An Iterative Path Integral Stochastic Optimal Control Approach for Learning Robotic Tasks

Evangelos Theodorou * **Freek Stulp** * **Jonas Buchli** **
Stefan Schaal *,***

* *Computational Learning and Motor Control Lab, University of Southern California, USA.*
** *Department of Advanced Robotics, Italian Institute of Technology.*
*** *ATR Computational Neuroscience Laboratories Kyoto 619-0288, Japan*

Abstract: Recent work on path integral stochastic optimal control theory Theodorou et al. (2010a); Theodorou (2011) has shown promising results in planning and control of nonlinear systems in high dimensional state spaces. The path integral control framework relies on the transformation of the nonlinear Hamilton Jacobi Bellman (HJB) partial differential equation (PDE) into a linear PDE and the approximation of its solution via the use of the Feynman Kac lemma. In this work, we are reviewing the generalized version of path integral stochastic optimal control formalism Theodorou et al. (2010a), used for optimal control and planing of stochastic dynamical systems with state dependent control and diffusion matrices. Moreover we present the iterative path integral control approach, the so called **P**olicy **I**mprovement with **P**ath **I**ntegrals or (PI^2) which is capable of scaling in high dimensional robotic control problems. Furthermore we present a convergence analysis of the proposed algorithm and we apply the proposed framework to a variety of robotic tasks. Finally with the goal to perform locomotion the iterative path integral control is applied for learning nonlinear limit cycle attractors with adjustable land scape.

Keywords: Path Integrals, Stochastic Optimal Control, Robotics

1. INTRODUCTION

The framework of nonlinear stochastic optimal control theory has been one of the most general control theoretic approaches with a variety of applications in domains that span from biology Todorov (2005), Li et al. (2004) and neuroscience Izawa et al. (2008) to vehicle and mobile robot control Papageorgiou and Bauschert (1994). There has been a broad applicability of nonlinear stochastic optimal control due to that fact that dynamical systems are usually highly nonlinear and stochastic. There are different sources of stochasticity and randomness that are related to the dynamics under control. For example, in case of neuromuscular systems stochasticity may come from the noisy neural commands while in humanoid and mobile robotics randomness may be caused due to noise in priopreceptive sensors such as odometers, gyros etc as well as contact with the environment.

⋆ This research was supported in part by National Science Foundation grants ECS-0325383, IIS-0312802, IIS-0082995, IIS-9988642, ECS-0326095, ANI-0224419, the DARPA program on Learning Locomotion, the Multidisciplinary Research Program of the Department of Defense (MURI N00014-00-1-0637), and the ATR Computational Neuroscience Laboratories. E.T. was supported by a Myronis Fellowship. F.S. was supported by a Research Fellowship from the German Research Foundation (DFG). J.B. was supported by an advanced researcher fellowship from the Swiss National Science Foundation.

One of the main issues with stochastic optimal control is that for the case of nonlinear systems, its solution requires the solution of a nonlinear and second order partial differential equation, the so called Hamilton Jacobi Bellman equation Stengel (1994); Jacobson and Mayne (1970). How to solve such partial differential equation especially for high dimensional state space models is still an open research problem. The challenges in solving this PDE have limited the use of stochastic optimal control to low dimensional control problems. In fact, in the area humanoid robot control Sciavicco and Siciliano (2000) where systems are nonlinear and they can have more than 35 degrees of freedom (= 70 states), the curse of dimensionality is the main obstacle in applying optimal control methods. In addition to the high dimensionality, accurate models of the underlying nonlinear robotic dynamics are not usually available.

Recent work on Path Integral stochastic optimal control Kappen (2007, 2005b,a) gave interesting insights into symmetry breaking phenomena while it provided conditions under which the nonlinear and second order HJB could be transformed into a linear PDE similar to the backward chapman Kolmogorov PDE. In Broek et al. (2008) the path integral stochastic optimal control was extended to the case of multi-agent dynamics. In all this work, the theory was developed for stochastic dynamical systems of low dimensioanality with control transition and diffusion matrices constant. Even though linear PDEs are easier to

be solved with the use of Feynamn - Kac lemmas this connection was not initially made in Kappen (2007, 2005b,a). In Theodorou et al. (2010a,b); Theodorou (2011)we have generalized the path integral control framework such that it could be applied to stochastic dynamics with state dependent control transition and diffusion matrices, while we have made use of the Feynman Kac lemma to approximate solution of the resulting linear PDE. Moreover we have proposed an iterative version of path integral control capable of scaling in high dimensional planning and control problems, the so called **P**olicy **I**mprovement with **P**ath **I**ntegrals or (PI^2) for short. In Buchli et al. (2010) PI^2 was applied to variable stiffness control which is an application equivalent to autonomously tuning PD gains in a 6DOF manipulator. In Stulp et al. (2010), PI^2 was able to learn a full-body humanoid motor skill in simulation, involving all 34-DOF of the robot.

In this paper, in Section 2, we are reviewing the generalized path integral stochastic optimal control (see Theodorou et al. (2010a); Theodorou (2011)). In Section 3, we develop the iterative version of path integral stochastic optimal control approach PI^2 and we present, for the first time, the convergence analysis of the underlying algorithm. This analysis provides the conditions of convergence as well as important insights for the application of PI^2 in high dimensional planning and control problems. In Section 4, we present learnable nonlinear systems which behave as point or limit cycle attractors. In Section 5, these attractors are used to parameterized either reference trajectories, for the case of planning, or control gains for the case of control. In the Section 6, we present applications of PI^2. More precisely in Section 6.1 PI^2 is used for the task of jumping over a gap with the 12DOF little dog robot while in 6.2 PI^2 is used for learning to open a door with a simulated CBi humanoid robot. Finally, towards learning locomotion, PI^2 is applied to learning rhythmic behaviors with nonlinear limit cycle attractor systems.

2. PATH INTEGRAL STOCHASTIC OPTIMAL CONTROL

The goal in stochastic optimal control is to control a stochastic dynamical system while minimizing a performance criterion. Therefore, in mathematical term a stochastic optimal control problem can be formulated as follows:

$$V(\mathbf{x}) = \min_{\mathbf{u}} J(\mathbf{x},\mathbf{u}) = \min_{\mathbf{u}} \int_{to}^{t_N} \boldsymbol{\mathcal{L}}(\mathbf{x},\mathbf{u},t)dt \qquad (1)$$

subject to the stochastic dynamical constrains:

$$\mathbf{dx} = (\mathbf{f}(\mathbf{x}_t) + \mathbf{G}(\mathbf{x})\mathbf{u})\,dt + \mathbf{B}(\mathbf{x})\mathbf{L}d\omega \qquad (2)$$

with $\mathbf{x}_t \in \Re^{n\times 1}$ denoting the state of the system, $\mathbf{G}_t = \mathbf{G}(\mathbf{x}_t) \in \Re^{n\times p}$ the control matrix, $\mathbf{B}_t = \mathbf{B}(\mathbf{x}_t) \in \Re^{n\times p}$ is the diffusions matrix $\mathbf{f}_t = \mathbf{f}(\mathbf{x}_t) \in \Re^{n\times 1}$ the passive dynamics, $\mathbf{u}_t \in \Re^{p\times 1}$ the control vector and $d\boldsymbol{\omega} \in \Re^{p\times 1}$ brownian noise. $\mathbf{L} \in\in \Re^{p\times p}$ is a state independent matrix with $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} = \mathbf{L}\mathbf{L}^T$. As immediate reward we consider

$$r_t = r(\mathbf{x}_t,\mathbf{u}_t,t) = q_t + \frac{1}{2}\mathbf{u}_t^T\mathbf{R}\mathbf{u}_t \qquad (3)$$

where $q_t = q(\mathbf{x}_t,t)$ is an arbitrary state-dependent cost function, and $\mathbf{R}$ is the positive definite weight matrix of the quadratic control cost. The stochastic HJB equation (Stengel, 1994; Fleming and Soner, 2006) associated with this stochastic optimal control problem is expressed as follows:

$$-\partial_t V_t = \min_{\mathbf{u}}\left(r_t + (\nabla_{\mathbf{x}}V_t)^T\mathbf{f}_t + \frac{1}{2}tr\left((\nabla_{\mathbf{xx}}V_t)\mathbf{G}_t\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\mathbf{G}_t^T\right)\right) \qquad (4)$$

