Actor-only Deterministic Policy Gradient via Zeroth-order Gradient Oracles in Action Space

Harshat Kumar*, Dionysios S. Kalogerias†, George J. Pappas*, and Alejandro Ribeiro*
*Department of Electrical and Systems Engineering, University of Pennsylvania,
Philadelphia, PA 19104, USA — email: {harshat, pappasg, aribeiro}@seas.upenn.edu
†Department of Electrical and Computer Engineering, Michigan State University,
East Lansing, MI 48824, USA — email: dionysis@msu.edu

2021 IEEE International Symposium on Information Theory (ISIT) | 978-1-5386-8209-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISIT45174.2021.9518023

***Abstract*—Deterministic policies demonstrate substantial empirical success over their stochastic counterparts as they remove a level of randomness in Policy Gradient (PG) methods when applied to stochastic search problems involving Markov decision processes. However, current implementations require the use of state-action value (Q-function) approximators, also known as critics, to obtain estimates of the associated policy-reward gradient. In this work, we propose the use of two-point stochastic evaluations to obtain gradient estimates of a smoothed Q-function surrogate, constructed by evaluating pairs of the Q-function at low-dimensional, randomized initial action perturbations. This procedure lifts the dependence on a critic and restores true model-free policy learning, and with provable algorithmic stability. In fact, our finite complexity bounds improve upon existing results by up to 2 orders of magnitude in terms of iteration complexity, and by up to $3/2$ orders of magnitude in terms of sample complexity. Simulation results on an agent navigation problem showcase the effectiveness of our proposed algorithm in a practical setting, as well.**

An unpublished preprint of this paper including proofs to all results is accessible at: https://arxiv.org/pdf/2006.07314.pdf

I. Introduction

Reinforcement Learning (RL) has proven itself a well-known and effective approach for tackling dynamic optimization problems and is ubiquitous in many application areas, such as robotic manipulation [1], source enhancement [2], wireless communications [3], [4], and underwater acoustic transmission [5] to name a few. A general approach for solving RL problems is *policy search*, which relies on directly learning a policy to maximize rewards through interaction with an unknown environment. Hypothesizing that actions are chosen according to a parameterized distribution (a.k.a. a randomized policy), *Policy Gradient (PG)* is a type of policy search algorithm that repeatedly updates the policy through a specially-derived stochastic gradient [6], enabling solutions in complex dynamic problems with continuous state-action spaces. Alternatively, *Deterministic Policy Gradient (DPG)* considers RL problems by naturally assuming that the policy is a deterministic mapping from states to actions [7]. This formulation removes policy randomization, otherwise integral to classical PG, and permits the complete characterization of the gradient of the composed policy-reward function with respect to the parameters of the policy. DPG has been shown to outperform its stochastic counterparts empirically on a wide variety of RL problems [8], [9].

In this work, we introduce *Zeroth-order Deterministic Policy Gradient (ZDPG)*, which advocates the use of (zeroth-order) noisy estimates of the Q-function for approximating the corresponding (deterministic) policy-reward gradient, in the standard infinite horizon MDP formulation. In particular, by exploiting the recently introduced idea of *random horizon rollouts* for obtaining unbiased stochastic estimates of the Q-function [10], [11], we show that it is possible to efficiently construct consistent two-point approximations (single or in batches) of the associated policy-reward gradient [12], [13]; this is achieved by evaluating pairs of the Q-function at low-dimensional random action-space perturbations involving *initial actions only*. Such a construction is important from a computational perspective for two main reasons. First, for most popular choices of policy parameterizations (e.g., Deep Neural Networks (DNNs)), the dimension of the corresponding parameter space is typically much larger than the dimension of the action space, where our random perturbations are introduced [14]. This is a key feature because it is well-known that the dimension of the perturbation space has drastic negative effects on the convergence rate of zeroth-order methods [12], [13]. Second, the fact that we (randomly) perturb only the initial action of the underlying dynamic program makes our ZDPG approach readily implementable and easily applicable to the standardized infinite horizon setting.

As compared with standard PG and DPG, ZDPG enjoys the additional operational advantages. First, as with DPG, ZDPG eliminates the need for randomized policies. Although we are not the first to achieve this goal [7], we would still like to perpetuate the narrative that randomized policies, although most common, are *not* essential in PG-based RL. At the same time, while ZDPG exploits gradient information of the adopted (and known) policy parameterization, it *restores* true *model-free* policy learning (as in standard PG); this is naturally lost in DPG schemes, where the use of an auxiliary critic is typically enforced [7]. Finally, ZDPG enjoys built-in and provable quasi-gradient variance stability (albeit at the expense of an additional system rollout), which is very similar in principle to (heuristic) baseline PG techniques [15].

