Free Energy based Policy Gradients

Evangelos A. Theodorou[1], Jiri Najemnik[2] , and Emo Todorov[2]

Abstract— **Despite the plethora of reinforcement learning algorithms in machine learning and control, the majority of the work in this area relies on discrete time formulations of stochastic dynamics. In this work we present a new policy gradient algorithm for reinforcement learning in continuous state action spaces and continuous time for free energy-like cost functions. The derivation is based on successive application of Girsanov's theorem and the use of the Radon Nikodým derivative as formulated for Markov diffusion processes. The resulting policy gradient is reward weighted. The use of Radon Nikodým extends analysis and results to more general models of stochasticity in which jump diffusions processes are considered. We apply the resulting algorithm in two simple examples for learning attractor landscapes in rhythmic and discrete movements.**

I. Introduction

Over the past fifteen years, there has been a significant amount of work in the area of reinforcement learning with applications to learning and control of nonlinear dynamical systems in continuous state-action spaces [1]–[3], [6]–[8], [14], [16], [18]–[22], [25]–[27], [30]. While reinforcement learning [24] has its origins in psychology and behavioral sciences, many researchers in control theory and machine learning have shown its links to traditional optimal and adaptive control, supervised and unsupervised learning. Among the different formalizations of reinforcement learning, Policy Gradient methods (PG) appear to be an efficient approach for hard control and planning problems.

One of the advantages of reinforcement learning based on PGs is the fact that it is a model free approach. This characteristic makes PG methods suitable for cases where the model of the underlying dynamics is either very difficult to be identified or not accurate. In addition, PG methods are easily applicable in practice. In a typical reinforcement learning scenario, a parameterized policy is formulated and the parameters are learned based on rollouts (state trajectories) of the stochastic dynamics. These rollouts may result from the evaluation of the parameterized policy on the real physical system.

In this work, we derive PGs for cost functions that have the form of free energy. Free energy functions appear in many areas of sciences and engineering from statistical physics and thermodynamics to stochastic optimal control, robust control and estimation. In statistical mechanics and thermodynamics free energy plays an essential role since it corresponds to the amount of energy that can be extracted or used to perform work. Moreover there is a dual relationship between free energy and relative entropy that is mathematically represented by the Legendre-Fenchel transformation [29]. In stochastic optimal control [9], [10], [15] value functions can be represented as free energy evaluated on state trajectories given by forward sampling of the uncontrolled dynamics. Furthermore, free energy cost functions have been used in robust control and stochastic differential games as in [4], [5].

In spite of the plethora of PG reinforcement learning approaches and variations, most algorithms are derived for stochastic dynamical systems in discrete time. Exception is the work in [11], [28], [32] where the mathematical derivations are in continuous time. The choice of working in continuous time allows the use of the machinery of stochastic calculus. In particular, in contrast to [14], [18]–[20] and in agreement with [32] our derivation is based on successive applications of Girsanov's theorem and the use of Radon-Nikodým derivative for nonlinear Markov diffusion processes affine in noise. The use of Radon-Nikodým derivative is essential for the derivation of PG update rule in continuous time. Moreover, even though equivalent formulations can be found by just passing the gradient inside the expectation of the cost function, the analysis based on Radon-Nikodým derivative is advantageous since its formalism allows easy extensions to general models of stochasticity. These observations will become clear in the analysis and derivations regarding PGs for jump diffusion processes. Finally, we provide discrete time approximation errors for the resulting free energy PGs and show how these errors depend on the design of the cost function and the discretization of the corresponding continuous dynamics.

The paper is organized as follows: In Section II, the reinforcement problem is formulated subject to the stochastic dynamics in continuous time. Section III includes the derivation of the free energy PGs in continuous time for general nonlinear parameterizations. In Section IV we provide the form of free energy PGs for polices linearly parameterized with time invariant parameters. In addition we provide the approximation of the resulting PG in discrete time and derive the approximation error. In Section V, we show that the analysis based on the Radon-Nikodým derivative is easily extended to jump diffusion processes. In Section VI, the proposed policy gradient is tested on learning attractor land-scapes for discrete and rhythmic primitives. Finally, in Section VIII, we conclude our work and provide future directions.

[1] Evangelos A. Theodorou is a Postdoctoral Research Associate with the Department of Computer Science and Engineering,University of Washington, Seattle. etheodor@cs.washington.edu

[2] Jiri Najemnik is an Acting Assistant Professor in Applied Math, University of Washington, Seattle. najemnik@uw.edu

[3] Emo Todorov is Associate professor with the Departments of Computer Science and Engineering, and Applied Math, University of Washington, Seattle. todorov@cs.washington.edu

II. Problem Formulation

We consider the stochastic dynamical system of the form:

$$d\mathbf{x}(t) = \mathbf{F}(\mathbf{x}, \mathbf{u}, t)dt + \mathbf{C}(\mathbf{x}, t)d\mathbf{w}(t) \tag{1}$$

in which $\mathbf{x} \in \Re^{n\times 1}$ is the state, $\mathbf{u} \in \Re^{p\times 1}$ the controls and $d\mathbf{w} \in \Re^{m}$ is brownian noise. The functions $\mathbf{F}(\mathbf{x}, \mathbf{u}, t)$ and $C(\mathbf{x}, t)$ are defined as $\mathbf{F} : \Re^n \times \Re^p \times \Re \rightarrow \Re^n$ and $\mathbf{C}(\mathbf{x}, t) : \Re^n \times \Re \rightarrow \Re^n \times \Re^{n\times m}$. In this paper we will assume that control is parameterized as $\mathbf{u} = \mathbf{u}(\mathbf{x}, \boldsymbol{\theta})$ with the term $\boldsymbol{\theta}$ denoting the parameters. We also consider the objective function $\xi(\mathbf{x}', \boldsymbol{\theta})$ defined as follows:

$$\xi(\mathbf{x}', \boldsymbol{\theta}) = -\frac{1}{|\rho|}\log J(\mathbf{x}', \boldsymbol{\theta}) \tag{2}$$

The term $J(\mathbf{x}', \boldsymbol{\theta})$ is defined as follows:

$$\begin{aligned} J(\mathbf{x}', \boldsymbol{\theta}) &= \mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}},\boldsymbol{\theta})}\Big(S(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}, \boldsymbol{\theta}))\Big) \\ &= \int S(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}, \boldsymbol{\theta}))d\mathbb{P}(\vec{\mathbf{x}}; \boldsymbol{\theta}) \end{aligned} \tag{3}$$

The symbol $\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}$ denotes the expectation under the probability measure $\mathbb{P}(\vec{\mathbf{x}}; \boldsymbol{\theta})$ which corresponds to (1) and it is parameterized by $\boldsymbol{\theta}$ due to control parameterization. Thus the expectation $\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}$ is taken with respect to trajectories $\vec{\mathbf{x}} = (\mathbf{x}', \mathbf{x}_1, ..., \mathbf{x}_N)$ starting from state $\mathbf{x}' = \mathbf{x}(t_0)$ and generated with forward sampling of (1) under the policy parameter $\boldsymbol{\theta}$. The term $S(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}, \boldsymbol{\theta}))$ is a cost function which depends on the state and control trajectories $\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}, \boldsymbol{\theta})$ with the control trajectory defined as $\vec{\mathbf{u}}(\mathbf{x}, \boldsymbol{\theta}) = (\mathbf{u}(\mathbf{x}', t_0, \boldsymbol{\theta}), \mathbf{u}(\mathbf{x}_1, t_1, \boldsymbol{\theta})..., \mathbf{u}(\mathbf{x}_{N-1}, t_{N-1}, \boldsymbol{\theta}))$. The case in which $S(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}, \boldsymbol{\theta}))$ is defined as $S(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}, \boldsymbol{\theta})) = \exp\left(-|\rho|\mathcal{L}(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}, \boldsymbol{\theta}))\right)$ is of particular interest since under this definition the objective function (6) takes the form of free energy performance criterion specified as:

$$\xi(\mathbf{x}', \boldsymbol{\theta}) = -\frac{1}{|\rho|}\log\int \exp\left(-|\rho|\mathcal{L}(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}, \boldsymbol{\theta}))\right)d\mathbb{P}(\vec{\mathbf{x}}; \boldsymbol{\theta}) \tag{4}$$