To find the minimum, the reward function (3) is inserted into (4) and the gradient of the expression inside the parenthesis is taken with respect to controls $\mathbf{u}$ and set to zero. The corresponding optimal control is given by the equation:

$$\mathbf{u}(\mathbf{x}_t) = \mathbf{u}_t = -\mathbf{R}^{-1}\mathbf{G}(\mathbf{x})^T(\nabla_{\mathbf{x}}V(\mathbf{x},t)) \qquad (5)$$

Substitution of the optimal control into the stochastic HJB (4) results in the following nonlinear and second order PDE:

$$\begin{aligned}-\partial_t V_t = q_t + (\nabla_{\mathbf{x}}V_t)^T\mathbf{f}_t - \frac{1}{2}(\nabla_{\mathbf{x}}V_t)^T\mathbf{G}_t\mathbf{R}^{-1}\mathbf{G}_t^T(\nabla_{\mathbf{x}}V_t)\\ + \frac{1}{2}tr\left((\nabla_{\mathbf{xx}}V_t)\mathbf{B}_t\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\mathbf{B}_t^T\right)\end{aligned} \qquad (6)$$

To transform the PDE above into a linear one, we use a exponential transformation of the value function$V_t = -\lambda\log\Psi_t$. Given this logarithmic transformation, the partial derivatives of the value function with respect to time and state are expressed as follows: $\partial_t V_t = -\lambda\frac{1}{\Psi_t}\partial_t\Psi_t$, $\nabla_{\mathbf{x}}V_t = -\lambda\frac{1}{\Psi_t}\nabla_{\mathbf{x}}\Psi_t$ and $\nabla_{\mathbf{xx}}V_t = \lambda\frac{1}{\Psi_t^2}\nabla_{\mathbf{x}}\Psi_t\ \nabla_{\mathbf{x}}\Psi_t^T - \lambda\frac{1}{\Psi_t}\nabla_{\mathbf{xx}}\Psi_t$ By inserting the logarithmic transformation and the derivatives of the value function as well as considering the assumption $\lambda\mathbf{G}(\mathbf{x})\mathbf{R}^{-1}\mathbf{G}(\mathbf{x})^T = \mathbf{B}(\mathbf{x})\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}\mathbf{B}(\mathbf{x})^T = \boldsymbol{\Sigma}(\mathbf{x}_t) = \boldsymbol{\Sigma}_t$ the resulting PDE is formulated as follows:

$$-\partial_t\Psi_t = -\frac{1}{\lambda}q_t\Psi_t + \mathbf{f}_t^T(\nabla_{\mathbf{x}}\Psi_t) + \frac{1}{2}tr\left((\nabla_{\mathbf{xx}}\Psi_t)\boldsymbol{\Sigma}_t\right) \qquad (7)$$

with boundary condition: $\Psi_{t_N} = \exp\left(-\frac{1}{\lambda}\phi_{t_N}\right)$. The partial differential equation (PDE) in (7) corresponds to the so called Chapman Kolmogorov PDE, which is of second order and linear. Analytical solutions of even linear PDEs are plausible only in very special cases which correspond to systems with trivial low dimensional dynamics. In this work we compute the solution of the linear PDE above with the use of the Feynman - Kac lemma Oksendal (2003). The Feynman- Kac lemma provides a connection between stochastic differential equations and PDEs and therefore its use is twofold. On one side it can be used to find probabilistic solutions of PDEs based on forward sampling of diffusions while on the other side it can be used find solution of SDEs based on deterministic methods that numerically solve PDEs. The solution of the PDE above can be found by evaluating the expectation:

$$\Psi\left(\mathbf{x}_{t_i}\right) = E_{\boldsymbol{\tau}_i}\left(e^{-\int_{t_i}^{t_N}\frac{1}{\lambda}q(\mathbf{x})dt}\Psi(\mathbf{x}_{t_N})\right) \qquad (8)$$

on sample paths $\boldsymbol{\tau}_i = (\mathbf{x}_i,...,\mathbf{x}_{t_N})$ generated with the forward sampling of the diffusion equation $\mathbf{dx} = \mathbf{f}(\mathbf{x}_t)dt + \mathbf{B}(\mathbf{x})d\omega$. Under the use of the Feynman Kac lemma the stochastic optimal control problem has been transformed into an approximation problem of a path integral. With a view towards a discrete time approximation, which will be needed for numerical implementations, the solution (8) can be formulated as:

$$\Psi_{t_i} = \lim_{dt \to 0} \int p\left(\boldsymbol{\tau}_i | \mathbf{x}_i\right) \exp\left[-\frac{1}{\lambda}\left(\phi_{t_N} + \sum_{j=i}^{N-1} q_{t_j} dt\right)\right] d\boldsymbol{\tau}_i \tag{9}$$

where $\boldsymbol{\tau}_i = (\mathbf{x}_{t_i}, ..., \mathbf{x}_{t_N})$ is a sample path (or trajectory piece) starting at state $\mathbf{x}_{t_i}$ and the term $p\left(\boldsymbol{\tau}_i | \mathbf{x}_i\right)$ is the probability of sample path $\boldsymbol{\tau}_i$ conditioned on the start state $\mathbf{x}_{t_i}$. Since equation (9) provides the exponential cost to go Ψ_{t_i} in state $\mathbf{x}_{t_i}$, the integration above is taken with respect to sample paths $\boldsymbol{\tau}_i = \left(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i+1}}, ..., \mathbf{x}_{t_N}\right)$. The differential term $d\boldsymbol{\tau}_i$ is defined as $d\boldsymbol{\tau}_i = (d\mathbf{x}_{t_i}, ..., d\mathbf{x}_{t_N})$. After the exponentiated value function $\Psi(\mathbf{x}, t)$ has been approximated, the optimal control are found according to the equation that follows:

$$\mathbf{u} = \lambda \mathbf{R}^{-1} \mathbf{G}(\mathbf{x})^T \frac{\nabla_{\mathbf{x}} \Psi(\mathbf{x}, t)}{\Psi(\mathbf{x}, t)} \tag{10}$$

Clearly optimal controls in the equation above act such that the stochastic dynamical system visits regions of the state space with high exponentiated values function $\Psi(\mathbf{x}, t)$ while in the optimal control formulation (5) controls will move the system towards part of the state space with minimum cost-to-go $V(\mathbf{x}, t)$. This observation is in complete agreement with the exponentiation of value function $\Psi(\mathbf{x}, t) = \exp\left(-\frac{1}{\lambda} V(\mathbf{x}, t)\right)$. Essentially, the resulting value function $\Psi(\mathbf{x}, t)$ can be thought as a probability of the state and thus states with high cost to go $V(\mathbf{x}, t)$ will be less probable(= small $\Psi(\mathbf{x}, t)$) while state with small cost to go will be most probable. In that sense the stochastic optimal control has been transformed from a minimization to maximization optimization problem. Finally the intuition behind the condition $\lambda \mathbf{G}(\mathbf{x}) \mathbf{R}^{-1} \mathbf{G}(\mathbf{x})^T = \mathbf{B}(\mathbf{x}) \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} \mathbf{B}(\mathbf{x})^T$ is that, since the weight control matrix $\mathbf{R}$ is inverse proportional to the variance of the noise, a high variance control input implies cheap control cost, while small variance control inputs have high control cost. From a control theoretic stand point such a relationship makes sense due to the fact that under a large disturbance (= high variance) significant control authority is required to bring the system back to a desirable state. This control authority can be achieved with corresponding low control cost in $\mathbf{R}$.