We also characterize the convergence rate of our proposed ZDPG algorithm under common problem regularity conditions. From a technical standpoint, our analysis is different from that of conventional zeroth-order methods for nonconvex

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:03:59 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SUMMARY OF RATE RESULTS FOR NUMBER OF ITERATIONS T AND ACCURACY ϵ (SPECIFIED BY MONTE CARLO REDUCTION N, SMOOTHNESS PARAMETER μ, AND STEPSIZE α)

	Smooth Q-function		Hessian-Lipschitz Q-function	
(N)	1	T	1	$\sqrt{T}$
(μ)	$T^{-1/4}$	$T^{-1/2}$	$T^{-1/6}$	$T^{-1/4}$
(α)	$T^{-3/4}$	$T^{-1/2}$	$T^{-2/3}$	$T^{-1/2}$
Rate	$\mathcal{O}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-2})$

optimization – which follows the general structure of defining a smoothed surrogate objective (reward), analyzing the convergence of the resulting smoothed problem, and then relating those results back to the original problem (see, e.g., [16], [17]). Our rate results are summarized in Table I, where we compare different smoothness assumptions with and without Monte Carlo variance reduction.

Lastly, the performance of ZDPG is empirically evaluated on a point agent navigation problem in the presence of environment obstacles, where there is no knowledge of the system dynamics. Without the use of any heuristics, our numerical results demonstrate that ZDPG substantially outperforms both standard PG and PG with a baseline under both noiseless and noisy system dynamics and in terms of both mean performance and variance. In particular, our simulations elucidate the fact that ZDPG enjoys clear operational advantages over PG with a baseline, which may be considered as the analogous two-point update for the stochastic policy setting.

II. DETERMINISTIC POLICY GRADIENT

We consider a Reinforcement Learning (RL) problem where an agent moves through a continuous and compact state space $\mathcal{S} \subset \mathbb{R}^q$, and takes actions in a finite dimensional action space $\mathcal{A} = \mathbb{R}^p$. The agent's task is to accumulate as much reward as possible in the long term, where the reward is revealed to the agent by the environment at each step. This generic problem may be abstracted by a Markov decision process (MDP) as a tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $R : \mathcal{S} \times \mathbb{R}^p \to \mathbb{R}$ is the reward function, and $P^a_{s\to s'} := p(s'|(s,a) \in \mathcal{S} \times \mathcal{A})$ determines the probability of moving to state s' starting from state s and action a, satisfying the Markov property $p(s_{t+1}|(s_u, a_u) \in \mathcal{S} \times \mathcal{A}, \forall u \le t) = p(s_{t+1}|(s_t, a_t) \in \mathcal{S} \times \mathcal{A})$. The value $\gamma \in (0,1)$ is the discount factor which determines how much future rewards matter to the behavior of the agent. Lastly, the *deterministic* policy $\pi : \mathcal{S} \to \mathcal{A}$ defines a mapping from a state to an action. Then, the problem of interest is to choose a policy that maximizes the expected discounted sum of rewards over all starting states; this means finding a policy that maximizes the expected value function $V : \mathcal{S} \to \mathbb{R}$ starting from state s, i.e.,

$$\max_{\pi} \mathbb{E}_{s\sim\rho^0}\left[V(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t = \pi(s_t)) \middle| s\right]\right], \quad (1)$$

where ρ^0 is the initial state distribution. By conditioning the value function on an action, we can define the state-action function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ by

$$Q^\pi(s,a) := R(s,a) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,a)}\left[V(s')\right] \quad (2)$$

The Q-function determines the quality of taking an action a at state s and then following policy π for the remainder of time. By selecting the first action according to the policy, the Q-function is equivalent to the value function, and hence problem (1) is equivalently written as

$$\max_{\pi} \mathbb{E}_{s\sim\rho^0}\left[Q(s, \pi(s))\right]. \quad (3)$$

Solving (3) requires searching over a function space, which is generally an intractable problem. To circumvent this complexity, the policy is parameterized by some $\theta \in \Theta \subset \mathbb{R}^d$ so that the search is over a Euclidean space of finite dimension. The problem of interest consequently becomes

$$\max_{\theta} \mathbb{E}_{s\sim\rho^0}\left[Q^{\pi_\theta}(s, \pi_\theta(s))\right] =: J(\theta). \quad (4)$$

