

Table 1: Comparison with some notable and recent path integral-related approaches.

3 Proposed Approach

3.1 Analytic path integral control: a forward-backward scheme

In order to derive the proposed framework, firstly we learn the function $\mathbf{f}_{\mathbb{GP}}(\mathbf{x}_t) = \tilde{\mathbf{f}}(\mathbf{x}, \mathbf{u}^{old})\mathrm{d}t + \mathbf{B}\mathrm{d}\boldsymbol{\omega}$ from sampled data. The function can be viewed as an inference with the goal of inferring the state transition $\mathrm{d}\tilde{\mathbf{x}}_t = \mathbf{f}_{\mathbb{GP}}(\mathbf{x}_t)$. Given a sequence of states $\mathbf{X} = \{\mathbf{x}_0, \dots \mathbf{x}_T\}$, and the corresponding state transition $\mathrm{d}\tilde{\mathbf{X}} = \{\mathrm{d}\tilde{\mathbf{x}}_0, \dots, \mathrm{d}\tilde{\mathbf{x}}_T\}$, a GP is completely defined by a mean function and a covariance function. The joint distribution of the observed state and state transition can be written as $\mathrm{p}\left(\begin{array}{c} \mathrm{d}\tilde{\mathbf{X}} \\ \mathrm{d}\mathbf{x}^* \end{array} \right) \sim \mathcal{N}\left(0, \left[\begin{array}{cc} \mathbf{K}(\mathbf{X}, \mathbf{X}) & \mathbf{K}(\mathbf{X}, \mathbf{x}^*) \\ \mathbf{K}(\mathbf{x}^*, \mathbf{X}) & \mathbf{K}(\mathbf{x}^*, \mathbf{x}^*) \end{array} \right] \right)$. The kernel function has been defined in Sec.2.1, which can be interpreted as a similarity measure of random variables. More specifically, if the training input $\mathbf{x}_i$ and $\mathbf{x}_j$ are close to each other in the kernel space, their outputs $\mathrm{d}\mathbf{x}_i$ and $\mathrm{d}\mathbf{x}_j$ are highly correlated. The posterior distribution can be obtained by conditioning the GP on the function value

at an observed state. In this work we make the standard assumption of independent outputs (no correlation between each output dimension).

To propagate the GP-based dynamics over a trajectory of time horizon T we employ the moment matching approach [24, 17] to compute the predictive distribution. Given an input distribution over the state $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, the predictive distribution over the state at $t+\mathrm{d}t$ can be computed as a Gaussian $\mathrm{p}(\mathbf{x}_{t+\mathrm{d}t}) \sim \mathcal{N}(\boldsymbol{\mu}_{t+\mathrm{d}t}, \boldsymbol{\Sigma}_{t+\mathrm{d}t})$ such that

$$\boldsymbol{\mu}_{t+\mathrm{d}t} = \boldsymbol{\mu}_t + \boldsymbol{\mu}_{ft}, \quad \boldsymbol{\Sigma}_{t+\mathrm{d}t} = \boldsymbol{\Sigma}_t + \boldsymbol{\Sigma}_{ft} + \mathbb{COV}[\mathbf{x}_t, \mathrm{d}\tilde{\mathbf{x}}_t] + \mathbb{COV}[\mathrm{d}\tilde{\mathbf{x}}_t, \mathbf{x}_t]. \tag{10}$$

The above formulation is used to approximate one-step transition probabilities over the trajectory. Details regarding the moment matching method can be found in [24, 17]. All mean and variance terms can be computed analytically. The hyper-parameters $\sigma_s, \mathbf{W}$ can be learned by maximizing the log-likelihood of the training outputs given the inputs [25]. Given the approximation of transition probability (10), we now introduce a Bayesian nonparametric formulation of path integral control based on probabilistic representation of the dynamics. Firstly we forward propagate to obtain the Gaussian belief (predictive mean and covariance of the state) over the trajectory. Since the exponential transformation of the state cost $\exp(-\frac{1}{\lambda}q(\mathbf{x})\mathrm{d}t)$ is an unnormalized Gaussian $\mathcal{N}(\mathbf{x}^d, \frac{2\lambda}{\mathrm{d}t}\mathbf{Q}^{-1})$. We can evaluate the following integral analytically