With the goal to find the $\Psi(\mathbf{x}, t)$ in equation (9), in the next section we derive the distribution $p\left(\boldsymbol{\tau}_i | \mathbf{x}_i\right)$ based on the passive dynamics. This is a generalization of results in Kappen (2007); Broek et al. (2008) for more details see Theodorou et al. (2010a), Theodorou (2011) .

2.1 Generalized Path Integral Formulation

In many stochastic dynamical systems, the diffusion transition matrix $\mathbf{B}_t$ is state depended and its structure depends on the partition of the state in directly and non-directly actuated parts. Since only some of the states are directly actuated, the state vector is partitioned into $\mathbf{x} = [\mathbf{x}^{(m)^T} \quad \mathbf{x}^{(c)^T}]^T$ with $\mathbf{x}^{(m)} \in \Re^{k \times 1}$ the non-directly actuated part and $\mathbf{x}^{(c)} \in \Re^{l \times 1}$the directly actuated part. Subsequently, the passive dynamics term and the diffusion transition matrix can be partitioned as $\mathbf{f}_t = [\mathbf{f}_t^{(m)^T} \quad \mathbf{f}_t^{(c)^T}]^T$ with $\mathbf{f}_m \in \Re^{k \times 1}$, $\mathbf{f}_c \in \Re^{l \times 1}$ and $\mathbf{B}_t = [\mathbf{0}_{k \times p} \quad \mathbf{B}_t^{(c)^T}]^T$ with $\mathbf{B}_t^{(c)} \in \Re^{l \times p}$. The discretized state space representation of such systems is given as:

$$\begin{pmatrix} \mathbf{x}_{t_{i+1}}^{(m)} \\ \mathbf{x}_{t_{i+1}}^{(c)} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{t_i}^{(m)} \\ \mathbf{x}_{t_i}^{(c)} \end{pmatrix} + \begin{pmatrix} \mathbf{f}_{t_i}^{(m)} \\ \mathbf{f}_{t_i}^{(c)} \end{pmatrix} dt + \begin{pmatrix} \mathbf{0}_{k \times p} \\ \mathbf{B}_{t_i}^{(c)} \end{pmatrix} \sqrt{dt} d\boldsymbol{\omega} \tag{11}$$

As it has been shown in Theodorou et al. (2010a), Theodorou (2011) with the formulation above, (9) is expressed as:

$$\Psi_{t_i} = \lim_{dt \to 0} \int \frac{1}{D(\boldsymbol{\tau}_i)} \exp\left(-\frac{1}{\lambda} S(\boldsymbol{\tau}_i)\right) d\boldsymbol{\tau}_i^{(c)} \tag{12}$$

with:

$$S(\boldsymbol{\tau}_i) = \phi_{t_N} + \sum_{j=i}^{N-1} \left(q_{t_j} + \left\| \frac{\mathbf{x}_{t_{j+1}}^{(c)} - \mathbf{x}_{t_j}^{(c)}}{dt} - \mathbf{f}_{t_j}^{(c)} \right\|_{\mathbf{H}_{t_j}^{-1}}^2 \right) dt$$

and $D(\boldsymbol{\tau}_i) = \Pi_{j=i}^{N-1} \left((2\pi)^{l/2} |\boldsymbol{\Sigma}_{t_j}|^{1/2}\right)$. Note that the integration is over $d\boldsymbol{\tau}_i^{(c)} = \left(d\mathbf{x}_{t_i}^{(c)}, ..., d\mathbf{x}_{t_N}^{(c)}\right)$. We can have a more compact formulation of equation (12) formulated as:

$$\Psi_{t_i} = \lim_{dt \to 0} \int \exp\left(-\frac{1}{\lambda} Z(\boldsymbol{\tau}_i)\right) d\boldsymbol{\tau}_i^{(c)} \tag{13}$$

where $Z(\boldsymbol{\tau}_i) = S(\boldsymbol{\tau}_i) + \lambda \log D(\boldsymbol{\tau}_i)$. It can be shown that $Z(\boldsymbol{\tau}_i) = \tilde{S}(\boldsymbol{\tau}_i) + \frac{\lambda(N-i)l}{2} \log\left(2\pi dt \lambda\right)$ where $\tilde{S}(\boldsymbol{\tau}_i) = S(\boldsymbol{\tau}_i) + \frac{\lambda}{2} \sum_{j=i}^{N-1} \log |\boldsymbol{\mathcal{B}}_{t_j}|$ and $\boldsymbol{\mathcal{B}} = \mathbf{B}(\mathbf{x}) \mathbf{B}(\mathbf{x})^T$. This formula is a necessary step for the derivation of optimal controls in the next section. As it is shown in Theodorou et al. (2010a); Theodorou (2011) the constant term $\frac{\lambda N l}{2} \log\left(2\pi dt \lambda\right)$ drops from our calculations.

2.2 Optimal Controls

For every moment of time, the optimal controls are given as $\mathbf{u}(\mathbf{x}_{t_i}) = -\mathbf{R}^{-1} \mathbf{G}_{t_i}^T (\nabla_{x_{t_i}} V_{t_i})$. Due to the exponential transformation of the value function, the equation of the optimal controls can be written as

$$\mathbf{u}(\mathbf{x}_{t_i}) = \lambda \mathbf{R}^{-1} \mathbf{G}_{t_i} \frac{\nabla_{\mathbf{x}_{t_i}} \Psi_{t_i}}{\Psi_{t_i}} \tag{14}$$

After substituting Ψ_{t_i} with (13) and canceling the state independent terms of the cost we have:

$$\mathbf{u}(\mathbf{x}_{t_i}) = \lim_{dt \to 0} \left(\lambda \mathbf{R}^{-1} \mathbf{G}_{t_i}^T \frac{\nabla_{\mathbf{x}_{t_i}^{(c)}} \left(\int e^{-\frac{1}{\lambda} \tilde{S}(\boldsymbol{\tau}_i)} d\boldsymbol{\tau}_i^{(c)} \right)}{\int e^{-\frac{1}{\lambda} \tilde{S}(\boldsymbol{\tau}_i)} d\boldsymbol{\tau}_i^{(c)}} \right) \tag{15}$$

It has been shown in Theodorou et al. (2010a), Theodorou (2011) the optimal controls are expressed as

$$\mathbf{u}(\mathbf{x}_{t_i}) = \lim_{dt \to 0} \int P\left(\boldsymbol{\tau}_i\right) \ \mathbf{u}_L\left(\boldsymbol{\tau}_i\right) \ d\boldsymbol{\tau}_i^{(c)} \tag{16}$$

with the probability $P\left(\boldsymbol{\tau}_i\right)$ and local controls $\mathbf{u}_L\left(\boldsymbol{\tau}_i\right)$ defined as

$$P\left(\boldsymbol{\tau}_i\right) = \frac{e^{-\frac{1}{\lambda} \tilde{S}(\boldsymbol{\tau}_i)}}{\int e^{-\frac{1}{\lambda} \tilde{S}(\boldsymbol{\tau}_i)} d\boldsymbol{\tau}_i} \tag{17}$$

and the local control are expressed as:

$$\mathbf{u}_L(\boldsymbol{\tau}_i) dt = \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \left(\mathbf{G}_{t_i}^{(c)} \mathbf{R}^{-1} \mathbf{G}_{t_i}^{(c)T} \right)^{-1} \mathbf{G}_{t_i}^{(c)} d\boldsymbol{\omega}_{t_i} \tag{18}$$

The optimal control are computed with the evaluation of (16) and (18),(17) on the sampled trajectories.