In their seminal work [7], Silver *et al.* proved that, under mild problem regularity conditions, the gradient of (4) can be expressed as

$$\nabla J(\theta) = \mathbb{E}_{s\sim\rho^{\pi_\theta}}\left[\nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s,a)|_{a=\pi_\theta(s)}\right], \quad (5)$$

where ρ^{π_θ} constitutes the so-called the (improper) *discounted state distribution* under control policy π_θ, defined as $\rho^{\pi_\theta}(s') = \int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t p_0(s) p(s \to s', t, \theta) ds$, and $p(s \to s', t, \theta)$ denotes the density at state s' after starting from state s and transitioning with t time steps given policy parameterized by θ. Next, we formally define some standard assumptions on the policy gradient, reward function, and Q-function, as follows.

Assumption 1: (Bounded Policy Gradient) The gradient $\nabla_\theta \pi_\theta(s)$ is uniformly bounded on $\Theta \times \mathcal{S}$, that is, there exists $B_\Theta < \infty$, such that $\sup_{(\theta,s)\in\Theta\times\mathcal{S}} \|\nabla_\theta \pi_\theta(s)\| \le B_\Theta$.

Assumption 2: (Bounded Rewards) The reward R is uniformly bounded on $\mathcal{S} \times \mathcal{A}$, that is, there exists $U_R < \infty$, such that $\sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} |R(s,a)| \le U_R$.

By Assumption 2, it also follows that the Q-function is also uniformly bounded as

$$|Q^{\pi_\theta}(s,a)| \le \sum_{t=0}^{\infty} \gamma^t \cdot U_R = \frac{1}{1-\gamma} U_R =: \mathcal{Q}.$$

Both Assumptions 1 and 2 are standard for deriving performance guarantees in the stochastic policy search literature [11], [18], [19], and are easily satisfied by common choices of the deterministic policy and reward functions. Additionally, we require sufficient smoothness of the Q-function, as follows.

Assumption 3: (Lipschitzness of the Q-function) The Q-function is action-Lipschitz uniformly on $\Theta \times \mathcal{S}$. That is, there is a number $L < \infty$, such that, for every $a_1, a_2 \in \mathcal{A}$,

$$\sup_{(\theta,s)\in\Theta\times\mathcal{S}} |Q^{\pi_\theta}(s,a_1) - Q^{\pi_\theta}(s,a_2)| \le L\|a_1 - a_2\|.$$

Assumption 4: (Smoothness of the Q-function) The Q-function is action-smooth uniformly on $\Theta \times \mathcal{S}$. That is, there is a number $G < \infty$, such that, for every $a_1, a_2 \in \mathcal{A}$,

$$\sup_{(\theta,s)\in\Theta\times\mathcal{S}} \|\nabla_a Q^{\pi_\theta}(s,a_1) - \nabla_a Q^{\pi_\theta}(s,a_2)\| \le G\|a_1 - a_2\|.$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:03:59 UTC from IEEE Xplore. Restrictions apply.

Intuitively, Assumptions 3 and 4 mean that similar actions from the same state should have similar values. Those principles have informed Actor-Critic algorithms in practice as they are known to avoid overfitting to narrow peaks in the value estimate [20]. Further, existing analysis for zeroth-order action space exploration in *finite* horizon exploits these assumptions as well [14]. Lastly, in some of our results we assume that the problem is well-conditioned with Lipschitz Hessians.

Assumption 5: (Smoothness of ∇Q) The gradient field ∇Q is action-smooth uniformly on $\Theta \times \mathcal{S}$. That is, there is a number $H < \infty$, such that, for every $a_1, a_2 \in \mathcal{A}$,

$$\sup_{(\theta,s)\in\Theta\times\mathcal{S}} \|\nabla_a^2 Q^{\pi_\theta}(s,a_1) - \nabla_a^2 Q^{\pi_\theta}(s,a_2)\| \leq H\|a_1 - a_2\|.$$

Assumption 5, though not entirely standard, holds for a variety of problems with well behaved reward functions [21] and policy representations, such as the Linear Quadratic Regulator (LQR) [22], [23] and its many (and potentially nonlinear) variations. In fact, Assumption 5 will be satisfied in problems where the Q-function is at least twice differentiable and the action space is compact. Also note that Assumption 5 will be satisfied locally for any reasonable problem which is locally quadratic around a locally optimal solution, at least asymptotically. The trade-off between our various smoothness assumptions and the convergence rates achieved in this work is demonstrated in Table I. In the following section, we review how to use zeroth-order information to evaluate stochastic gradient estimates.