$$\Phi_j = \int \mathrm{p}(\mathbf{x}_j)\exp(-\frac{1}{\lambda}q_j\mathrm{d}t)\mathrm{d}\mathbf{x}_j = \left|\mathbf{I} + \frac{\mathrm{d}t}{2\lambda}\boldsymbol{\Sigma}_j\mathbf{Q}\right|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\boldsymbol{\mu}_j - \mathbf{x}_j^d)^{\mathrm{T}}\frac{\mathrm{d}t}{2\lambda}\mathbf{Q}(\mathbf{I} + \frac{\mathrm{d}t}{2\lambda}\lambda\boldsymbol{\Sigma}_j\mathbf{Q})^{-1}(\boldsymbol{\mu}_j - \mathbf{x}_j^d)\right)$$
$$\tag{11}$$

for $j = t + \mathrm{d}t, ..., T$. Based on the above expression, next we compute the path integral (8), which is the major computational challenge of PI-related approaches. Since we use Gaussian belief propagation based on (10) instead of explicitly sampling from the corresponding SDE, we assume that the predictive distribution $\mathrm{p}(\mathbf{x}_j) = \mathrm{p}(\mathbf{x}_j | \boldsymbol{\mu}_{j-\mathrm{d}t}, \boldsymbol{\Sigma}_{j-\mathrm{d}t})$ is a good approximation of the conditional distribution $\mathrm{p}(\mathbf{x}_j | \mathbf{x}_{j-\mathrm{d}t})$. Therefore the analytic approximation of path integral Ψ_t (8) can be rewritten in the following recursive form

$$\Psi_t = \int \mathrm{p}\left(\tau_t | \mathbf{x}_t\right)\exp\left(-\frac{1}{\lambda}\left(\sum_{j=t}^{T}q_j\mathrm{d}t\right)\right)\Psi_T\mathrm{d}\tau_t$$
$$= \int ... \int \mathrm{p}\left(\mathbf{x}_{T-\mathrm{d}t}\right)\exp\left(-\frac{1}{\lambda}q_{T-\mathrm{d}t}\mathrm{d}t\right)\underbrace{\int \mathrm{p}\left(\mathbf{x}_T\right)\exp\left(-\frac{1}{\lambda}q_T\mathrm{d}t\right)\Psi_T\mathrm{d}\mathbf{x}_T}_{\Psi_{T-\mathrm{d}t}=\Phi_T\Psi_T}\mathrm{d}\mathbf{x}_{T-\mathrm{d}t}...\mathrm{d}\mathbf{x}_{t+\mathrm{d}t}$$
$$\underbrace{\phantom{\int ... \int \mathrm{p}\left(\mathbf{x}_{T-\mathrm{d}t}\right)\exp\left(-\frac{1}{\lambda}q_{T-\mathrm{d}t}\mathrm{d}t\right)\int \mathrm{p}\left(\mathbf{x}_T\right)\exp\left(-\frac{1}{\lambda}q_T\mathrm{d}t\right)\Psi_T\mathrm{d}\mathbf{x}_T}}_{\Psi_{T-2\mathrm{d}t}=\Phi_{T-\mathrm{d}t}\Psi_{T-\mathrm{d}t}}$$
$$= \Phi_{t+\mathrm{d}t}\Psi_{t+\mathrm{d}t} \tag{12}$$

Next we evaluate the desirability Ψ_t backward in time by successive computation using the above recursive expression . To obtain the optimal control law $\hat{\mathbf{u}}_t$ (9), we need gradients of the desirability function with respect to the state, which can be computed backward in time as well, more precisely we compute the gradient of the recursive expression (12)

$$\nabla_{\mathbf{x}}\Psi_{T-\mathrm{d}t} = \Psi_T\nabla_{\mathbf{x}}\Phi_T + \Phi_T\nabla_{\mathbf{x}}\Psi_T, \quad \cdots \quad , \nabla_{\mathbf{x}}\Psi_t = \Psi_{t+\mathrm{d}t}\nabla_{\mathbf{x}}\Phi_{t+\mathrm{d}t} + \Phi_{t+\mathrm{d}t}\nabla_{\mathbf{x}}\Psi_{t+\mathrm{d}t}, \tag{13}$$