3. ITERATIVE PATH INTEGRAL STOCHASTIC OPTIMAL CONTROL

In this section, we show how Path Integral Control is transformed into an iterative process, which has several advantages for use on a real robot. In particular, the expectation (8) in the Feynman Kac Lemma is evaluated over the trajectories $\boldsymbol{\tau}_i = \left(\mathbf{x}_{t_i}, \mathbf{x}_{t_{i+1}}, ..., \mathbf{x}_{t_N}\right)$ sampled with the forward propagation of uncontrolled diffusion $\mathbf{dx} = \mathbf{f}(\mathbf{x}_t)dt + \mathbf{B}(\mathbf{x})d\omega$. This sampling approach is inefficient since it is very likely that parts of the state space relevant to the optimal control task may not be reached by the sampled trajectories at once. In addition, it has poor scalability properties when applied to high dimensional robotic optimal control problems. Besides the reason of poor sampling, it is very common in robotics applications to have an initial controller-policy which is manually tuned and found based on experience. In such cases, the goal is to improve this initial policy by performing an iterative process. At every iteration (i) the policy $\delta\mathbf{u}^{(i-1)}$ is applied to the dynamical system to generate state space trajectories which are going to be used for improving the current policy. The policy improvement results from the evaluation of the expectation (9) of the Feynman - Kac Lemma on the sampled trajectories and the use of the path integral control formalism to find $\delta\mathbf{u}^{(i)}$. The old policy $\delta\mathbf{u}^{(i-1)}$ is updated according to $\delta\mathbf{u}^{(i-1)} + \delta\mathbf{u}^{(i)}$ and the process repeats again with the generation of the new state space trajectories according to the updated policy. In mathematical terms the iterative version of Path Integral Control is expressed as follows:

$$\begin{aligned} V^{(i)}(\mathbf{x}) &= \min_{\delta\mathbf{u}^{(i)}} J(\mathbf{x},\mathbf{u}) \\ &= \min_{\delta\mathbf{u}^{(i)}} E\left(\int_{to}^{t_N} \left(q(\mathbf{x},t) + \delta\mathbf{u}^{(i)T}\ \mathbf{R}\ \delta\mathbf{u}^{(i)}\right) dt\right) \end{aligned} \tag{19}$$

subject to the stochastic dynamical constrains:

$$\mathbf{dx} = \left(\mathbf{f}^{(i)}(\mathbf{x}_t) + \mathbf{G}(\mathbf{x})\delta\mathbf{u}^{(i)}\right) dt + \mathbf{B}(\mathbf{x})\mathbf{L}d\omega \tag{20}$$

where $\mathbf{f}^{(i)}(\mathbf{x}_t) = \mathbf{f}^{(i-1)}(\mathbf{x}_t) + \mathbf{G}(\mathbf{x})\delta\mathbf{u}^{(i-1)}$ where $\delta\mathbf{u}^{(i-1)}$ is the control correction found in the previous iteration. The linear HJB equation is now formulated as:

$$-\partial_t\Psi_t^{(i)} = -\frac{1}{\lambda}q_t\Psi_t^{(i)} + \mathbf{f}_t^{(i)T}(\nabla_{\mathbf{x}}\Psi_t^{(i)}) + \frac{1}{2}tr\left((\nabla_{\mathbf{xx}}\Psi_t^{(i)})\boldsymbol{\Sigma}\right) \tag{21}$$

The solution of PDE above is given by: $\Psi^{(i)}(\mathbf{x}_t) = E_{\boldsymbol{\tau}^{(i)}}\left(e^{-\int_{t_i}^{t_N}\frac{1}{\lambda}q(\mathbf{x})dt}\Psi(\mathbf{x}_{t_N})\right)$, where the state trajectories $\boldsymbol{\tau}^{(i)}$ are sampled with the diffusion: $\mathbf{dx} = \mathbf{f}^{(i)}(\mathbf{x}_t)dt + \mathbf{B}(\mathbf{x})d\omega$. The optimal control at iteration (i) is expressed as:

$$\delta\mathbf{u}^{(i)} = \lambda\mathbf{R}^{-1}\mathbf{G}(\mathbf{x})^T\frac{\nabla_{\mathbf{x}}\Psi^{(i)}(\mathbf{x},t)}{\Psi^{(i)}(\mathbf{x},t)} \tag{22}$$

and it is applied to the dynamics $\mathbf{f}^{(i)}(\mathbf{x}_t)$. The application of the new control results in updating the previous control $\delta\mathbf{u}^{(i-1)}$ and creating the new dynamics $\mathbf{f}^{(i+1)}(\mathbf{x}_t) = \mathbf{f}^{(i)}(\mathbf{x}_t) + \mathbf{G}(\mathbf{x})\delta\mathbf{u}^{(i)} = \mathbf{f}^{(i-1)}(\mathbf{x}_t) + \mathbf{G}(\mathbf{x})\left(\delta\mathbf{u}^{(i)} + \delta\mathbf{u}^{(i-1)}\right)$. At the next iteration $(i+1)$ of the iterative path integral control, the corresponding exponentiated value function $\Psi^{(i+1)}$ is given by the following PDE:

$$\begin{aligned} -\partial_t\Psi_t^{(i+1)} = &-\frac{1}{\lambda}q_t\Psi_t^{(i+1)} + \mathbf{f}_t^{(i+1)T}(\nabla_{\mathbf{x}}\Psi_t^{(i+1)}) \\ &+ \frac{1}{2}tr\left((\nabla_{\mathbf{xx}}\Psi_t^{(i+1)})\boldsymbol{\Sigma}\right) \end{aligned} \tag{23}$$

The solution of the PDE is now expressed as : $\Psi^{(i+1)}(\mathbf{x}_t) = E_{\boldsymbol{\tau}^{(i+1)}}\left(e^{-\int_{t_i}^{t_N}\frac{1}{\lambda}q(\mathbf{x})dt}\Psi(\mathbf{x}_{t_N})\right)$ where the state trajectories $\boldsymbol{\tau}^{(i+1)}$ are sampled with the diffusion: $\mathbf{dx} = \mathbf{f}^{(i+1)}(\mathbf{x}_t)dt + \mathbf{B}(\mathbf{x})d\omega$.

Our ultimate goal for the iterative path integral control is the sufficient conditions so that at every iteration the value function improves $V^{(i+1)}(\mathbf{x},t) < V^{(i)}(\mathbf{x},t) < .... < V^{(0)}(\mathbf{x},t)$. Since in the path integral control formalism we make use of the transformation $\Psi(\mathbf{x},t) = \exp\left(-\frac{1}{\lambda}V(\mathbf{x},t)\right)$ it suffices to show that $\Psi^{(i+1)}(\mathbf{x},t) > \Psi^{(i)}(\mathbf{x},t) > .... > \Psi^{(0)}(\mathbf{x},t)$. If the last condition is true then at every (i) iteration the stochastic dynamical system visits to regions of state space with more and more probable states(= states with high $\Psi(\mathbf{x},t)$). These states correspond to small value function $V(\mathbf{x},t)$. To find the condition under which the above is true, we proceed with the analysis that follows. Since we know that $\mathbf{f}^{(i+1)}(\mathbf{x}_t) = \mathbf{f}^{(i)}(\mathbf{x}_t) + \mathbf{G}(\mathbf{x})\delta\mathbf{u}^{(i)}$ we substitute in (23) and we will have that:

$$\begin{aligned} -\partial_t\Psi_t^{(i+1)} = &-\frac{1}{\lambda}q_t\Psi_t^{(i+1)} + \mathbf{f}_t^{(i)T}(\nabla_{\mathbf{x}}\Psi_t^{(i+1)}) \\ &+ \frac{1}{2}tr\left((\nabla_{\mathbf{xx}}\Psi_t^{(i+1)})\boldsymbol{\Sigma}\right) \\ &+ \delta\mathbf{u}^{(i)T}\mathbf{G}^T(\nabla_{\mathbf{x}}\Psi_t^{(i+1)}) \end{aligned} \tag{24}$$