III. Zeroth-order Gradient Estimates and ZDPG

First, let us introduce a smoothed approximation to the Q-function by exploiting low-dimensional perturbations with respect to the action a. Let $\mu > 0$ be *the smoothing parameter*, and let $\mathbf{u} \sim \mathcal{N}(0, I_p)$. Then, the μ-smoothed Q-function is defined as

$$Q_\mu^{\pi_\theta}(s,a) := \mathbb{E}_{\mathbf{u}}\left[Q^{\pi_\theta}(s, a+\mu\mathbf{u})\right].$$

By introducing smoothing, we are able to sample an unbiased stochastic estimate of $\nabla_a Q_\mu^{\pi_\theta}(s,a)$ via a two-point evaluation of the original Q-function. Namely, we recall the following key property (in terms of the Q-function), true in general for every globally Lipschitz function as originally presented in [13], then extended to a wider class of functions in [16].

Lemma 1: [16, Lemma 2] For every $\mu > 0$, the μ-smoothed Q-function surrogate $Q_\mu^{\pi_\theta}$ is differentiable, and its gradient admits the representations

$$\begin{aligned}\nabla_a Q_\mu^{\pi_\theta}(s,a) &\equiv \mathbb{E}_{\mathbf{u}}\left[\frac{Q^{\pi_\theta}(s,a+\mu\mathbf{u}) - Q^{\pi_\theta}(s,a)}{\mu}\mathbf{u}\right] \qquad (6)\\ &\equiv \mathbb{E}_{\mathbf{u}}\left[\frac{Q^{\pi_\theta}(s,a+\mu\mathbf{u}) - Q^{\pi_\theta}(s,a-\mu\mathbf{u})}{2\mu}\mathbf{u}\right].\end{aligned}$$

By exploiting Lemma 1, we propose the following estimate of the gradient of the objective defined in (5) where, instead of the true Q-function gradient $\nabla_a Q^{\pi_\theta}(s,a)$ we use the smoothed gradient $\nabla_a Q_\mu^{\pi_\theta}(s,a)$ to obtain the *quasi-gradient*

$$\hat{\nabla} J(\theta) = \mathbb{E}_{s\sim\rho^{\pi_\theta}}\left[\nabla_\theta \pi_\theta(s) \nabla_a Q_\mu^{\pi_\theta}(s,a)|_{a=\pi_\theta(s)}\right]. \qquad (7)$$

Algorithm 1 Zeroth-order Deterministic Policy Gradient

Require: θ_0, γ, μ, α, N
  for $t = 0, 1, \ldots$ **do**
    Sample $T_Q \sim$ Geom$(1-\gamma)$
    Sample $s_t \sim (1-\gamma)\rho^{\pi_{\theta_t}}$ [Algorithm3(T_Q, θ_t)]
    Sample $\mathbf{u} \sim \mathcal{N}(0, I_p)$
    Initialize Estimates $\hat{Q}_t^+ \leftarrow 0$, $\hat{Q}_t^- \leftarrow 0$
    for $n = 0, 1, \ldots, N-1$ **do**
      $\hat{Q}_{t,n}^+ \leftarrow$ Algorithm2$(T_Q, s_t, \pi_{\theta_t}(s_t) + \mu\mathbf{u}, \theta_t)$
      $\hat{Q}_{t,n}^- \leftarrow$ Algorithm2$(T_Q, s_t, \pi_{\theta_t}(s_t), \theta_t)$
      $\hat{Q}_t^+ \leftarrow \hat{Q}_t^+ + \frac{1}{N}\hat{Q}_{t,n}^+$
      $\hat{Q}_t^- \leftarrow \hat{Q}_t^- + \frac{1}{N}\hat{Q}_{t,n}^-$
    end for
    Evaluate $\Psi_t = \nabla_\theta \pi_{\theta_t}(s_t)$
    $g_t \leftarrow \frac{1}{1-\gamma}\Psi_t\left(\frac{\hat{Q}_t^+ - \hat{Q}_t^-}{\mu}\right)\mathbf{u}$
    $\theta_{t+1} \leftarrow \theta_t + \alpha g_t$
  end for