where all gradients are evaluated analytically. For $j = t + \mathrm{d}t, ..., T$, given the expression in (11) we compute the gradient terms in (13) as

$$\nabla_{\mathbf{x}}\Phi_j = \frac{\mathrm{d}\Phi_j}{\mathrm{dp}(\mathbf{x}_j)}\frac{\mathrm{dp}(\mathbf{x}_j)}{\mathrm{d}\mathbf{x}_t} = \frac{\partial\Phi_j}{\partial\boldsymbol{\mu}_j}\frac{\mathrm{d}\boldsymbol{\mu}_j}{\mathrm{d}\mathbf{x}_t} + \frac{\partial\Phi_j}{\partial\boldsymbol{\Sigma}_j}\frac{\mathrm{d}\boldsymbol{\Sigma}_j}{\mathrm{d}\mathbf{x}_t}, \text{ where } \frac{\partial\Phi_j}{\partial\boldsymbol{\mu}_j} = \Phi_j(\boldsymbol{\mu}_j - \mathbf{x}_j^d)^T\frac{\mathrm{d}t}{2\lambda}\mathbf{Q}(\mathbf{I} + \frac{\mathrm{d}t}{2\lambda}\lambda\boldsymbol{\Sigma}_j\mathbf{Q})^{-1},$$

$$\frac{\partial\Phi_j}{\partial\boldsymbol{\Sigma}_j} = \frac{\Phi_j}{2}\left(\frac{\mathrm{d}t}{2\lambda}\mathbf{Q}(\mathbf{I} + \frac{\mathrm{d}t}{2\lambda}\lambda\boldsymbol{\Sigma}_j\mathbf{Q})^{-1}(\boldsymbol{\mu}_j - \mathbf{x}_j^d)(\boldsymbol{\mu}_j - \mathbf{x}_j^d)^T - \mathbf{I}\right)\frac{\mathrm{d}t}{2\lambda}\mathbf{Q}(\mathbf{I} + \frac{\mathrm{d}t}{2\lambda}\lambda\boldsymbol{\Sigma}_j\mathbf{Q})^{-1}, \text{ and }$$

$$\frac{\mathrm{d}\{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}}{\mathrm{d}\mathbf{x}_t} = \left\{\frac{\partial\boldsymbol{\mu}_j}{\partial\boldsymbol{\mu}_{j-\mathrm{d}t}}\frac{\mathrm{d}\boldsymbol{\mu}_{j-\mathrm{d}t}}{\mathrm{d}\mathbf{x}_t} + \frac{\partial\boldsymbol{\mu}_j}{\partial\boldsymbol{\Sigma}_{j-\mathrm{d}t}}\frac{\mathrm{d}\boldsymbol{\Sigma}_{j-\mathrm{d}t}}{\mathrm{d}\mathbf{x}_t}, \frac{\partial\boldsymbol{\Sigma}_j}{\partial\boldsymbol{\mu}_{j-\mathrm{d}t}}\frac{\mathrm{d}\boldsymbol{\mu}_{j-\mathrm{d}t}}{\mathrm{d}\mathbf{x}_t} + \frac{\partial\boldsymbol{\Sigma}_j}{\partial\boldsymbol{\Sigma}_{j-\mathrm{d}t}}\frac{\mathrm{d}\boldsymbol{\Sigma}_{j-\mathrm{d}t}}{\mathrm{d}\mathbf{x}_t}\right\}.$$

The partial derivatives $\frac{\partial\boldsymbol{\mu}_j}{\partial\boldsymbol{\mu}_{j-\mathrm{d}t}}, \frac{\partial\boldsymbol{\mu}_j}{\partial\boldsymbol{\Sigma}_{j-\mathrm{d}t}}, \frac{\partial\boldsymbol{\Sigma}_j}{\partial\boldsymbol{\mu}_{j-\mathrm{d}t}}, \frac{\partial\boldsymbol{\Sigma}_j}{\partial\boldsymbol{\Sigma}_{j-\mathrm{d}t}}$ can be computed similarly as in [17]. We compute all gradients using this scheme without any numerical method (finite differences, etc.). Therefore the computational efficiency is significantly improved compared to model-free frameworks. Given Ψ_t and $\nabla_{\mathbf{x}}\Psi_t$, the optimal control takes a analytic form as in eq.(9). Since Ψ_t and