substitution of $\delta\mathbf{u}$ results in:

$$\begin{aligned} -\partial_t\Psi_t^{(i+1)} = &-\frac{1}{\lambda}q_t\Psi_t^{(i+1)} + \mathbf{f}_t^{(i)T}(\nabla_{\mathbf{x}}\Psi_t^{(i+1)}) \\ &+ \frac{1}{2}tr\left((\nabla_{\mathbf{xx}}\Psi_t^{(i+1)})\boldsymbol{\Sigma}\right) + \boldsymbol{\mathcal{F}}(\mathbf{x},t) \end{aligned}$$

where

$$\boldsymbol{\mathcal{F}}(\mathbf{x},t) = \frac{\lambda}{\Psi^{(i)}(\mathbf{x},t)}\nabla_{\mathbf{x}}\Psi^{(i)}(\mathbf{x},t)^T\ \mathbf{GRG}^T\ \nabla_{\mathbf{x}}\Psi^{(i+1)}(\mathbf{x},t) \tag{25}$$

correspond to a force term which is the inner product of the gradients of the value functions at iterations (i) and $(i+1)$ under the metric $\boldsymbol{\mathcal{M}} = \frac{\lambda}{\Psi^{(i)}(\mathbf{x},t)}\mathbf{GRG}^T$. Clearly $\boldsymbol{\mathcal{M}} > 0$ since the matrix product $\mathbf{GRG}^T > 0$ is positive definite and $\lambda > 0, \Psi(\mathbf{x},t) > 0$. Comparing the two PDEs at iteration (i) and $(i+1)$ and by using the linear operator $\boldsymbol{\mathcal{L}}^{(i)} = -\frac{1}{\lambda}q_t + \mathbf{f}_t^{(i)T}\nabla_{\mathbf{x}} + \frac{1}{2}tr(\boldsymbol{\Sigma}\nabla_{\mathbf{xx}})$ we have:

$$-\partial_t\Psi_t^{(i+1)} = \boldsymbol{\mathcal{L}}^{(i)}\Psi_t^{(i+1)} + \boldsymbol{\mathcal{F}}(\mathbf{x},t) \tag{26}$$

$$-\partial_t\Psi_t^{(i)} = \boldsymbol{\mathcal{L}}^{(i)}\Psi_t^{(i)} \tag{27}$$

under the same terminal condition $\Psi_{t_N}^{(i)} = \Psi_{t_N}^{(i+1)} = \exp\left(-\frac{1}{\lambda}\phi(\mathbf{x}_{t_N})\right)$. We claim that $\Psi^{(i+1)} < \Psi^{(i)}$ if $\boldsymbol{\mathcal{F}}(\mathbf{x},t) >$

$0\ \forall \mathbf{x}, t$. To see this result we rewrite equation (24) in the following form:

$$\begin{aligned}-\partial_t \Psi_t^{(i+1)} =& -\frac{1}{\lambda} q_t \Psi_t^{(i+1)} + \mathbf{f}_t^{(i)T} (\nabla_{\mathbf{x}} \Psi_t^{(i+1)}) \\ &+ \frac{1}{2} tr\left((\nabla_{\mathbf{xx}} \Psi_t^{(i+1)}) \mathbf{\Sigma} \right) \\ &+ \frac{1}{\lambda} \delta \mathbf{u}^{(i)T}\ \mathbf{R}\ \delta \mathbf{u}^{(i+1)T} \Psi_t^{(i+1)} \end{aligned} \tag{28}$$

or in a more compact form:

$$\begin{aligned}-\partial_t \Psi_t^{(i+1)} =& -\frac{1}{\lambda} \tilde{q}_t \Psi_t^{(i+1)} + \mathbf{f}_t^{(i)T} (\nabla_{\mathbf{x}} \Psi_t^{(i+1)}) \\ &+ \frac{1}{2} tr\left((\nabla_{\mathbf{xx}} \Psi_t^{(i+1)}) \mathbf{\Sigma} \right) \end{aligned} \tag{29}$$

where the term $\tilde{q} = \tilde{q}(\mathbf{x}, t, \delta \mathbf{u}^{(i)}, \delta \mathbf{u}^{(i+1)})$ is defined as $\tilde{q} = q(\mathbf{x}, t) - \frac{1}{\lambda} \delta \mathbf{u}^{(i)T}\ \mathbf{R}\ \delta \mathbf{u}^{(i+1)T}$. Clearly there are 3 cases depending on the sign of $F(\mathbf{x}, t)$ and therefore the sign of $\frac{1}{\lambda} \delta \mathbf{u}^{(i)T}\ \mathbf{R}\ \delta \mathbf{u}^{(i+1)T}$. More precisely we will have that

- If $\boldsymbol{\mathcal{F}}(\mathbf{x}, t) > 0 \Rightarrow \delta \mathbf{u}^{(i)T}\ \mathbf{R}\ \delta \mathbf{u}^{(i+1)T} > 0$. By comparing (21) with (29) we see that state cost $\tilde{q}$ subtracted from $\Psi^{(i+1)}$ is smaller than the state cost q subtracted from $\Psi^{(i)}$ and therefore $\Psi^{(i+1)}(\mathbf{x}, t) > \Psi^{(i)}(\mathbf{x}, t)$.
- If $\boldsymbol{\mathcal{F}}(\mathbf{x}, t) = 0 \Rightarrow \delta \mathbf{u}^{(i)T}\ \mathbf{R}\ \delta \mathbf{u}^{(i+1)T} = 0$ the two PDEs (21) and (29) are identical. Therefore under the same boundary condition $\Psi^{(i+1)}(\mathbf{x}, t_N) = \Psi^{(i)}(\mathbf{x}, t_N)$ we will have that $\Psi^{(i+1)}(\mathbf{x}, t) = \Psi^{(i)}(\mathbf{x}, t)$.
- If $\boldsymbol{\mathcal{F}}(\mathbf{x}, t) < 0 \Rightarrow \delta \mathbf{u}^{(i)T}\ \mathbf{R}\ \delta \mathbf{u}^{(i+1)T} < 0$. By comparing (21) with (29) we see that the state cost $\tilde{q}$ subtracted from $\Psi^{(i+1)}$ is bigger than the state cost q subtracted from $\Psi^{(i)}$ and therefore $\Psi^{(i+1)}(\mathbf{x}, t) < \Psi^{(i)}(\mathbf{x}, t)$.

4. LEARNABLE NONLINEAR ATTRACTOR SYSTEMS

The basic idea of our approach is to use an analytically well understood dynamical system with convenient stability properties and modulate it with nonlinear terms such that it achieves a desired attractor behavior. In the two subsections that follows we present the nonlinear point attractors and the nonlinear limit cycle attractors.