Lemma 1 provides a procedure to obtain an unbiased estimate of $\nabla_a Q_\mu^{\pi_\theta}(s,a)$, which is to sample $\mathbf{u} \sim \mathcal{N}(0, I_p)$ and then evaluate the Q-function with *initial* actions a and $a + \mu\mathbf{u}$. By selecting the smoothing parameter μ sufficiently small, the gradient estimate becomes close to the deterministic policy gradient. Note that, while obtaining two samples of the Q-function at a specific state may not be practical on a real system, this can be made feasible through the use of simulators which allow resetting and reproduction of the same stochastic environment; see, e.g., [24]. Further, it is possible for function approximators to be used to make the evaluation of the action perturbations trivial. We also point out that this is precisely what is done in [7]. Also note that ZDPG can be implemented using an one-sided gradient approximation (i.e., by removing $\hat{Q}^-$ altogether from Algorithm 1). This would lift the need for a simulator, but would affect gradient approximation stability. This is fully analogous to standard REINFORCE and baseline REINFORCE: the second does require a simulator or function approximator, while the first does not.

A key advantage of ZDPG is that it is a *primitive* RL method such as (baseline-)REINFORCE, which systematically enables model-free RL without need for a randomized policy; the policy herein is a deterministic function (e.g., a deep neural network, or a kernel machine). This has important advantages from an operational point of view, because ZDPG does not inject unnecessary artificial exploration noise during policy training, which would result in high variability during both training and implementation (which happens in the case of randomized policies –see Section IV). The resulting stochastic search scheme using two-point zeroth-order gradient estimates is presented in Algorithm 1, where α is the policy parameter stepsize. We employ Monte-Carlo variance reduction by running the system $N \geq 1$ times ($N = 1$ implies no reduction).

Algorithm 1 assumes that *both* an unbiased estimate of the Q-function evaluated at any state-action pair *and* sampling from the discounted state distribution ρ^{π_θ} of our original,

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:03:59 UTC from IEEE Xplore. Restrictions apply.

infinite horizon problem are feasible. Here, we describe both procedures which involve *sampling* a random but *finite* horizon length from a geometric distribution. Indeed, it is known that the procedure described by Algorithm 2 (which is called by Algorithm 1; also see next page) results in unbiased estimates of the Q-function defined by (2) [10, Proposition 2]. As the next result asserts, the aforementioned estimates are of finite variance, as well.

Lemma 2: Let Assumption 2 be in effect. Then, the estimate $\hat{Q}$ produced by Algorithm 2 is of bounded variance, independent of the inputs of Algorithm 2.

For later reference, let $w_{t,n}^{\pm} := \hat{Q}_{t,n}^{\pm} - Q^{\pi_{\theta_t}}(\pm)$, where "$\pm$" refers to the respective initial state-action pair (see Algorithm 1). Also introduced in [10], Algorithm 3 (also called by Algorithm 1) describes a similar procedure to obtain samples $s \sim (1-\gamma)\rho^{\pi_\theta}$, where $(1-\gamma)\rho^{\pi_\theta}$ is the *proper* (i.e., scaled to sum to unity) version of the discounted state distribution ρ^{π_θ}.

We are now ready to state our main theorem, which rigorously establishes the convergence rate of Algorithm 1 with respect to the *true* gradient of the original objective $J(\theta)$.

Theorem 1: Let Assumptions 1-3 be in effect, and suppose that the function $J(\theta)$ is L_J-smooth and G_J-Lipschitz continuous. Under Assumption 4 and specific choices of smoothing paramter μ and stepsize α (see Table I), it is true that

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right] \leq \begin{cases} \mathcal{O}\left(p^2 T^{-1/4}\right), & \text{if } N=1 \\ \mathcal{O}\left(p^2 T^{-1/2}\right), & \text{if } N=T \end{cases}.$$

Alternatively, under Assumption 5, it holds that

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right] \leq \begin{cases} \mathcal{O}\left(p^2 T^{-1/3}\right), & \text{if } N=1 \\ \mathcal{O}\left(p^2 T^{-1/2}\right), & \text{if } N=T^{1/2} \end{cases}.$$

Theorem 1 provides iteration complexity bounds for ZDPG (i.e., relative to T). Compared with [14], ZDPG gains up to two orders of magnitude in iteration complexity.

With respect to the sample complexity, for Lipchitz Hessians of the Q-function, improvement is of at least one full order: namely, with the same number of samples, ZDPG obtains an order of (iteration) improvement with no Monte-Carlo variance reduction (i.e., when $N=1$ rollouts are used); further for an additional overhead of $N=\sqrt{T}$ rollouts, ZDPG gains another order of (iteration) improvement (this amounts to a total of $3/2 = 1+1/2$ orders of improvement in sample complexity).