$\nabla_{\mathbf{x}}\Psi_t$ are explicit functions of $\mathbf{x}_t$, the resulting control law is essentially different from the feed-forward control in most sampling-based path integral control frameworks [5, 6, 7, 10, 18] as well as the parameterized state feedback PI control policies [14, 8]. Notice that at current time step t, we update the control sequence $\hat{\mathbf{u}}_{t,...,T}$ using the presented forward-backward propagation scheme. Only $\hat{\mathbf{u}}_t$ is applied to the system to move to the next step, while the controls $\hat{\mathbf{u}}_{t+\mathrm{d}t,...,T}$ are used for control update at future steps. The data recorded at each time step is incorporated to update the GP model of the dynamics. A summary of the proposed algorithm is shown in **Algorithm** 1.

Algorithm 1 Sample efficient path integral control under uncertain dynamics

---

1: **Initialization:** Apply random controls $\hat{\mathbf{u}}_{0,..,T}$ to the physical system (1), record data.
2: **repeat**
3:     **for** t=0:T **do**
4:         Incorporate sampled data to learn GP dynamics model.
5:         **repeat**
6:             Forward Gaussian belief propagation (10) using controls $\mathbf{u}^{old}_{t,..,T} = \hat{\mathbf{u}}_{t,..,T}$.
7:             Backward computation of optimal control updates $\delta\hat{\mathbf{u}}_{t,..,T}$, see (12)(13)(9).
8:             Update optimal controls $\hat{\mathbf{u}}_{t,..,T} = \mathbf{u}^{old}_{t,..,T} + \delta\hat{\mathbf{u}}_{t,..,T}$.
9:         **until** Convergence.
10:         Apply optimal control $\hat{\mathbf{u}}_t$ to the system. Move one step forward and record data.
11:     **end for**
12: **until** Task learned.

3.2 Generalization to unlearned tasks without sampling

In this section we describe how to generalize the learned controllers for new (unlearned) tasks without any interaction with the real system. The proposed approach is based on the compositionality theory [26] in linearly solvable optimal control (LSOC). We use superscripts to denote previously learned task indexes. Firstly we define a distance measure between the new target $\bar{\mathbf{x}}^d$ and old targets $\mathbf{x}^{dk}, k = 1, .., K$, i.e., a Gaussian kernel

$$\omega^k = \exp\left(-\frac{1}{2}(\bar{\mathbf{x}}^d - \mathbf{x}^{dk})^{\mathrm{T}}\mathbf{P}(\bar{\mathbf{x}}^d - \mathbf{x}^{dk})\right), \tag{14}$$

where $\mathbf{P}$ is a diagonal matrix (kernel width). The composite terminal cost $\bar{q}(\mathbf{x}_T)$ for the new task becomes

$$\bar{q}(\mathbf{x}_T) = -\lambda \log\left(\frac{\sum_{k=1}^{K}\omega^k \exp(-\frac{1}{\lambda}q^k(\mathbf{x}_T))}{\sum_{k=1}^{K}\omega^k}\right), \tag{15}$$

where $q^k(\mathbf{x}_T)$ is the terminal cost for old tasks. For conciseness we define a normalized distance measure $\tilde{\omega}^k = \frac{\omega^k}{\sum_{k=1}^{K}\omega^k}$, which can be interpreted as a probability weight. Based on (15) we have the composite terminal desirability for the new task which is a linear combination of Ψ_T^k

$$\bar{\Psi}_T = \exp\left(-\frac{1}{\lambda}\bar{q}(\mathbf{x}_T)\right) = \sum_{k=1}^{K}\tilde{\omega}^k \Psi_T^k. \tag{16}$$

Since Ψ_t^k is the solution to the linear Chapman-Kolmogorov PDE (7), the linear combination of desirability (16) holds everywhere from t to T as long as it holds on the boundary (terminal time step). Therefore we obtain the composite control

$$\bar{\mathbf{u}}_t = \sum_{k=1}^{K}\frac{\tilde{\omega}^k \Psi_t^k}{\sum_{k=1}^{K}\tilde{\omega}^k \Psi_t^k}\hat{\mathbf{u}}_t^k. \tag{17}$$

The composite control law in (17) is essentially different from an interpolating control law[26]. It enables sample-free controllers that constructed from learned controllers for different tasks. This scheme can not be adopted in policy search or trajectory optimization methods such as [10, 11, 14, 17, 19, 20, 21, 22]. We will demonstrate the effectiveness of this control law in the next section.