The term $|\rho|$ in the equation above could be interpreted as $|\rho| = \frac{1}{kT}$ with $k = 1.3806503\times 10^{-23}\frac{J}{K}$ the Boltzmann constant and T the temperature, while $\mathcal{L}(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}, \boldsymbol{\theta}))$ plays the role of energy. Performance criteria of the form as in (4) are important due to the fact that they can lead to optimization of risk seeking functions. To see that one can show that for $\rho \rightarrow 0$ the objective function above corresponds to: $\xi(\mathbf{x}', \boldsymbol{\theta}) \approx \mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big(\mathcal{L}(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}; \boldsymbol{\theta}))\Big) - \frac{|\rho|}{2}\mathbb{VAR}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big(\mathcal{L}(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}; \boldsymbol{\theta}))\Big) + O(\rho^2)$ where the term $\mathbb{VAR}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}$ is the variance under $\mathbb{P}(\vec{\mathbf{x}}; \boldsymbol{\theta})$.

Alternatively one may consider risk sensitive objective functions of the form:

$$\xi(\mathbf{x}', \boldsymbol{\theta}) = \frac{1}{|\rho|}\log\int \exp\left(|\rho|\mathcal{L}(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}; \boldsymbol{\theta}))\right)d\mathbb{P}(\vec{\mathbf{x}}; \boldsymbol{\theta}) \tag{5}$$

Similarly for $\rho \rightarrow 0$ we will have that $\xi(\mathbf{x}', \boldsymbol{\theta}) \approx \mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big(\mathcal{L}(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}; \boldsymbol{\theta}))\Big) + \frac{|\rho|}{2}\mathbb{VAR}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big(\mathcal{L}(\vec{\mathbf{x}}, \vec{\mathbf{u}}(\mathbf{x}; \boldsymbol{\theta}))\Big) + O(\rho^2)$.

In this work we derive policy gradients that minimize objective functions of the form (4). Thus the problem formulation is expressed as:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\,\xi(\mathbf{x}, \boldsymbol{\theta}) \tag{6}$$

with $\xi(\mathbf{x}, \boldsymbol{\theta})$ defined in (4) subject to the stochastic dynamics in (1).

III. Nonlinear Free Energy Policy Gradients .

In this section we derive the gradient of the performance criterion as in (4). Our analysis is based on the use of Girsanov's theorem and the Radon-Nikodým derivative [32] and applied for case of free-energy like performance criterion. In addition, the results in this section hold for nonlinear policy parameterizations and stochastic dynamics affine with respect to the terms of noise as in (1).

For simplicity we will treat the parameter $\boldsymbol{\theta}$ as scalar and then we will extend our result to the vector case. More precisely we will have:

$$\begin{aligned} \lim_{\delta\boldsymbol{\theta}\rightarrow 0}\frac{\delta\xi(\mathbf{x}', \boldsymbol{\theta})}{\delta\boldsymbol{\theta}} &= -\frac{1}{|\rho|J(\mathbf{x}', \boldsymbol{\theta})}\lim_{\delta\boldsymbol{\theta}\rightarrow 0}\frac{\delta J(\mathbf{x}', \boldsymbol{\theta})}{\delta\boldsymbol{\theta}} \\ &= -\frac{1}{|\rho|J(\mathbf{x}', \boldsymbol{\theta})}\lim_{\delta\boldsymbol{\theta}\rightarrow 0}\left(\frac{J(\mathbf{x}', \boldsymbol{\theta}+\delta\boldsymbol{\theta}) - J(\mathbf{x}', \boldsymbol{\theta})}{\delta\boldsymbol{\theta}}\right) \end{aligned} \tag{7}$$

Next we work with the expression inside the parenthesis in the last equation (7). Therefore we will have:

$$\begin{aligned} &\lim_{\delta\boldsymbol{\theta}\rightarrow 0}\frac{\delta J}{\delta\boldsymbol{\theta}} = \\ &= \lim_{\delta\boldsymbol{\theta}\rightarrow 0}\frac{\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}\Big(S(\vec{\mathbf{x}}, \boldsymbol{\theta}+\delta\boldsymbol{\theta})\Big) - \mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big(S(\vec{\mathbf{x}}, \boldsymbol{\theta})\Big)}{\delta\boldsymbol{\theta}} \\ &= \lim_{\delta\boldsymbol{\theta}\rightarrow 0}\frac{\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big(S(\vec{\mathbf{x}}, \boldsymbol{\theta}+\delta\boldsymbol{\theta})\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big) - \mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big(S(\vec{\mathbf{x}}, \boldsymbol{\theta})\Big)}{\delta\boldsymbol{\theta}} \\ &= \lim_{\delta\boldsymbol{\theta}\rightarrow 0}\frac{\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}},\boldsymbol{\theta})}\Big(S(\vec{\mathbf{x}}, \boldsymbol{\theta}+\delta\boldsymbol{\theta})\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})} - S(\vec{\mathbf{x}}, \boldsymbol{\theta})\Big)}{\delta\boldsymbol{\theta}} \\ &= \lim_{\delta\boldsymbol{\theta}\rightarrow 0}\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left(\frac{S(\vec{\mathbf{x}}, \boldsymbol{\theta}+\delta\boldsymbol{\theta})\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})} - S(\vec{\mathbf{x}}, \boldsymbol{\theta})\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}}{\delta\boldsymbol{\theta}}\right) \\ &+ \lim_{\delta\boldsymbol{\theta}\rightarrow 0}\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}},\boldsymbol{\theta})}\left(\frac{S(\vec{\mathbf{x}}, \boldsymbol{\theta})\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})} - S(\vec{\mathbf{x}}, \boldsymbol{\theta})}{\delta\boldsymbol{\theta}}\right) \end{aligned}$$

$$= \lim_{\delta\boldsymbol{\theta}\to 0} \frac{\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}},\boldsymbol{\theta})}\left[\left(S(\vec{\mathbf{x}},\boldsymbol{\theta}+\delta\boldsymbol{\theta}) - S(\vec{\mathbf{x}},\boldsymbol{\theta})\right)\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\right]}{\delta\boldsymbol{\theta}}$$
$$+ \lim_{\delta\boldsymbol{\theta}\to 0} \frac{\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}},\boldsymbol{\theta})}\left[S(\vec{\mathbf{x}},\boldsymbol{\theta})\left(\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})} - 1\right)\right]}{\delta\boldsymbol{\theta}}$$

We incorporate the last line into (7) we get the result:

$$\lim_{\delta\boldsymbol{\theta}\to 0} \frac{\delta\xi(\mathbf{x}',\boldsymbol{\theta})}{\delta\boldsymbol{\theta}} = -\frac{1}{|\rho| J(\mathbf{x}',\boldsymbol{\theta})}\mathfrak{M}(\mathbf{x}',\boldsymbol{\theta}) \tag{8}$$

where the term $\mathfrak{M}(\mathbf{x},\boldsymbol{\theta})$ in the equation above, is defined as follows:

$$\mathfrak{M}(\mathbf{x}',\boldsymbol{\theta}) = \lim_{\delta\boldsymbol{\theta}\to 0}\left(\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[\frac{\delta S(\vec{\mathbf{x}},\boldsymbol{\theta})}{\delta\boldsymbol{\theta}}\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\right]\right)$$
$$+ \lim_{\delta\boldsymbol{\theta}\to 0}\left(\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[\frac{S(\vec{\mathbf{x}},\boldsymbol{\theta})}{\delta\boldsymbol{\theta}}\left(\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})} - 1\right)\right]\right) \tag{9}$$