In the case of a smooth Q-function (Lipschitz gradients), while the order of *sample* complexity is not improved as compared with [14] (in contrast to *iteration* complexity), the incorporation of multiple rollouts is *not* equivalent to squaring the number of outer iterations, T. This is because the use of multiple rollouts allows for significant gradient variance reduction (by construction). In fact, from Table I, if we exclude the choice of stepsize α (this is reasonable since in practice α is chosen as constant –see also our experiments), even in the case of a merely smooth Q-function, the order of the gradient variance is growing as $\mathcal{O}((\mu^2 N)^{-1})$ [25], which reduces to $\mathcal{O}(\sqrt{T})$ when $N=1$ (no MC reduction), and improves to a constant $\mathcal{O}(1)$ when $N=T$, which also corresponds to an improvement of two orders of magnitude. Same is true for the case of Lipschitz Hessians.

IV. Numerical Results

We consider a two-dimensional continuous state-action navigation problem where a point agent aims to avoid a spherical (for simplicity) obstacle and converge quickly to some fixed target while satisfying its motion control dynamics. This goal is achieved by appropriately shaping the associated reward function. Let $s^* \in \mathbb{R}^2$ represent the target that the point agent wishes to reach. Let $s^c \in \mathbb{R}^2$ and $r \in \mathbb{R}_+$ denote the center and radius of the spherical obstacle the agent wishes to avoid. Then, we define the reward $R : \mathbb{R}^2 \to \mathbb{R}$ as a combination of two artificial potential functions: an attractive potential $\phi^{att} : \mathbb{R}^2 \to \mathbb{R}_+$ with a maximum at the target, and a repulsive artificial potential $\phi^{rep} : \mathbb{R}^2 \to \mathbb{R}_+$ with a minimum at the center of the object. Specifically, we set the attractive potential to be the negative Euclidean distance to the goal location s^* squared, that is, $\phi^{att}(s) := -\|s-s^*\|^2$, where s is the location of the agent. We let $\nu(s) := 1/2\left(1-\text{sign}\left(\|s-s^c\|-r\right)\right)$. Similar to [26], we define the repulsive potential as $\phi^{rep}(s) := 1-1/\beta(s)$, where

$$\beta(s) := \left(1-\left(\frac{1+r^4}{r^4}\right)\cdot\frac{\left(\|s-s^c\|^2-r^2\right)^2}{1+\left(\|s-s^c\|^2-r^2\right)^2}\right)^{\nu(s)}.$$

Completely characterized by the state of the agent, the reward is equal to the sum of these attractive and repulsive potentials, that is, $R(s) = \phi^{att}(s) + \phi^{rep}(s)$. Also, the motion dynamics of the point agent are described by the controlled random walk

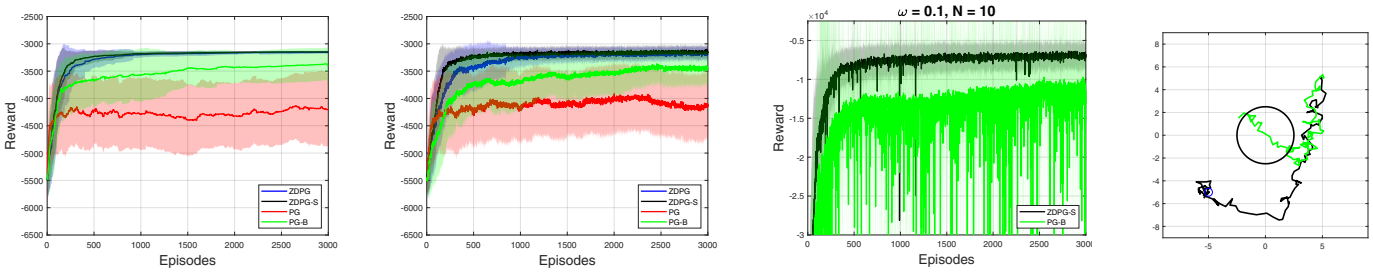
$$s_{t+1} = s_t + \eta\frac{\pi_\theta(s_t)}{\|\pi_\theta(s_t)\|} + n_t, \tag{8}$$

where $\{n_t\}$ is a random process modeling state process noise, and $\eta > 0$ (together with policy normalization) imposes implicit resource constraints on the agent motion. Here, we assume that, except from the agent control vector (second term on the right-hand side of (8)), the dynamics of the agent are *unknown*, an assumption which conforms with the standard policy gradient setting. Lastly, we consider a deterministic policy to represent optimal directions that the point agent will follow given its location to its prescribed destination. In particular, we adopt a Radial-Basis-Function (RBF) expansion as our (universal) policy parameterization, i.e.,