4 Experiments and Analysis

We consider 3 simulated RL tasks: cart-pole (CP) swing up, double pendulum on a cart (DPC) swing up, and PUMA-560 robotic arm reaching. The CP and DPC systems consist of a cart and a single/double-link pendulum. The tasks are to swing-up the single/double-link pendulum from the initial position (point down). Both CP and DPC are under-actuated systems with only one control acting on the cart. PUMA-560 is a 3D robotic arm that has 12 state dimensions, 6 degrees of freedom with 6 actuators on the joints. The task is to steer the end-effector to the desired position and orientation.

In order to demonstrate the performance, we compare the proposed control framework with three related methods: iterative path integral control [18] with known dynamics model, PILCO [17] and PDDP [19]. Iterative path integral control is a sampling-based stochastic control method. It is based on importance sampling using controlled diffusion process rather than passive dynamics used in standard path integral control [5, 6, 7]. Iterative PI control is used as a baseline with a given dynamics model. PILCO is a model-based policy search method that features state-of-the-art data efficiency in terms of number of trials required to learn a task. PILCO requires an extra optimizer (such as BFGS) for policy improvement. PDDP is a Gaussian belief space trajectory optimization approach. It performs dynamic programming based on local approximation of the learned dynamics and value function. Both PILCO and PDDP are applied with unknown dynamics. In this work we do not compare our method with model-free PI-related approaches such as [10, 11, 12, 14] since these methods would certainly cost more samples than model-based methods such as PILCO and PDDP. The reason for choosing these two methods for comparison is that our method adopts a similar model learning scheme while other state-of-the-art methods, such as [20] is based on a different model.

In **experiment 1** we demonstrate the sample efficiency of our method using the CP and DPC tasks. For both tasks we choose $T = 3$ and $\mathrm{d}t = 0.05$ (60 time steps per rollout). The iterative PI [18] with a given dynamics model uses $10^3/10^4$ (CP/DPC) sample rollouts per iteration and 500 iterations at each time step. We initialize PILCO and the proposed method by collecting 2/6 sample rollouts for CP/DPC tasks respectively. At each trial (on the true dynamics model), we use 1 sample rollout for PILCO and our method. PDDP uses 4/5 samples for initialization as well as at each trial for the CP/DPC tasks. Fig. 1 shows the results in terms of Ψ_T and computational time. For both tasks our method shows higher desirability (lower terminal state cost) at each trial, which indicates higher sample efficiency for task learning. However we assume partial information of the dynamics ($\mathbf{G}$ matrix) is given. While PILCO and PDDP perform optimization on entirely unknown dynamics. In many robotic systems $\mathbf{G}$ corresponds to the inverse of the inertia matrix, which can be identified based on data as well. In terms of computational efficiency, our method outperforms PILCO since we compute the optimal control update analytically, while PILCO solves large scale nonlinear optimization problems to obtain policy parameters. Our method is more computational expensive than PDDP because PDDP seeks local optimal controls that rely on linear approximations, while our method is a global optimal control approach. Despite the relatively higher computational burden than PDDP, our method offers reasonable efficiency in terms of the time required to reach the baseline performance.