4.1 Nonlinear Point Attractors with adjustable attractor Land-scape

The nonlinear point attractor consists of two sets of differential equations, the canonical and transformation system which are coupled through a nonlinearity (Ijspeert et al., 2003). The canonical system is formulated as $\frac{1}{\tau} \dot{x}_t = -\alpha x_t$. That is a first - order linear dynamical system for which, starting from some arbitrarily chosen initial state x_0 , e.g., $x_0 = 1$, the state x converges monotonically to zero. x can be conceived of as a phase variable, where $x = 1$ would indicate the start of the time evolution, and x close to zero means that the goal g (see below) has essentially been achieved. The transformation system consist of the following two differential equations:

$$\begin{aligned}\tau \dot{z} =& \alpha_z \beta_z \left(\left(g + \frac{f}{\alpha_z \beta_z} \right) - y \right) - \alpha_z z \\ \tau \dot{y} =& z \end{aligned} \tag{30}$$

Essentially, these 3 differential equations code a learnable point attractor for a movement from y_{t_0} to the goal g, where $\boldsymbol{\theta}$ determines the shape of the attractor. $y_t, \dot{y}_t$ denote the position and velocity of the trajectory, while z_t, x_t are internal states. α_z, β_z, τ are time constants. The nonlinear coupling or forcing term f is defined as: $f(x) = \frac{\sum_{i=1}^{N} K(x_t, c_i) \theta_i x_t}{\sum_{i=1}^{N} K(x_t, c_i)} (g - y_0) = \mathbf{\Phi}_P(x)^T \boldsymbol{\theta}$. The basis functions $K(x_t, c_i)$ are defined as $K(x_t, c_i) = \exp\left(-0.5 h_j (x_t - c_j)^2\right)$ with bandwith h_j and center c_j of the Gaussian kernels – for more details see (Ijspeert et al., 2003). The full dynamics or the rhythmic movement primitives have the form of $d\mathbf{x} = F(\mathbf{x})dt + \mathbf{G}(\mathbf{x})\mathbf{u}dt$ where the state $\mathbf{x}$ is specified as $\mathbf{x} = (x, y, z)$ while the controls are specified as $\mathbf{u} = \boldsymbol{\theta} = (\theta_1, ..., \theta_p)^T$.The representation above is advantageous as it guarantees attractor properties towards the goal while remaining linear in the parameters $\boldsymbol{\theta}$ of the function approximator. By varying the parameter $\boldsymbol{\theta}$ the shape of the trajectory changes while the goal state g and initial state y_{t_0} remain fixed. These properties facilitate learning (Peters and Schaal, 2008).

4.2 Nonlinear Limit Cycle Attractors with adjustable attractor Land-scape

The canonical system for the case of limit cycle attractors consist the differential equation $\tau \dot{\phi} = 1$ where the term $\phi \in [0, 2\pi]$ correspond to the phase angle of the oscillator in polar coordinates. The amplitude of the oscillation is assumed to be r. This oscillator produces a stable limit cycle when projected into Cartesian coordinated with $v_1 = r\cos(\phi)$ and $v_2 = r\sin(\phi)$. In fact, it corresponds to form of the (Hopf-like) oscillator equations

$$\tau \dot{v_1} = -\mu \frac{\sqrt{v_1^2 + v_2^2} - r}{\sqrt{v_1^2 + v_2^2}} v_1 - v_2 \tag{31}$$

$$\tau \dot{v_2} = -\mu \frac{\sqrt{v_1^2 + v_2^2} - r}{\sqrt{v_1^2 + v_2^2}} v_2 + v_1 \tag{32}$$

where μ is a positive time constant. The system above evolve to the limit cycle $v_1 = r\cos(t/\tau + c)$ and $v_2 = r\sin(t/\tau + c)$ with c a constant, given any initial conditions except $[v_1, v_2] = [0, 0]$ which is an unstable fixed point. Therefore the canonical system provides the amplitude signal (r) and a phase signal (ϕ) to the forcing term $f(\phi, r) = \frac{\sum_{i=1}^{N} K(\phi, c_i) \theta_i}{\sum_{i=1}^{N} K(\phi, c_i)} r = \mathbf{\Phi}_R(\phi)^T \boldsymbol{\theta}$, where the basis function $K(\phi, c_i)$ are defined as $K(\phi, c_i) = \exp\left(h_i \left(\cos(\phi - c_i) - 1\right)\right)$. The forcing term is incorporated into the transformation system which is expressed by the equations (30). The full dynamics of the rhythmic movement primitives have the form of $d\mathbf{x} = F(\mathbf{x})dt + \mathbf{G}(\mathbf{x})\mathbf{u}dt$ where the state $\mathbf{x}$ is specified as $\mathbf{x} = (\phi, v_1, v_2, z, y)$ while the controls are specified as $\mathbf{u} = \boldsymbol{\theta} = (\theta_1, ..., \theta_p)^T$. The term g for the case of limit cycle attractors is interpreted as an-anchor point (or set point) for the oscillatory trajectory, which can be changed to accommodate any desired

baseline of the oscillation The complexity of attractors is restricted only by the abilities of the function approximator used to generate the forcing term, which essentially allows for almost arbitrarily complex (smooth) attractors with modern function approximators

5. PI^2 FOR SIMULTANEOUS ROBOT CONTROL AND PLANNING

In this section we show how the Path integral optimal control formalism in combination with the point and limit cycle attractors can be used for optimal planning Theodorou et al. (2010a) and gain scheduling Theodorou (2011); Buchli et al. (2010) of robotic systems in high dimensions. As an example, consider a robotic system with rigid body dynamics (RBD) equations (Sciavicco and Siciliano, 2000) using a parameterized policy:

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}\left(-\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{v}(\mathbf{q})\right) + \mathbf{M}(\mathbf{q})^{-1}\mathbf{u} \quad (33)$$

$$\mathbf{u} = \mathbf{K}_P(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_D(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) \quad (34)$$

where $\mathbf{M}$ is the RBD inertia matrix, $\mathbf{C}$ are Coriolis and centripetal forces, and $\mathbf{v}$ denotes gravity forces. The state of the robot is described by the joint angles $\mathbf{q}$ and joint velocities $\dot{\mathbf{q}}$. The proportional-Derivative (PD) controller with positive definite gain matrices $\mathbf{K}_P$ and $\mathbf{K}_D$ have the form $\mathbf{K}_P = \mathbf{diag}\left(K_p^{(1)}, K_p^{(2)}, ..., K_p^{(N)}\right)$ and $\mathbf{K}_D = \mathbf{diag}\left(K_d^{(1)}, K_d^{(2)}, ..., K_d^{(N)}\right)$ where $K_p^{(i)}, K_d^{(i)}$ are the proportional and derivative gains for every DOF i. These gains converts a desired trajectory $\mathbf{q}_d, \dot{\mathbf{q}}_d$ into a motor command $\mathbf{u}$. The gains are parameterized as follows:

$$dK_p^{(i)} = \alpha_K\left(\mathbf{\Phi}_P^{(i)T}\left(\boldsymbol{\theta}^{(i)}dt + d\boldsymbol{\omega}^{(i)}\right) - K_p^{(i)}dt\right) \quad (35)$$

This equation models the time course of the position gains which are are represented by a basis function $\mathbf{\Phi}_P^{(i)T}\boldsymbol{\theta}^{(i)}$ linear with respect to the learning parameter $\boldsymbol{\theta}^{(i)}$, and these parameter can be learned with the (PI^2). We will assume that the time constant α_K is so large, that for all practical purposes we can assume that $K_P^{(i)} = \mathbf{\Phi}_p^{(i)T}\left(\boldsymbol{\theta}^{(i)} + \boldsymbol{\epsilon}_t^{(i)}\right)$ holds at all time where $\boldsymbol{\epsilon}_t^{(i)} = \frac{d\boldsymbol{\omega}^{(i)}}{dt}$. In our experiments $\mathbf{K}_D$ gains are specified as $K_d^{(i)} = \xi\sqrt{K_p^{(i)}}$ where ξ is user determined. Alternatively, for the case of optimal planing we could create another form of control structure in which we add for the RBD system (33) the following equation:

$$\ddot{\mathbf{q}}_d = \mathbf{G}(\mathbf{q}_d, \dot{\mathbf{q}}_d)(\boldsymbol{\theta} + \boldsymbol{\epsilon}_t) \quad (36)$$

where we represent the desired trajectory with point or limit cycle attractor. The control or learning parameter for this case is the parameter $\boldsymbol{\theta}$ in (36).