The next step is to find the limit of the expression in (9) as $\delta\boldsymbol{\theta} \to 0$. To do so we make use of Girsanov's theorem and the Radon-Nikodým derivative as applied to nonlinear diffusion processes. More precisely the ratio of the probability measures $\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}$ (see appendix) is specified as follows:

$$\frac{d\mathbb{P}\left(\mathbf{x}_N, t_N|\mathbf{x}_0, t_0;\boldsymbol{\theta}+\delta\boldsymbol{\theta}\right)}{d\mathbb{P}\left(\mathbf{x}_N, t_N|\mathbf{x}_0, t_0;\boldsymbol{\theta}\right)} =$$
$$\exp\left[\int_{t_0}^{t_N}\delta\mathbf{F}^T\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}\mathbf{C}(\mathbf{x},t)d\mathbf{w}_\theta(t)\right]$$
$$\times\exp\left[-\frac{1}{2}\int_{t_0}^{t_N}\delta\mathbf{F}^T\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}\delta\mathbf{F}dt\right] \tag{10}$$

The term $\delta\mathbf{F}$ above is defined as the difference $\delta\mathbf{F} = \mathbf{F}(\mathbf{x},\mathbf{u}(\mathbf{x},\boldsymbol{\theta}+\delta\boldsymbol{\theta}),t) - \mathbf{F}(\mathbf{x},\mathbf{u}(\mathbf{x},\boldsymbol{\theta}),t)$ and we assume that $\lim_{\delta\boldsymbol{\theta}\to 0}\delta\mathbf{F} = 0$. In addition the term $\boldsymbol{\Sigma}_{\mathbf{C}}$ is defines as $\boldsymbol{\Sigma}_C = \mathbf{C}(\mathbf{x},t)\mathbf{C}(\mathbf{x},t)^T$. Next we find the terms $\lim_{\delta\boldsymbol{\theta}\to 0}\left(\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}},\boldsymbol{\theta})}\left[\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}},\boldsymbol{\theta})}\right]\right)$ and $\lim_{\delta\boldsymbol{\theta}\to 0}\left(\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[\frac{1}{\delta\boldsymbol{\theta}}\left(\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})} - 1\right)\right]\right)$. Since $e^x - 1 = x + \frac{x}{2!} + \frac{x^2}{3!}$ it can be shown that that:

$$\lim_{\delta\boldsymbol{\theta}\to 0}\frac{1}{\delta\boldsymbol{\theta}}\left(\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}},\boldsymbol{\theta})} - 1\right) =$$
$$= \int_{t_0}^{t_N}\frac{\delta\mathbf{F}}{\delta\boldsymbol{\theta}}^T\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}\mathbf{C}(\mathbf{x},t)d\mathbf{w}_\theta(t) \tag{11}$$

Also $\lim_{\delta\boldsymbol{\theta}\to 0}\left(\frac{d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta}+\delta\boldsymbol{\theta})}{d\mathbb{P}(\vec{\mathbf{x}},\boldsymbol{\theta})}\right) = 1$ in the mean sense and therefore the final result will be:

$$\lim_{\delta\to 0}\frac{\delta\xi(\mathbf{x}',\boldsymbol{\theta})}{\delta\boldsymbol{\theta}} = -\frac{1}{|\rho| J(\mathbf{x}',\boldsymbol{\theta})}$$
$$\times\left(\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[\frac{\delta S(\vec{\mathbf{x}},\boldsymbol{\theta})}{\delta\boldsymbol{\theta}}\right] + \mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[S(\vec{\mathbf{x}},\boldsymbol{\theta})\mathbf{dw}\right]\right) \tag{12}$$

where the term $\mathbf{dw}$ is defined as:

$$\mathbf{dw} = \int_{t_0}^{t_N}\frac{\delta\mathbf{F}}{\delta\boldsymbol{\theta}}^T\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}\mathbf{C}(\mathbf{x},t)d\mathbf{w}_\theta(t)$$

When $\boldsymbol{\theta}$ is a parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_\nu)$ then we will have that:

$$\mathbf{dw}_i = \int_{t_0}^{t_N}\frac{\delta\mathbf{F}}{\delta\boldsymbol{\theta}_i}^T\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}\mathbf{C}(\mathbf{x},t)d\mathbf{w}_\theta(t),\ \forall i = 1,...,\nu$$

We summarize the analysis above with the following nonlinear policy gradient lemma:

Lemma 1: *Consider the stochastic dynamics as in (1) the nonlinear stochastic gradient of the free energy performance criterion:*

$$\xi(\mathbf{x}',\boldsymbol{\theta}) = -\frac{1}{|\rho|}\log\int S(\vec{\mathbf{x}},\vec{\mathbf{u}}(\mathbf{x},\boldsymbol{\theta}))d\mathbb{P}$$

is expressed as follows:

$$\nabla_{\boldsymbol{\theta}}\xi(\mathbf{x}',\boldsymbol{\theta}) = -\frac{1}{|\rho| J(\mathbf{x}',\boldsymbol{\theta})}$$
$$\times\left(\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[\nabla_{\boldsymbol{\theta}}S(\vec{\mathbf{x}},\boldsymbol{\theta})\right] + \mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[S(\vec{\mathbf{x}},\boldsymbol{\theta})\mathbf{dw}\right]\right)$$

The term $\mathbf{dw}$ *is defined as follows:*

$$\mathbf{dw} = \int_{t_0}^{t_N}\mathbf{J}_{\boldsymbol{\theta}}\mathbf{F}^T\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}\mathbf{C}(\mathbf{x},t)d\mathbf{w}_\theta(t) \tag{13}$$

The term $\mathbf{J}_{\boldsymbol{\theta}}\mathbf{F}$ *is the Jacobian of the drift term* $\mathbf{F}(\mathbf{x},\mathbf{u}(\mathbf{x},\boldsymbol{\theta}),t)$ *w.r.t parameter vector* $\boldsymbol{\theta} \in \Re^{\nu\times 1}$*. In addition the term* $\boldsymbol{\Sigma}_C$ *is defined as* $\boldsymbol{\Sigma}_C = \mathbf{C}(\mathbf{x},t)\mathbf{C}(\mathbf{x},t)^T$.

Next we consider the case where the cost $S(\mathbf{x},\mathbf{u}(\mathbf{x};\boldsymbol{\theta}))$ has the form of an exponential $S(\mathbf{x},\mathbf{u}(\mathbf{x},\boldsymbol{\theta})) = \exp\left(-|\rho|\mathcal{L}(\mathbf{x},\mathbf{u}(\mathbf{x},\boldsymbol{\theta}))\right)$. The gradient of the objective function can be further formulated as follows:

$$\nabla_{\boldsymbol{\theta}}\xi(\mathbf{x}',\boldsymbol{\theta}) = -\frac{1}{J(\mathbf{x}',\boldsymbol{\theta})}$$
$$\times\frac{1}{|\rho|}\left(\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[S(\vec{\mathbf{x}},\boldsymbol{\theta})\mathbf{dw}\right] + \mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[\nabla_{\boldsymbol{\theta}}S(\vec{\mathbf{x}},\boldsymbol{\theta})\right]\right)$$
$$= -\frac{1}{J(\mathbf{x}',\boldsymbol{\theta})}$$
$$\times\left(\frac{1}{|\rho|}\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[S(\vec{\mathbf{x}},\boldsymbol{\theta})\mathbf{dw}\right] - \mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[\nabla_{\boldsymbol{\theta}}\mathcal{L}(\vec{\mathbf{x}},\boldsymbol{\theta})S(\vec{\mathbf{x}},\boldsymbol{\theta})\right]\right)$$
$$= -\frac{1}{|\rho|}\mathbb{E}_{\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[\mathbf{dw}\right] + \mathbb{E}_{\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})}\left[\nabla_{\boldsymbol{\theta}}\mathcal{L}(\vec{\mathbf{x}},\boldsymbol{\theta})\right]$$