$$\pi_\theta(s) = \sum_{i=1}^{d}\theta_i\kappa(s,\bar{s}_i), \tag{9}$$

where $\kappa(s,s') = \exp(-\|s-s'\|_2^2/2\sigma_\kappa^2)$ defines the RBF kernel, $\theta_i \in \mathbb{R}^2, i=1,\ldots,d$ are the parameter vectors to be learned, σ_κ^2 determines the variance of the kernel, and $\bar{s}_i, i=1,\ldots,d$ are the associated RBF centers. For all simulations, we select the target of the agent to be located at $s^* = (-5,-5)$, and the obstacle to be centered at $s^c = (0,0)$ with a radius of $r = 2.5$. We define the potential with a slightly larger radius ($r = 3$) to ensure that the agent does not collide with the boundary of the obstacle. We select the centers of the RBF

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:03:59 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c) (d)

Fig. 1. Average reward per episode with confidence bounds over 50 trials without MC variance reduction. (a) Noiseless case ($\omega = 0$) with no MC variance reduction ($N = 1$) with trajectory $T = 20$ (b) Noisy Case ($\omega = 0.01$) with no MC variance reduction ($N = 1$) with trajectory $T = 20$ (c) Noisy case ($\omega = 0.1$) with MC variance reduction ($N = 10$) with trajectory $T = 100$ comparing only ZDPG-S and PG-B (d) Sample trajectory with fixed length $T = 100$ of learned policies corresponding (c). We refer the reader to [25] for extended numerical results.

Algorithm 2 Q-function Sampler [10]

Require: $\hat{Q} \leftarrow 0$, T, s_0, a_0, θ
  for $t = 0, 1, \ldots, T-1$ **do**
    Collect Reward $\hat{Q} \leftarrow \hat{Q} + R(s_t, a_t)$
    Advance System $s_{t+1} \sim \mathbb{P}(s'|s_t, a_t)$
    Select Action $a_{t+1} \leftarrow \pi_\theta(s_{t+1})$
  end for
  Collect Reward $\hat{Q} \leftarrow \hat{Q} + R(s_T, a_T)$

Algorithm 3 Discounted State Sampler [10]

Require: T, θ
  Sample $s_0 \sim \rho^0$
  for $t = 0, 1, \ldots, T-1$ **do**
    Select Action $a_t \leftarrow \pi_\theta(s_t)$
    Advance System $s_{t+1} \sim \mathbb{P}(s'|s_t, a_t)$
  end for
  Return s_T

kernels to be spaced 0.25 units from each other populating the $[-10, 10] \times [-10, 10]$ grid. The initial state distribution is uniform on $[-10, 10] \times [-10, 10]$.

We compare the performance of ZDPG and its symmetric counterpart ZDPG-S (based on the *symmetric* expression in (6)), compared with PG and PG with a *stochastic* baseline (PG-B), with discount factor $\gamma = 0.8$ in the case with and without system noise ($n_t = \mathbf{0}$ and $n_t \sim \mathcal{N}(\mathbf{0}, \omega \cdot I_2)$, respectively). For PG and PG-B, we let $\tilde{\pi}$ denote the stochastic policy where the action is selected by $a_t \sim \tilde{\pi}_\theta(s_t)$. In particular, we let $\tilde{\pi}$ be a Gaussian policy with mean equal to $\pi_\theta(s)$ [c.f. (9)] and a standard choice of covariance matrix $\Sigma = \sigma_a^2 \cdot I_2$. For PG-B, the baseline is fairly chosen and is analogous as compared to ZDPG in that it scales the update direction $\nabla_\theta \log \tilde{\pi}_\theta(s)$ by a two-point difference $(\hat{Q}(s, a) - \tilde{V}(s))$, where $\tilde{V}(s)$ is a single-rollout estimate of the associated value function at state s. After each policy update, we evaluate the system by generating a rollout of fixed length T from a fixed starting point $s^0 = (5, 5)$. As often done in practice (as a heuristic), we evaluate the stochastic policies by setting the covariance matrix equal to zero thereby making the policy deterministic.

As shown in Fig. 1 (a)-(c), regardless of the level of the system noise, ZDPG(-S) consistently outperforms PG-B (and also PG), both in mean performance and the associated variance. In fact, we observe that, *either* with *or* without system noise, the perturbation of the initial state with a deterministic policy indeed provides more meaningful information for policy improvement than in the setting of a stochastic policy (also see Fig. 1 (d)). Our empirical results corroborate our previous claim that inherent randomness in the policy is *not* necessary to solve MDPs within the PG-based RL framework.