In **experiment 2** we demonstrate the generalizability of the learned controllers to new tasks using the composite control law (17) based on the PUMA-560 system. We use $T = 2$ and $\mathrm{d}t = 0.02$ (100 time steps per rollout). First we learn 8 independent controllers using **Algorithm** 1. The target postures are shown in Fig. 2. For all tasks we initialize with 3 sample rollouts and 1 sample at each trial. Blue bars in Fig. 2b shows the desirabilities Ψ_T after 3 trials. Next we use the composite law (17) to construct controllers without re-sampling using 7 other controllers learned using **Algorithm** 1. For instance the composite controller for task#1 is found as $\bar{\mathbf{u}}_t^1 = \sum_{k=2}^{8} \frac{\tilde{\omega}^k \Psi_t^k}{\sum_{k=2}^{8} \tilde{\omega}^k \Psi_t^k} \hat{\mathbf{u}}_t^k$. The performance comparison of the composite controllers with controllers learned from trials is shown in Fig. 2. It can be seen that the composite controllers give close performance as independently learned controllers. The compositionality theory [26] generally does not apply to policy search methods and trajectory optimizers such as PILCO, PDDP, and other recent methods [20, 21, 22]. Our method benefits from the compositionality of control laws that can be applied for multi-task control without re-sampling. Alternatively, generalization can be achieved by imposing task-dependent policies [27]. However, this approach might restrict the choice of optimal controls given the assumed structure of control policy.
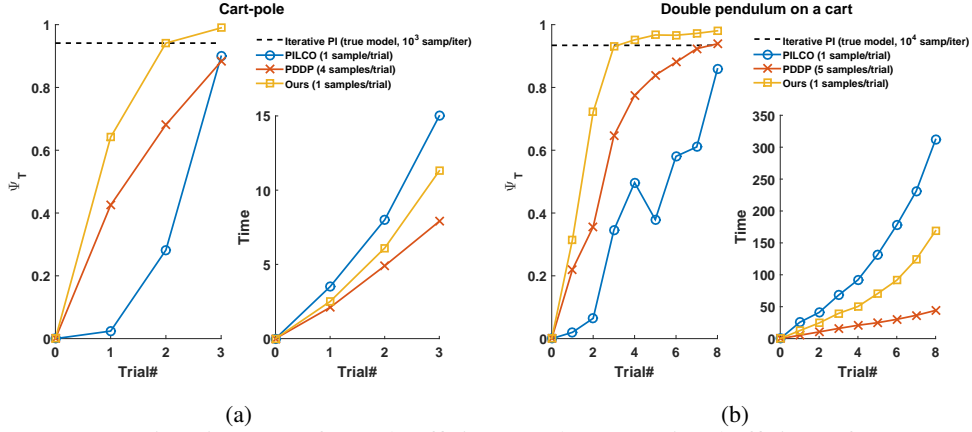
Figure 1: Comparison in terms of sample efficiency and computational efficiency for (a) cart-pole and (b) double pendulum on a cart swing-up tasks. Left subfigures show the terminal desirability Ψ_T (for PILCO and PDDP, Ψ_T is computed using terminal state costs) at each trial. Right subfigures show computational time (in minute) at each trial.
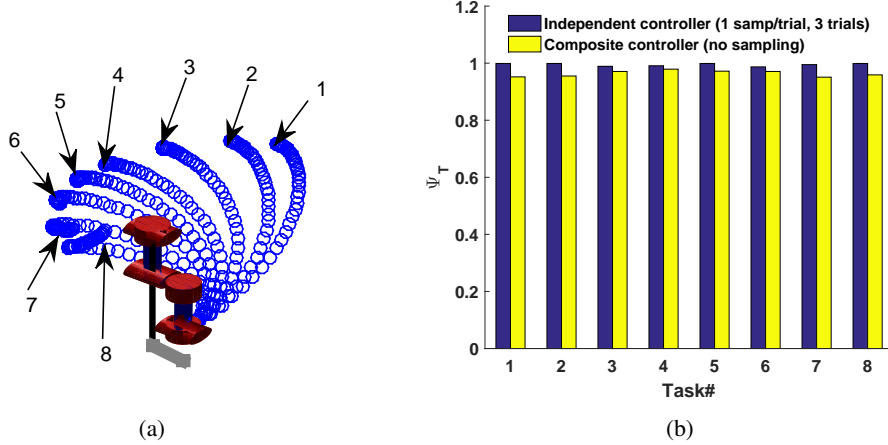

Figure 2: Resutls for the PUMA-560 tasks. (a) 8 tasks tested in this experiment. Each number indicates a corresponding target posture. (b) Comparison of the controllers learned independently from trials and the composite controllers without sampling. Each composite controller is obtained (17) from 7 other independent controllers learned from trials.