The term $\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})$ is defined as follows:

$$
\begin{aligned}
d\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta}) &= \frac{S(\vec{\mathbf{x}},\boldsymbol{\theta})d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}{\int S(\vec{\mathbf{x}},\boldsymbol{\theta})d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})} \\
&= \frac{\exp\left(-|\rho|\mathcal{L}(\vec{\mathbf{x}},\vec{\mathbf{u}}(\mathbf{x};\boldsymbol{\theta}))\right)d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}{\int \exp\left(-|\rho|\mathcal{L}(\vec{\mathbf{x}},\vec{\mathbf{u}}(\mathbf{x};\boldsymbol{\theta}))\right)d\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})} \qquad (14)
\end{aligned}
$$

The analysis above is summarized by the following nonlinear policy gradient lemma:

Lemma 2: *Consider the stochastic dynamics as in (1) the nonlinear stochastic gradient of the free energy performance criterion:*

$$
\xi(\mathbf{x}',\boldsymbol{\theta}) = -\frac{1}{|\rho|}\log\int \exp\left(-|\rho|\mathcal{L}(\vec{\mathbf{x}},\vec{\mathbf{u}}(\mathbf{x},\boldsymbol{\theta}))\right)d\mathbb{P}
$$

is expressed as follows:

$$
\nabla_{\boldsymbol{\theta}}\xi(\mathbf{x}',\boldsymbol{\theta}) = -\frac{1}{|\rho|}\mathbb{E}_{\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big[\mathbf{dw}\Big] + \mathbb{E}_{\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big[\nabla_{\boldsymbol{\theta}}\mathcal{L}(\vec{\mathbf{x}},\boldsymbol{\theta})\Big]
$$

The term $\mathbf{dw}$ *is defined as in (13) while the expectation* $\mathbb{E}_{\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})}$ *is under* $d\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})$ *defined as in (14):*

IV. LINEAR FREE ENERGY POLICY GRADIENTS.

A special class of systems of this form in (1) may include dynamics affine in controls. Such dynamics are formulated as follows:

$$
\begin{aligned}
\begin{pmatrix} d\mathbf{x}_1(t) \\ d\mathbf{x}_2(t) \end{pmatrix} &= \begin{pmatrix} \mathbf{f}_1(\mathbf{x},t) \\ \mathbf{f}_2(\mathbf{x},t) \end{pmatrix} dt + \begin{pmatrix} 0 \\ \boldsymbol{\mathcal{B}}(\mathbf{x},t) \end{pmatrix} \mathbf{u}dt \\
&+ \begin{pmatrix} 0 \\ \frac{1}{\sqrt{|\rho|}}\mathbf{L}(\mathbf{x},t) \end{pmatrix} d\mathbf{w}(t) \qquad (15)
\end{aligned}
$$

with the terms $\mathbf{x}_1 \in \Re^{n-k}$, $\mathbf{x}_2 \in \Re^{k}$ and $\mathbf{B}(\mathbf{x},t) : \Re^n \times \Re \rightarrow \Re^{k\times p}$ denoting the control matrix, $\mathbf{f}_1(\mathbf{x},t) \in \Re^n \times \Re \rightarrow \Re^{n-k}$ and $\mathbf{f}_1(\mathbf{x},t) : \Re^n \times \Re \rightarrow \Re^{k\times p}$ denoting the passive dynamics and $\mathbf{L}(\mathbf{x},t) \in \Re^n \times \Re \rightarrow \Re^{k\times m}$ being the diffusions matrix.

In this section we consider deterministic and linear policy parameterizations of the form:

$$
\mathbf{u}(\mathbf{x},t,\boldsymbol{\theta}) = \boldsymbol{\Phi}(\mathbf{x},t)\boldsymbol{\theta} \qquad (16)
$$

The term $\boldsymbol{\Phi} \in \Re^{p\times\nu}$ corresponds to the basis function of the parameterization and $\boldsymbol{\theta} \in \Re^{\nu\times 1}$ is the parameter vector. Our analysis and derivations are based on parameterizations of the form (16). The case of time dependent parameters together with experiments is ongoing work which will be presented in the near future.

Substitution of the parameterized policy into differential equation (15) results in the stochastic differential equation:

$$
\begin{aligned}
\begin{pmatrix} d\mathbf{x}_1(t) \\ d\mathbf{x}_2(t) \end{pmatrix} &= \begin{pmatrix} \mathbf{f}_1(\mathbf{x},t) \\ \mathbf{f}_2(\mathbf{x},t) \end{pmatrix} dt + \begin{pmatrix} 0 \\ \boldsymbol{\Gamma}(\mathbf{x},t) \end{pmatrix} \boldsymbol{\theta}dt \\
&+ \begin{pmatrix} 0 \\ \frac{1}{\sqrt{|\rho|}}\mathbf{L}(\mathbf{x},t) \end{pmatrix} d\mathbf{w}(t) \qquad (17)
\end{aligned}
$$

in which $\boldsymbol{\Gamma}(\mathbf{x},t) = \boldsymbol{\mathcal{B}}(\mathbf{x},t)\boldsymbol{\Phi}(\mathbf{x},t) : \Re^n \times \Re \rightarrow \Re^{k\times\nu}$. We also introduced the terms $\boldsymbol{\Sigma}_\rho(\mathbf{x},t), \boldsymbol{\Sigma}(\mathbf{x},t) : \Re^n \times \Re \rightarrow \Re^{k\times k}$ defined as $\boldsymbol{\Sigma}_\rho(\mathbf{x},t) = \frac{1}{|\rho|}\mathbf{L}(\mathbf{x},t)\mathbf{L}(\mathbf{x},t)^T = \frac{1}{|\rho|}\boldsymbol{\Sigma}_{\mathbf{L}}$ where $\boldsymbol{\Sigma}_{\mathbf{L}}(\mathbf{x},t) = \mathbf{L}(\mathbf{x},t)\mathbf{L}(\mathbf{x},t)^T$ and the term $\mathbf{F}_2(\mathbf{x},t) : \Re^n \times \Re \rightarrow \Re^k$ defined by the equation $\mathbf{F}_2(\mathbf{x},t) = \mathbf{f}_2(\mathbf{x},t) + \boldsymbol{\Gamma}(\mathbf{x},t)\boldsymbol{\theta}$.