V. Conclusion

We have proposed the ZDPG algorithm, which incorporates simple two-point evaluations of the associated Q-function to obtain efficient approximations of the corresponding deterministic policy gradient, within an infinite horizon setting. This has been made possible by exploiting random horizon rollouts to obtain unbiased estimates of the Q-function which, in conjunction with low-dimensional action-space random perturbations, lifted the dependence on critics, perpetuating the narrative that deterministic policies are superior to stochastic policies in model-free, actor-only PG methods. Our results were corroborated on an agent navigation problem, and showed substantial improvement over vanilla stochastic policy-based PG, as well as PG with a baseline. Future work includes incorporating zeroth-order gradient approximations into the design of full-blown, truly model-free actor-critic type methods, as well as identifying deeper intrinsic connections between PG methods and zeroth-order stochastic optimization.

References

[1] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[2] Y. Koizumi, K. Niwa, Y. Hioka, K. Kobayashi, and Y. Haneda, "Dnn-based source enhancement self-optimized by reinforcement learning using sound quality measurements," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 81–85.

[3] M. Eisen, C. Zhang, L. F. Chamon, D. D. Lee, and A. Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2775–2790, 2019.

[4] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2087–2091.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:03:59 UTC from IEEE Xplore. Restrictions apply.

[5] C. Wang, Z. Wang, W. Sun, and D. R. Fuhrmann, "Reinforcement learning-based adaptive transmission in time-varying underwater acoustic channels," *IEEE Access*, vol. 6, pp. 2541–2558, 2017.
[6] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.
[7] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14. JMLR.org, 2014, p. I–387–I–395.
[8] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
[9] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.
[10] S. Paternain, J. A. Bazerque, A. Small, and A. Ribeiro, "Stochastic policy gradient ascent in reproducing kernel hilbert spaces," *arXiv preprint arXiv:1807.11274*, 2018.
[11] K. Zhang, A. Koppel, H. Zhu, and T. Başar, "Global convergence of policy gradient methods to (almost) locally optimal policies," *arXiv preprint arXiv:1906.08383*, 2019.
[12] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2788–2806, 2015.
[13] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, 2017.
[14] A. Vemula, W. Sun, and J. A. Bagnell, "Contrasting exploration in parameter and action space: A zeroth-order optimization perspective," *arXiv preprint arXiv:1901.11503*, 2019.
[15] E. Greensmith, P. L. Bartlett, and J. Baxter, "Variance reduction techniques for gradient estimates in reinforcement learning," *Journal of Machine Learning Research*, vol. 5, no. Nov, pp. 1471–1530, 2004.
[16] D. S. Kalogerias and W. B. Powell, "Zeroth-order stochastic compositional algorithms for risk-aware learning," *arXiv preprint arXiv:1912.09484*, 2019.
[17] D. S. Kalogerias, M. Eisen, G. J. Pappas, and A. Ribeiro, "Model-free learning of optimal ergodic policies in wireless systems," *arXiv preprint arXiv:1911.03988*, 2019.
[18] H. Kumar, A. Koppel, and A. Ribeiro, "On the sample complexity of actor-critic method for reinforcement learning with function approximation," *arXiv preprint arXiv:1910.08412*, 2019.
[19] J. Bhandari, D. Russo, and R. Singal, "A finite time analysis of temporal difference learning with linear function approximation," *arXiv preprint arXiv:1806.02450*, 2018.
[20] S. Fujimoto, H. Van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," *arXiv preprint arXiv:1802.09477*, 2018.
[21] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *ICML*, vol. 99, 1999, pp. 278–287.
[22] M. Fazel, R. Ge, S. M. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," *arXiv preprint arXiv:1801.05039*, 2018.
[23] H. Mohammadi, A. Zare, M. Soltanolkotabi, and M. R. Jovanović, "Convergence and sample complexity of gradient methods for the model-free linear quadratic regulator problem," *arXiv preprint arXiv:1912.11899*, 2019.
[24] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, 2015, pp. 1889–1897.
[25] H. Kumar, D. S. Kalogerias, G. J. Pappas, and A. Ribeiro, "Zeroth-order deterministic policy gradient," *arXiv preprint arXiv:2006.07314*, 2020.
[26] D. S. Kalogerias and A. P. Petropulu, "Mobi-cliques for improving ergodic secrecy in fading wiretap channels under power constraints," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1578–1591.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 15:03:59 UTC from IEEE Xplore. Restrictions apply.