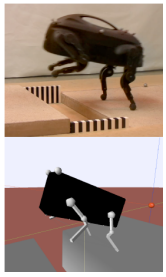
6. APPLICATIONS

6.1 Task 1: Learning to jump Little dog Robot

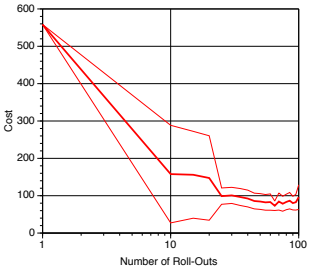
Goal: The task for the robot dog is to jump across as gap. The jump should make forward progress as much as possible, as it is a maneuver in a legged locomotion competition which scores the speed of the robot. The robot has three DOFs per leg, and thus a total of $d = 12$ DOFs. Each DOF was represented as a DMP with 50 basis functions.

Initial Trajectories: An initial seed behavior was taught by learning from demonstration, which allowed the robot to barely reach the other side of the gap without falling into the gap – the demonstration was generated from a manual adjustment of spline nodes in a spline-based trajectory plan for each leg.

Cost function: Iterative path integral control learning used primarily the forward progress as a reward, and slightly penalized the squared acceleration of each DOF, and the length of the parameter vector. Additionally, a penalty was incurred if the yaw or the roll exceeded a threshold value – these penalties encouraged the robot to jump straight forward and not to the side, and not to fall over. The exact cost function was:



(a) Real & Simulated Robot Dog



(b) Learning curve $\mu \pm 1\sigma$ for Dog Jump with

Fig. 1. Reinforcement learning of optimizing to jump over a gap with a robot dog.

$$r_t = r_{roll} + r_{yaw} + \sum_{i=1}^{d}\left(a_1 f_{i,t}^2 + 0.5a_2\ \boldsymbol{\theta}_i^T\boldsymbol{\theta}\right) \quad (37)$$

where $r_{roll} = 100 * (|roll_t| - 0.3)^2$ if $|roll_t| > 0.3$ and $r_{roll} = 0$ otherwise. Similarly $r_{yaw} = 100 * (|yaw_t| - 0.1)^2$ if $|yaw_t| > 0.1$ and $r_{roll} = 0$ otherwise .The terminal cost $\phi_{t_N} = 50000(goal - x_{nose})^2$ with x_{nose} the position of the front tip (the "nose") of the robot in the forward direction, which is the direction towards the *goal*.

PI^2 parameters: The parameters a_1 and a_2 are tuned as $(a_1 = 1.e-6, a_2 = 1.e-8)$ The multipliers for each reward component were tuned to have a balanced influence of all terms. Ten learning trials were performed initially for the first parameter update. The best 5 trials were kept, and five additional new trials were performed for the second and all subsequent updates. Essentially, this method performs importance sampling, as the rewards for the 5 trials in memory were re-computed with the latest parameter vectors. A total of 100 trials was performed per run, and ten runs were collected for computing mean and standard deviations of learning curves. Learning was performed on a physical simulator of the robot dog, as the real robot dog was not available for this experiment.

Results: Figure 1 illustrates that after about 30 trials (i.e., 5 updates), the performance of the robot was significantly improved, such that after the jump, almost the

entire body was lying on the other side of the gap. It should be noted that applying iterative path integral control was algorithmically very simple, and manual tuning only focused on generated a good cost function, which is a different research topic beyond the scope of this paper.

6.2 Task 2: Pushing open a door

Goal. In this task, the simulated CBi humanoid robot Cheng et al. (2007) is required to open a door. This robot is accurately simulated with the SL software Schaal (2009). For this task, we not only learn the gain schedules, but also improve the planned joint trajectories with PI^2 simultaneously.

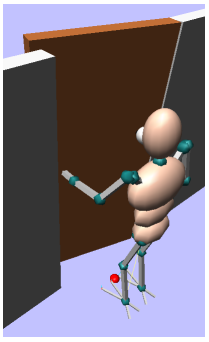
Initial trajectory. In this task, we fix the base of the robot, and consider only the 7 degrees of freedom in the left arm. The initial trajectory before learning is a minimum jerk trajectory in joint space. In the initial state, the upper arm is kept parallel to the body, and the lower arm is pointing forward. The target state is depicted in Fig. 2. With this task, we demonstrate that our approach can not only be applied to imitation of observed behavior, but also to manually specify trajectories, which are fine-tuned along with the gain schedules.

Initial gains. The gains of the 7 joints are initialized to $1/10th$ of their default values. This leads to extremely compliant behavior, whereby the robot is not able to exert enough force to overcome the static friction of the door, and thus cannot move it. The minimum gain for all joints was set to 5. Optimizing both joint trajectories and gains leads to a 14-dimensional learning problem.

Cost function. The terminal cost is the degree to which the door was opened, i.e. $\phi_{t_N} = 10^4 \cdot (\psi_{max} - \psi_N)$, where the maximum door opening angle ψ_{max} is $0.3rad$ (it is out of reach otherwise). The immediate cost for the gains is again $q_t = \frac{1}{N} \sum_{i=1}^{3} K_P^i$.

PI^2 parameters. The variance of the exploration noise for the gains is again $10^{-4}\gamma^n$, and for the joint trajectories $10\gamma^n$, both with decay parameter $\lambda = 0.99$ and n the number of updates [1].

Results. Fig. 2 (right) depicts the total cost of the noiseless test trial after each update. The costs for the gains are plotted separately. When all of the costs are due to gains, i.e. the door is opened completely to ψ_{max} and the task is achieved, the graphs of the total cost and that of the gains coincide. The joint trajectories and gain schedules after 0, 6 and 100 updates are depicted in Fig. 3.



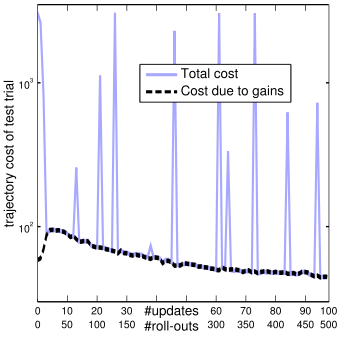


Fig. 2. Left: Task scenario. Right: Learning curve for the door task. The costs specific to the gains are plotted separately.

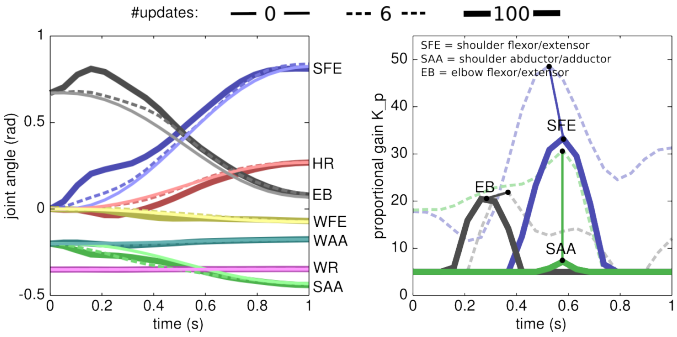
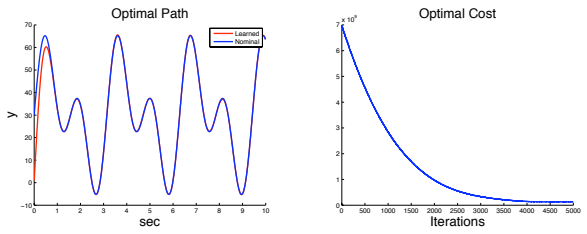


Fig. 3. Learned joint angle trajectories (center) and gain schedules (right) of the CBi arm after 0/6/100 updates. The gain schedules of only three joints have been depicted for sake of clarity.



(a) Rhythmic task (b) Cost

Fig. 4. Learning a rhythmic task with dynamic movement primitives. On figure (a) the desired(blue) and the learned(red) behaviors are illustrated. Figure (b) illustrates the convergence of iterative path integral control.