For this special case of parameterized dynamics affine in control as in (17) and linear policy parameterization as in (16) we will have the policy gradient:

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}}\xi(\mathbf{x}',\boldsymbol{\theta}) &= -\frac{1}{J(\mathbf{x}',\boldsymbol{\theta})} \\
&\times \left(\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big[S(\vec{\mathbf{x}},\boldsymbol{\theta})\mathbf{dv}\Big] + \frac{1}{|\rho|}\mathbb{E}_{\mathbb{P}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big[\nabla_{\boldsymbol{\theta}}S(\vec{\mathbf{x}},\boldsymbol{\theta})\Big]\right)
\end{aligned}
$$

with the term $\mathbf{dv}$ is given by the expression that follows:

$$
\mathbf{dv} = \frac{1}{\sqrt{|\rho|}}\int_{t_0}^{t_N} \boldsymbol{\Gamma}(\mathbf{x},t)^T\boldsymbol{\Sigma}_{\mathbf{L}}^{-1}\mathbf{L}(\mathbf{x},t)d\mathbf{w}_\theta(t) \qquad (18)
$$

When the function $S(\vec{\mathbf{x}},\boldsymbol{\theta})$ is defined as $S(\vec{\mathbf{x}},\vec{\mathbf{u}}(\mathbf{x},\boldsymbol{\theta})) = \exp\left(-|\rho|\mathcal{L}(\vec{\mathbf{x}},\vec{\mathbf{u}}(\mathbf{x},\boldsymbol{\theta}))\right)$ then we will have that:

$$
\nabla_{\boldsymbol{\theta}}\xi(\mathbf{x}',\boldsymbol{\theta}) = -\mathbb{E}_{\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big[\mathbf{dv}\Big] + \mathbb{E}_{\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big[\nabla_{\boldsymbol{\theta}}\mathcal{L}(\vec{\mathbf{x}},\boldsymbol{\theta})\Big]
$$

The analysis above is summarized by the following linear policy gradient theorem:

Theorem 1: *Consider the stochastic dynamics as in (17), the stochastic gradient of the free energy performance criterion:*

$$
\xi(\mathbf{x}',\boldsymbol{\theta}) = -\frac{1}{|\rho|}\log\int \exp\left(-|\rho|\mathcal{L}(\vec{\mathbf{x}},\vec{\mathbf{u}}(\mathbf{x},\boldsymbol{\theta}))\right)d\mathbb{P}
$$

subject to the dynamics in (17) is:

$$
\nabla_{\boldsymbol{\theta}}\xi(\mathbf{x}',\boldsymbol{\theta}) = -\mathbb{E}_{\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big[\mathbf{dv}\Big] + \mathbb{E}_{\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})}\Big[\nabla_{\boldsymbol{\theta}}\mathcal{L}(\vec{\mathbf{x}},\boldsymbol{\theta})\Big]
$$

The term $\mathbf{dv}$ *is defined as in (18) while the expectation* $\mathbb{E}_{\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})}$ *is under* $d\mathbb{Q}(\vec{\mathbf{x}};\boldsymbol{\theta})$ *defined as in (14).*

Next we provide ways to numerically implement free energy policy gradients.

A. Numerical Implementation

In this section we show how the free energy policy gradients can be numerically implemented. The basic update equation has the form:

$$
\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \gamma\frac{\delta\xi(\mathbf{x},\boldsymbol{\theta})}{\delta\boldsymbol{\theta}_k} \qquad (19)
$$

where γ is a positive term used for line search. We remind at this point that the parameters $\boldsymbol{\theta}$ are used to parameterize controls in the form $\mathbf{u}(\mathbf{x},t) = \boldsymbol{\Phi}(\mathbf{x},t)\boldsymbol{\theta}$ or $\mathbf{u}(\mathbf{x},t) = \boldsymbol{\Phi}(\mathbf{x})\boldsymbol{\theta}(t)$. At every iteration k these controls are applied to stochastic dynamics to generate trajectories and therefore will be multiplied by δt. We re-write the update rules in the form $\boldsymbol{\theta}_{k+1}dt = \boldsymbol{\theta}_k dt - \gamma\frac{\delta\xi(\mathbf{x},\boldsymbol{\theta})}{\delta\boldsymbol{\theta}_k}dt$ to further simplify mathematical results by finding terms of order $\boldsymbol{\mathcal{O}}(dt^2)$ and assuming that they converge to 0 as $dt \rightarrow 0$. As we will see in the analysis that follows, finding terms of order $\boldsymbol{\mathcal{O}}(dt^2)$

will depend on the design of the cost function and whether the policy parameters appear explicitly in this function.

Next we will consider the discrete time approximation of $\mathbf{dv}$ in (18) which can be written as follows:

$$\mathbf{dv} = \frac{1}{\sqrt{|\rho|}} \sum_{i=1}^{N} \boldsymbol{\Gamma}(\mathbf{x}, t_i)^T \boldsymbol{\Sigma}_{\mathbf{L}}(t_i)^{-1} \mathbf{L}(\mathbf{x}, t_i) d\mathbf{w}(t_i) \tag{20}$$

In a more compact form, the equation above can be further expressed:

$$\boxed{\mathbf{dv} = \frac{1}{\sqrt{|\rho|}} \mathfrak{B}(\vec{\mathbf{x}})^T d\mathbf{W}} \tag{21}$$

The term $\mathfrak{B}(\vec{\mathbf{x}})$ is expressed as follows

$$\mathfrak{B}(\vec{\mathbf{x}}) = \Bigg(\mathfrak{D}_0 \;\Bigg|\; \mathfrak{D}_1 \;\Bigg|\; ... \;\Bigg|\; \mathfrak{D}_N \Bigg)$$

Every submatrix element $\mathfrak{D}_i$ is defined as $\mathfrak{D}_i = \mathfrak{D}(\mathbf{x}(t_i), t_i) = \boldsymbol{\Gamma}(\mathbf{x}, t_i)^T \boldsymbol{\Sigma}_{\mathbf{L}}(t_i)^{-1} \mathbf{L}(\mathbf{x}, t_i)$. The term $d\mathbf{W}$ is given by the equation:

$$d\mathbf{W}^T = \Bigg(d\mathbf{w}(t_0)^T \;\Bigg|\; d\mathbf{w}(t_1)^T \;\Bigg|\; ... \;\Bigg|\; d\mathbf{w}(t_N)^T \Bigg)$$

Next we consider the case where the cost is quadratic in the parameters of the control policy and therefore it has the form: $\mathcal{L}(\mathbf{x}, \mathbf{u}(\mathbf{x}, \boldsymbol{\theta})) = \int_{t_0}^{t_N} \Big(q(\mathbf{x}, t) + \frac{1}{2} \boldsymbol{\theta}^T \mathbf{R} \boldsymbol{\theta} \Big) \, dt$ with the parameter $\mathbf{R}$ being a positive definite matrix. The gradient of the cost term $\mathcal{L}$ with respect to policy parameters will take the form:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{x}, \mathbf{u}(\mathbf{x}, \boldsymbol{\theta})) = \mathbf{R} \boldsymbol{\theta} N dt$$

where the term N is specified as $t_N - t_0 = N dt$. The policy update rules are formulated as follows:

$$\boldsymbol{\theta}_{k+1} dt = \boldsymbol{\theta}_k dt + \frac{\gamma}{\sqrt{|\rho|}} E_{\mathbb{Q}(\vec{\mathbf{x}}; \boldsymbol{\theta})} \Big(\mathfrak{B}(\vec{\mathbf{x}})^T d\mathbf{W} \Big) \delta t + \boldsymbol{\mathcal{O}}_2(dt^2) \tag{22}$$

where the term $\boldsymbol{\mathcal{O}}_2(dt^2)$ is specified as $\boldsymbol{\mathcal{O}}_2(dt^2) = -\mathbf{R} \boldsymbol{\theta} N dt^2$. As $dt \to 0$ the update rule takes the form:

$$\boldsymbol{\theta}_{k+1} dt \approx \boldsymbol{\theta}_k dt + \frac{\gamma}{\sqrt{|\rho|}} E_{\mathbb{Q}(\vec{\mathbf{x}}; \boldsymbol{\theta})} \Big(\mathfrak{B}(\vec{\mathbf{x}})^T d\mathbf{W} \Big) dt \tag{23}$$