5 Conclusion and Discussion

We presented an iterative stochastic control framework that can find optimal controllers under uncertain dynamics using very small number of samples. This approach is closely related to the family of path integral (PI) control algorithms. Our method is based on a forward-backward optimization scheme, which differs significantly from current PI-related approaches. Moreover, it combines the attractive characteristics of probabilistic model-based reinforcement learning and linearly solvable optimal control theory. These characteristics include sample efficiency, optimality and generalizability. By iteratively updating the control laws based on probabilistic representation of the learned dynamics, our method demonstrated encouraging performance compared to state-of-the-art model-based methods. In addition, our method showed promising potential in performing multi-task control based on the compositionality of learned controllers. Besides the assumed structural constraint between control cost and uncertainty of the dynamics, the major limitation is that we have not taken into account the uncertainty in the control matrix $\mathbf{G}$. Future work will focus on further generalization of this framework and applications to real systems.

References

[1] D.P. Bertsekas and J.N. Tsitsiklis. Neuro-dynamic programming (optimization and neural computation series, 3). *Athena Scientific*, 7:15–23, 1996.

[2] A.G. Barto, W. Powell, J. Si, and D.C. Wunsch. Handbook of learning and approximate dynamic programming. 2004.

[3] W.H. Fleming. Exit probabilities and optimal stochastic control. *Applied Math. Optim*, 9:329–346, 1971.

[4] W. H. Fleming and H. M. Soner. *Controlled Markov processes and viscosity solutions*. Applications of mathematics. Springer, New York, 1st edition, 1993.

[5] H. J. Kappen. Linear theory for control of nonlinear stochastic systems. *Phys Rev Lett*, 95:200–201, 2005.

[6] H. J. Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics: Theory and Experiment*, 11:P11011, 2005.

[7] H. J. Kappen. An introduction to stochastic control theory, path integrals and reinforcement learning. *AIP Conference Proceedings*, 887(1), 2007.

[8] S. Thijssen and H. J. Kappen. Path integral control and state-dependent feedback. *Phys. Rev. E*, 91:032104, Mar 2015.

[9] E. Todorov. Efficient computation of optimal actions. *Proceedings of the national academy of sciences*, 106(28):11478–11483, 2009.

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

[11] F. Stulp and O. Sigaud. Path integral policy improvement with covariance matrix adaptation. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 281–288. ACM, 2012.

[12] K. Rawlik, M. Toussaint, and S. Vijayakumar. Path integral control by reproducing kernel hilbert space embedding. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, pages 1628–1634, 2013.

[13] Y. Pan and E. Theodorou. Nonparametric infinite horizon kullback-leibler stochastic control. In *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 1–8. IEEE, 2014.

[14] V. Gómez, H.J. Kappen, J. Peters, and G. Neumann. Policy search for path integral control. In *Machine Learning and Knowledge Discovery in Databases*, pages 482–497. Springer, 2014.

[15] K. Dvijotham and E Todorov. Linearly solvable optimal control. *Reinforcement learning and approximate dynamic programming for feedback control*, pages 119–141, 2012.

[16] M.P. Deisenroth, G. Neumann, and J. Peters. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2(1-2):1–142, 2013.

[17] M. Deisenroth, D. Fox, and C. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transsactions on Pattern Analysis and Machine Intelligence*, 27:75–90, 2015.

[18] E. Theodorou and E. Todorov. Relative entropy and free energy dualities: Connections to path integral and kl control. In *51st IEEE Conference on Decision and Control*, pages 1466–1473, 2012.

[19] Y. Pan and E. Theodorou. Probabilistic differential dynamic programming. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1907–1915, 2014.

[20] S. Levine and P. Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1071–1079, 2014.

[21] S. Levine and V. Koltun. Learning complex neural network policies with trajectory optimization. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 829–837, 2014.

[22] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. Trust region policy optimization. *arXiv preprint arXiv:1502.05477*, 2015.

[23] P. Hennig. Optimal reinforcement learning for gaussian systems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 325–333, 2011.

[24] J. Quinonero Candela, A. Girard, J. Larsen, and C. E. Rasmussen. Propagation of uncertainty in bayesian kernel models-application to multiple-step ahead forecasting. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003.

[25] C.K.I Williams and C.E. Rasmussen. *Gaussian processes for machine learning*. MIT Press, 2006.

[26] E. Todorov. Compositionality of optimal control laws. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1856–1864, 2009.

[27] M.P. Deisenroth, P. Englert, J. Peters, and D. Fox. Multi-task policy search for robotics. In *Proceedings of 2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.