6.3 Task 3: Learning limit cycle nonlinear attractors

Learning a complex rhythmic behavior is of critical importance for tasks such as locomotion and squatting. Therefore with goal towards learning locomotion in this section we apply PI^2 to learn a rhythmic trajectory which result from the superposition of sinusoid functions. The nominal behavior us for this task is generated as $y_{nom}(t) = y_{offset} + \sum_{j=1}^{2} A_j cos(\omega\ t + \phi)$ where y_{offset} is the offset of the rhythmic behavior, $A_{1,2}$ are the amplitudes, $\omega_{1,2}$ are the frequencies and $\phi_{1,2}$ the phases. For our particular simulation we picked $y_{offset} = 30$, $A_1 = A_2 = 20$,$\omega_1 = 1, \omega_2 = 2$ and $\phi_1 = \phi_2 = 0$. The cost used for this task is defined as $r_t = \sum_{t_o}^{t_N} \delta(t - t_s) \left(y_{nom}(t) - y(t)\right)^2$ where $\delta(t - t_s) = 1$ for $t = t_s$ and $\delta(t - t_s) = 0$ otherwise. We use a rhythmic movement primitive with 10 basis functions, sufficient to match complex rhythmic behaviors. The results are illustrated in figure 4. The baseline of the limit cycle attractor $g = 0$ was initialized far away from the offset of the desired task $y_{offset} = 30$. PI^2 learns the desired behavior under any initialization of the baseline g.

[1] The relatively high exploration noise for the joint trajectories does not express less exploration per se, but is rather due to numerical differences in using the function approximator to model the gains directly rather than as the non-linear component of a DMP.

6.4 Discussion

In Task 1, jumping over the gap, the improvement in cost corresponds to about 15 cm improvement in jump distance, which changed the robot's behavior from an initial barely successful jump to jump that completely traversed the gap with entire body. This learned behavior allowed the robot to traverse a gap at much higher speed in a competition on learning locomotion. It is worth noticing that the jumping task involves contact with the ground which results in non-smooth dynamics and cost function. That is actually the reason of the high variability in the first updates of PI^2 which is reduced as PI^2 settles to the optimal solution in which the little dog robot completely traverses the gap.

For the task of opening the door there are two distinct phases during learning. In the first few updates, the gains are increased in order to achieve the task, i.e. flip the light switch, or open the door. This leads to a strong decrease in the cost for not achieving the task, which is traded off against a higher cost for higher gains. This is clearly seen in Fig. 2, where the cost due to the gains increases dramatically in the first few updates, whereas the overall cost decreases. Essentially, the robot is learning that it is able to solve the task with high-gain control [2] . In the second phase, the exact timing and magnitudes of the gains required to achieve the task are determined. On the CBi robot, there is a peak in the elbow joint before contact, as the elbow must be lifted to reach the door. During door opening, the gains of the shoulder flexor-extensor joint (SFE) increase, again to exert the force necessary to open the door. Too much compliance during this time will not allow the robot to achieve its task. After 100 updates, the sum of the gains (i.e. the 'cost due to gains' in Fig. 2) for the CBi robot is actually 25% lower than at initialization, when it could not open the door. But by timing and tuning the gains appropriately as depicted in Fig. 3, the robot is now able to open the door.

Learning rhythmic behaviors is an essential step towards learning robotic tasks such as locomotion and squatting. Task 3 demonstrates that PI^2 is able to learn complex rhythmic tasks under any initialization of the baseline of the nonlinear limit cycle attractor. Moreover PI^2 learns rhythmic task in multi-agent scenarios in which every DOF corresponds to a nonlinear limit cycle attractor. Due to space limitations we do not show these results here.

7. CONCLUSIONS

In summary, in this paper we have presented PI^2 an iterative version of Path Integral Control capable of scaling to high dimensional optimal control and planning problems. We have presented an analysis on iterative version of Path Integral Control PI^2 and find the sufficient conditions under which convergence is achieved. Moreover we have demonstrated the applicability of PI^2 to the robotics tasks of jumping over a gap and opening a door. Finally with the goal towards learning locomotion we have shown learning of nonlinear limit cycle attractors. Future work will continue applications of PI^2 to limit cycle attractors for learning locomotion as well as object manipulation tasks.

[2] This is also apparent when inspecting the (dashed) gain schedules after a few updates (2/4/6) Fig. 3: the gains are much higher than their low values with which they are initialized

REFERENCES

Broek, B.v.d., Wiegerinck, W., and Kappen., H.J. (2008). Graphical model inference in optimal control of stochastic multi-agent systems. *Journal of Artificial Intelligence Research*, 32(1), 95–122.

Buchli, J., Theodorou, E., Stulp, F., and Schaal, S. (2010). Variable impedance control - a reinforcement learning approach. In *Robotics: Science and Systems Conference (RSS)*.

Cheng, G., Hyon, S., Morimoto, J., Ude, A., Hale, J., Colvin, G., Scroggin, W., and Jacobsen, S.C. (2007). Cb: A humanoid research platform for exploring neuroscience. *Journal of Advanced Robotics*, 21(10), 1097–1114.

Fleming, W.H. and Soner, H.M. (2006). *Controlled Markov processes and viscosity solutions*. Applications of mathematics. Springer, New York, 2nd edition.

Ijspeert, A., Nakanishi, J., and Schaal, S. (2003). Learning attractor landscapes for learning motor primitives. In S. Becker, S. Thrun, and K. Obermayer (eds.), *Advances in Neural Information Processing Systems 15*, 1547–1554. Cambridge, MA: MIT Press.

Izawa, J., Rane, T., Donchin, O., and Shadmehr, R. (2008). Motor adaptation as a process of reoptimization. *Journal Of Neuroscience*, 28(11), 2883–2891.

Jacobson, D.H. and Mayne, D.Q. (1970). *Differential dynamic programming*. American Elsevier Pub. Co., New York,.

Kappen, H.J. (2005a). Linear theory for control of nonlinear stochastic systems. *Phys Rev Lett*, 95, 200201. Journal Article United States.

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

Kappen, H.J. (2007). An introduction to stochastic control theory, path integrals and reinforcement learning. In J. Marro, P.L. Garrido, and J.J. Torres (eds.), *Cooperative Behavior in Neural Systems*, volume 887 of *American Institute of Physics Conference Series*, 149–181.

Li, W., Todorov, E., and Pan, X. (2004). Hierarchical optimal control of redundant biomechanical systems. In *in 26th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society*.

Oksendal, B.K. (2003). *Stochastic differential equations : an introduction with applications*. Springer, Berlin ; New York, 6th edition.

Papageorgiou, M. and Bauschert, T. (1994). Stochastic optimal control of moving vehicles in a dynamic environment. In *International Journal of Robotic Research*, volume 13, 342–354.

Peters, J. and Schaal, S. (2008). Reinforcement learning of motor skills with policy gradients. *Neural Netw*, 21(4), 682–97.

Schaal, S. (2009). The SL simulation and real-time control software package. Technical report, University of Southern California.

Sciavicco, L. and Siciliano, B. (2000). *Modelling and control of robot manipulators*. Advanced textbooks in control and signal processing. Springer, London ; New York.

Stengel, R.F. (1994). *Optimal control and estimation*. Dover books on advanced mathematics. Dover Publications, New York.

Stulp, F., Buchli, J., Theodorou, E., and Schaal, S. (2010). Reinforcement learning of full-body humanoid motor skills. In *10th IEEE-RAS International Conference on Humanoid Robots*.

Theodorou, E. (2011). *Iterative Path Integral Stochastic Optimal Control: Theory and Applications to Motor Control*. Ph.D. thesis, university of southern California.

Theodorou, E., Buchli, J., and Schaal, S. (2010a). A generalized path integral approach to reinforcement learning. *Journal of Machine Learning Research*, (11), 3137–3181.

Theodorou, E., Buchli, J., and Schaal, S. (2010b). Reinforcement learning of motor skills in high dimensions: A path integral approach. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Todorov, E. (2005). Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system. *Neural Computation*, 17(5), 1084.