For implementation purposes we will have

$$\begin{aligned} \boldsymbol{\theta}_{k+1} dt \approx \boldsymbol{\theta}_k dt + & \\ + \frac{\gamma}{\sqrt{|\rho|}} & \frac{\frac{1}{\#paths} \sum_{j=1}^{\#paths} S(\mathbf{x}_j, \mathbf{u}(\mathbf{x}_j, \boldsymbol{\theta})) \Big(\mathfrak{B}(\vec{\mathbf{x}}_j)^T d\mathbf{W}_j \Big) dt}{\frac{1}{\#paths} \sum_{j=1}^{\#paths} S(\mathbf{x}_j, \mathbf{u}(\mathbf{x}_j, \boldsymbol{\theta}))} \\ = \boldsymbol{\theta}_k dt + & \frac{\gamma}{\sqrt{|\rho|}} \sum_{j=1}^{\#paths} \mathfrak{R}(\vec{\mathbf{x}}_j) \Big(\mathfrak{B}(\vec{\mathbf{x}}_j)^T d\mathbf{W}_j \Big) dt \end{aligned}$$

With the term $\mathfrak{R}(\vec{\mathbf{x}}_j)$ defined as:

$$\mathfrak{R}(\vec{\mathbf{x}}_j) = \frac{\exp\left(-\mathcal{L}(\mathbf{x}_j, \mathbf{u}(\mathbf{x}_j, \boldsymbol{\theta}))\right)}{\sum_{j=1}^{\#paths} \exp\left(-\mathcal{L}(\mathbf{x}_j, \mathbf{u}(\mathbf{x}_j, \boldsymbol{\theta}))\right)} \tag{24}$$

V. Free Energy based Policy Gradients for Markov Jump Diffusions

We consider the markov jump diffusions processes specified as:

$$d\mathbf{x}(t) = \mathbf{F}(\mathbf{x}, \mathbf{u}, t) \delta t + \mathbf{C}(\mathbf{x}, t) d\mathbf{w}(t) + \mathbf{h}(\mathbf{x}) d\mathbf{P}^{(1)}(t) \tag{25}$$

The term $P(t) \in \Re^{q \times 1}$ is Poisson distributed and $\mathbf{h}(\mathbf{x}, t) \in \Re^{n \times q}$ is the jump-amplitude or the Poisson process coefficient with $E\left(d\mathbf{P}(t)^{(i)}\right) = \nu_i \delta t$ and $\text{Var}\left(d\mathbf{P}(t)^{(i)}\right) = \nu_i \delta t$, for $i = 1, ..., m$. The term $\nu(t) > 0$ is the ith jump rate or jump density and $\nu \delta t$ is the mean count of the Poisson process in the time interval $(t, t + dt]$. Poisson processes obey the Markov property while they also have independent increments and thus $\text{Cov}\left[d\mathbf{P}(t_j) d\mathbf{P}(t_k)\right] = \nu(t_j) dt \delta_{k,j}$.

Again we assume that the control is parameterized as in (16). The changes in the parameters of the policy correspond to changes only in the drift of the dynamics and therefore the Radon - Nikodým derivative will have the same form as in (10). To see that we consider the simple example with the one dimensional Markov jump diffusion of the form:

$$\begin{aligned} \mathbb{P}_\theta : \quad dx(t) = & f(x) \delta t + C(x, t) \Big(\theta(t) \delta t + \frac{1}{\sqrt{|\rho|}} dw^\theta(t) \Big) \\ & + h(x) dP^{(1)}(t) \end{aligned}$$

$$\begin{aligned} \mathbb{P}_{\theta + \delta\theta} : dx(t) = & f(x) \delta t + C(x, t) \Big(\theta(t) + \delta\theta \Big) \delta t \\ & + C(x, t) \frac{1}{\sqrt{|\rho|}} dw^{(\theta + \delta\theta)}(t) + h(x) dP^{(1)}(t) \end{aligned}$$

Based on Girsanov's theorem [12] for markov jump diffusion processes, the Radon-Nikodým derivative is now specified as $\frac{d\mathbb{P}_\theta}{d\mathbb{P}_{\theta + \delta\theta}} = \exp\left(-\zeta(\mathbf{u})\right)$ with $\zeta(\mathbf{u})$ defined as follows:

$$\zeta(\mathbf{u}) = -\int_{t_i}^{t_N} \frac{1}{2} |\rho| \delta\theta^2 \delta t + \sqrt{|\rho|} \int_{t_i}^{t_N} \delta\theta dw^{(\theta)}(t) + \mathcal{V}(\gamma^{(J)})$$

$\mathcal{V}(\gamma^{(J)}(t)) = \int_{t_i}^{t_N} \left(\left(\gamma^{(J)}(t) - 1 \right) \nu_0(t) \right) \delta t + \sum_{j=1}^{\mathbf{P}^{(1)}(t)} \log \gamma^{(J)}(t)$ and $\gamma^{(J)}(t) = \frac{\nu^{(1)}(t)}{\nu^{(0)}(t)}$. Since $\nu_0(t) = \nu_1(t)$ we have that $\mathcal{V}(\gamma^{(J)}(t)) = 0$ and we get the Radon-Nikodým for the case of diffusion processes. Therefore all the results derived so far for the case of diffusion processes hold also for the more general case of markov jump diffusions. The analysis above is based on the Radon-Nikodym derivative. Alternative derivations pass explicitly the gradient inside the expectation but that requires the path integral formulation of Markov jump diffusions and makes the derivation much more difficult.

VI. Simulations

We apply the proposed method on two simple examples that include the use of nonlinear point and limit cycle attractors with adjustable landscape. In particular, the first task is for the 1D nonlinear point attractor to pass via a pre-specified target point at a pre-specified time as illustrated in Fig.1. The second task consists of learning a rhythmic sinusoid 1D movement with the nonlinear limit cycle attractor, see 2.

VII. Nonlinear Point Attractors with adjustable attractor Land-scape

In this subsection we provide the mathematical formulation of nonlinear point attractor that consist of two sets of differential equations, the canonical and transformation system which are coupled through a nonlinearity [13]. 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:

$$\begin{aligned}\tau\dot{z} =& \alpha_z\beta_z\left(\left(g+\frac{f}{\alpha_z\beta_z}\right)-y\right)-\alpha_z z \qquad (26)\\ \tau\dot{y} =& z\end{aligned}$$

The nonlinear 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 $K(x_t, c_i) = \exp\left(-0.5h_j(x_t - c_j)^2\right)$ with bandwith h_j and center c_j of the Gaussian kernels – for more details see [13]. The terms y_{t_0} to the goal g are the initial and goal state, 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 full dynamics or the rhythmic movement primitives have the form of $d\mathbf{x} = F(\mathbf{x})dt + \mathbf{G}(\mathbf{x})\boldsymbol{\theta}dt$ where the state $\mathbf{x}$ is specified as $\mathbf{x} = (x, y, z)$ while the controls are specified as $\boldsymbol{\theta} = (\theta_1, ..., \theta_p)^T$.The representation above guarantees attractor properties towards the goal while remaining linear in the parameters $\boldsymbol{\theta}$ of the function approximator. Variations in the parameter $\boldsymbol{\theta}$ will result in changing the shape of the trajectory while the goal g and initial state y_{t_0} remain fixed.

In Fig 1(a) the optimal trajectory is illustrated. The task is for the point attractor to pass via a target point p^* at time t^*. Figure 1(b) illustrates the cost at every update of the policy parameters. The Lagrangian of the cost under minimization for the aforementioned task is $\mathcal{L}(\mathbf{x};\boldsymbol{\theta}) = \int_{t_i}^{t_N}(\mathbf{x}-p^*)^2\delta(\tau-t^*) + 0.5\boldsymbol{\theta}^T\boldsymbol{\theta}\delta\tau$ with $p^* = 0.2$ and $t^* = 30$.

A. Nonlinear Limit Cycle Attractors with adjustable attractor Land-scape

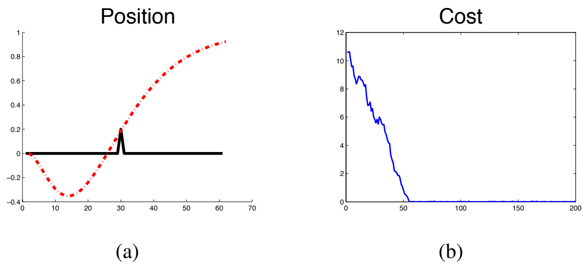


Fig. 1. Passing via a point with nonlinear point attractors. In subfigure (a) the optimal trajectory after learning. In (b) the cost per iteration is illustrated.

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

$$\tau\dot{v}_1 = -\mu\frac{\sqrt{v_1^2+v_2^2}-r}{\sqrt{v_1^2+v_2^2}}v_1 - v_2 \qquad (27)$$

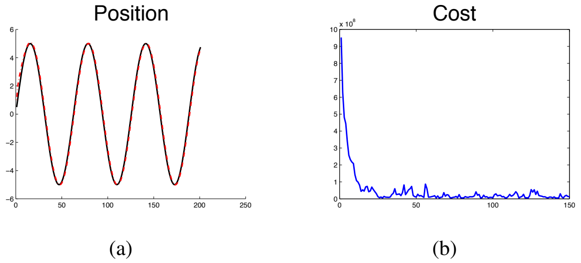
$$\tau\dot{v}_2 = -\mu\frac{\sqrt{v_1^2+v_2^2}-r}{\sqrt{v_1^2+v_2^2}}v_2 + v_1 \qquad (28)$$

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 full dynamics of the rhythmic movement primitives have the form of $d\mathbf{x} = F(\mathbf{x})dt + \mathbf{G}(\mathbf{x})\boldsymbol{\theta}dt$ where the state $\mathbf{x}$ is specified as $\mathbf{x} = (\phi, v_1, v_2, z, y)$. 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.

Figure 2 illustrates the result from the application of Girsanov's direct policy gradient method to learn rhythmic tasks. The task here is to learn a sinusoid movement and therefore the Lagrangian of the cost is $\mathcal{L}(\mathbf{x};\boldsymbol{\theta}) = \int_{t_i}^{t_N}(\mathbf{x} - \cos(\omega\tau))^2\delta\tau$.

VIII. Discussion

We derive free energy policy gradients for nonlinear stochastic dynamics. In contrast to previous work on policy gradient methods, our derivation is in continuous time. Moreover it exploits the use of Radon-Nikodým derivative. This choice allows for direct extensions to general model of stochasticity that include but they are not limited to jump diffusion processes. In addition the resulting formulation of free energy PGs applies to policies that include nonlinear and linear parameterizations. In this work, we specialize our analysis to linear policies and derive update rules which are



(a)

(b)

Fig. 2. Learning 1D sinusoid movement with nonlinear limit cycle attractor. In (a) the target and the learned trajectory are illustrated. In (b) the cost at every time horizon is given until convergence.

easy to implement. Furthermore, we provide approximation errors for the update rule of the free energy PGs and show how this errors depend on the cost function design and discretization of the underlying stochastic dynamics.

There are multiple future directions on the free energy PGs. Clearly, more numerical experiments are required to evaluate the performance of free energy PGs. Therefore future work should include testing and evaluation of the free energy PGs in simulation as well as on real robotic systems for learning motor control tasks. Our particular interest is on learning control for humanoid robotic systems such as manipulators, bipeds, tendon-actuated hands etc. The cases of policy parameterizations with time varying parameters as well as extensions to free energy policy gradients for non-smooth cost functions is ongoing work. An interesting question here is on how approximation errors depend on the choice of policy parameterization. Incorporating subsampling into the derivation of free energy PGs is straightforward and it results in new updates rules. Finally comparisons with current state of the art PGs methods such as Policy Improvement with Path Integrals (PI^2) on a variety of learning control problems is also ongoing work.

IX. Acknowledgments

This work was funded by the US National Science Foundation and DARPA.

X. Appendix

We will consider Girsanov's theorem for the case of stochastic diffusions:

$$d\mathbf{x}(t) = \mathbf{F}(\mathbf{x}, \mathbf{u}(\mathbf{x}, \boldsymbol{\theta}), t)\delta t + \mathbf{C}(\mathbf{x}, t) d\mathbf{w}_{\theta}(t) \tag{29}$$

$$d\mathbf{x}(t) = \mathbf{F}(\mathbf{x}, \mathbf{u}(\mathbf{x}, \boldsymbol{\theta} + \delta\boldsymbol{\theta}), t)\delta t + \mathbf{C}(\mathbf{x}, t) d\mathbf{w}_{\theta+\delta\theta}(t) \tag{30}$$

we also define $\delta\mathbf{F} = \mathbf{F}(\mathbf{x}, \mathbf{u}(\mathbf{x}, \boldsymbol{\theta} + \delta\boldsymbol{\theta}), t)\delta t - \mathbf{F}(\mathbf{x}, \mathbf{u}(\mathbf{x}, \boldsymbol{\theta}), t)\delta t$. We are going to write the probability measures for each one of the diffusions above. More precisely we will have that:

$$d\mathbb{P}\Big(\mathbf{x}_N, t_N | \mathbf{x}_0, t_0; \boldsymbol{\theta}\Big) = \frac{\exp\left(-\frac{1}{2}\left(\int_{t_0}^{t_N} ||\mu(\mathbf{x};\boldsymbol{\theta})||^2_{\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}} \delta t\right)\right)}{(2\pi\delta t)^{m/2}|\boldsymbol{\Sigma}_{\mathbf{C}}|^{1/2}} d\mathbf{x}$$

and

$$d\mathbb{P}\Big(\mathbf{x}_N, t_N | \mathbf{x}_0, t_0; \boldsymbol{\theta} + \delta\boldsymbol{\theta}\Big) = \frac{\exp\left(-\frac{1}{2}\left(\int_{t_0}^{t_N} ||\mu(\mathbf{x};\boldsymbol{\theta}+\delta\boldsymbol{\theta})||^2_{\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}} \delta t\right)\right)}{(2\pi\delta t)^{m/2}|\boldsymbol{\Sigma}_{\mathbf{C}}|^{1/2}} d\mathbf{x}$$

with $\mu(\mathbf{x};\boldsymbol{\theta}) = \frac{\delta\mathbf{x}}{\delta t} - \mathbf{F}(\mathbf{x}, \mathbf{u}(\mathbf{x}, \boldsymbol{\theta}), t)$ and $\mu(\mathbf{x};\boldsymbol{\theta}+\delta\boldsymbol{\theta}) = \frac{\delta\mathbf{x}}{\delta t} - \mathbf{F}(\mathbf{x}, \mathbf{u}(\mathbf{x}, \boldsymbol{\theta}+\delta\boldsymbol{\theta}), t)$. Multiplication with δt will result in $\mu(\mathbf{x};\boldsymbol{\theta})\delta t = \delta\mathbf{x} - \mathbf{F}(\mathbf{x}, \mathbf{u}(\mathbf{x}, \boldsymbol{\theta}), t)\delta t = \mathbf{C}(\mathbf{x}, t)d\mathbf{w}_{\theta}(t)$ and $\mu(\mathbf{x};\boldsymbol{\theta}+\delta\boldsymbol{\theta})\delta t = \delta\mathbf{x} - \mathbf{F}(\mathbf{x}, \mathbf{u}(\mathbf{x}, \boldsymbol{\theta}+\delta\boldsymbol{\theta}), t)\delta t = \mathbf{C}(\mathbf{x}, t)d\mathbf{w}_{\theta+\delta\theta}(t)$. Also $\mu(\mathbf{x};\boldsymbol{\theta}+\delta\boldsymbol{\theta}) = \frac{\delta\mathbf{x}}{\delta t} - \mathbf{F}(\mathbf{x};\boldsymbol{\theta}) - \delta\mathbf{F} = \mu(\mathbf{x};\boldsymbol{\theta}) - \delta\mathbf{F}$ where $\delta\mathbf{F} = \mathbf{F}(\mathbf{x}, \mathbf{u}(\mathbf{x}, \boldsymbol{\theta}+\delta\boldsymbol{\theta}), t) - \mathbf{F}(\mathbf{x}, \mathbf{u}(\mathbf{x}, \boldsymbol{\theta}), t)$. Now we would like to find the expression:

$$\frac{d\mathbb{P}\Big(\mathbf{x}_N, t_N | \mathbf{x}_0, t_0; \boldsymbol{\theta}\Big)}{d\mathbb{P}\Big(\mathbf{x}_N, t_N | \mathbf{x}_0, t_0; \boldsymbol{\theta}+\delta\boldsymbol{\theta}\Big)}$$

$$= \frac{\exp\left(-\frac{1}{2}\left(\int_{t_0}^{t_N} ||\mu(\mathbf{x};\boldsymbol{\theta})||^2_{\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}} \delta t\right)\right)}{\exp\left(-\frac{1}{2}\left(\int_{t_0}^{t_N} ||\mu(\mathbf{x};\boldsymbol{\theta}+\delta\boldsymbol{\theta})||^2_{\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}} \delta t\right)\right)}$$

$$= \exp\left[-\frac{1}{2}\int_{t_0}^{t_N}\Big(||\mu(\mathbf{x};\boldsymbol{\theta})||^2_{\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}} - \mu(\mathbf{x};\boldsymbol{\theta}+\delta\boldsymbol{\theta})||^2_{\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}}\Big)\delta t\right]$$

$$= \exp\left[-\frac{1}{2}\int_{t_0}^{t_N}\Big(2\mu(\mathbf{x};\boldsymbol{\theta})^T\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}\delta\mathbf{F} - \delta\mathbf{F}^T\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}\delta\mathbf{F}\Big)\delta t\right]$$

$$= \exp\left[\int_{t_0}^{t_N}\Big(-\delta\mathbf{F}^T\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}\mathbf{C}(\mathbf{x}, t)d\mathbf{w}_{\theta}(t) + \frac{1}{2}\delta\mathbf{F}^T\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}\delta\mathbf{F}\delta t\Big)\right]$$

For the case where $\mathbf{C}(\mathbf{x}, t)$ is invertible we will have that

$$\frac{d\mathbb{P}\Big(\mathbf{x}_N, t_N | \mathbf{x}_0, t_0; \boldsymbol{\theta}\Big)}{d\mathbb{P}\Big(\mathbf{x}_N, t_N | \mathbf{x}_0, t_0; \boldsymbol{\theta}+\delta\boldsymbol{\theta}\Big)}$$

$$= \exp\left[\int_{t_0}^{t_N}\Big(-\delta\mathbf{F}^T\mathbf{C}(\mathbf{x}, t)^{-1}d\mathbf{w}_{\theta}(t) + \frac{1}{2}\delta\mathbf{F}^T\boldsymbol{\Sigma}_{\mathbf{C}}^{-1}\delta\mathbf{F}\delta t\Big)\right]$$

This is the same expression for Girsanov theorem as in [32].

References

[1] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):115–133, 1983.

[2] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal. Learning variable impedance control. *nternational journal of robotics research*, pages 820–833, April 2011.

[3] Jonas Buchli, Evangelos Theodorou, Freek Stulp, and Stefan Schaal. Variable impedance control - a reinforcement learning approach. In *Robotics: Science and Systems Conference (RSS)*, 2010.

[4] Charalambos D. Charalambous and Farzad Rezaei. Stochastic uncertain systems subject to relative entropy constraints: Induced norms and monotonicity properties of minimax games. *IEEE Trans. Automat. Contr.*, 52(4):647–663, 2007.

[5] Paolo Dai Pra, Lorenzo Meneghini, and Wolfgang Runggaldier. Connections between stochastic control and dynamic games. *Mathematics of Control, Signals, and Systems (MCSS)*, 9(4):303–326, 1996-12-08.

[6] P. Dayan and G. Hinton. Using em for reinforcement learning. *Neural Computation*, 9, 1997.

[7] Marc P. Deisenroth, Carl E. Rasmussen, and Jan Peters. Gaussian Process Dynamic Programming. *Neurocomputing*, 72(7–9):1508–1524, March 2009.

[8] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, and G. Cheng. Learning cpg sensory feedback with policy gradient for biped locomotion for a full-body humanoid. In *AAAI 2005*, 2005.

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

[10] W. H. Fleming and H. Mete Soner. *Controlled Markov processes and viscosity solutions*. Applications of mathematics. Springer, New York, 2nd edition, 2006.

[11] Emmanuel Gobet and Rmi Munos. Sensitivity analysis using it malliavin calculus and martingales. application to stochastic optimal control. 43:1676–1713, 2005.

[12] Floyd B. Hanson. *Applied Stochastic Processes and Control for Jump-Diffusions*. SIAM, 2007.

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

[14] J. Kober and J. Peters. Policy search for motor primitives. In D. Schuurmans, J. Benigio, and D. Koller, editors, *Advances in Neural Information Processing Systems 21 (NIPS 2008)*. Cambridge, MA: MIT Press, 2008.

[15] Sanjoy K. Mitter and Nigel J. Newton. A variational approach to nonlinear estimation. *SIAM J. Control Optim.*, 42(5):1813–1833, May 2003.

[16] Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural Comput.*, 17(2), 2005.

[17] J. Peters. *Machine learning of motor skills for robotics.* PhD thesis, University of Southern California, 2007.

[18] J. Peters and S. Schaal. Learning to control in operational space. *International Journal of Robotics Research*, 27:197–212, 2008.

[19] J. Peters and S. Schaal. Natural actor critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

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

[21] J. Peters, S. Vijayakumar, and S. Schaal. Natural actor-critic. In *Proceedings of the 16th european conference on machine learning (ecml 2005)*, pages 280–291. springer, 2005.

[22] M. Sato, Y. Nakamura, and S. Ishii. Reinforcement learning for biped locomotion. In *International Conference on Artificial Neural Networks (ICANN)*, Lecture Notes in Computer Science, pages 777–782. Springer-Verlag, 2002.

[23] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Processing Systems 12*, Denver, CO, 2000. MIT Press.

[24] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning : An introduction*. Adaptive computation and machine learning. MIT Press, Cambridge, 1998.

[25] Russ Tedrake, Teresa Weirui Zhang, and H. Sebastian Seung. Learning to walk in 20 minutes. In *Proceedings of the Fourteenth Yale Workshop on Adaptive and Learning Systems*, 2005.

[26] E.. Theodorou. *Iterative Path Integral Stochastic Optimal Control: Theory and Applications to Motor Control*. PhD thesis, university of southern California, May 2011.

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

[28] Emanuel Todorov. Policy gradients in linearly-solvable mdps. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *NIPS*, pages 2298–2306. Curran Associates, Inc., 2010.

[29] H. Touchette. The large deviation approach to statistical mechanics. *Physics Reports*, 478:1–69, 2009.

[30] N. Vlassis, M. Toussaint, G. Kontes, and Piperidis. S. Learning model-free control by a monte-carlo em algorithm. *Autonomous Robots*, 27(2):123–130, 2009.

[31] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[32] J. YANG and J. H. Kushner. A monte carlo method for sensitivity analysis and parametric optimization of nonlinear stochastic systems. *SIAM Journal in Control and Optimization*, 29(5):1216–1249, 1991.