Adaptive Step-Size for Policy Gradient Methods

Authored by:

Marcello Restelli
Matteo Pirotta
Luca Bascetta

Abstract

In the last decade, policy gradient methods have significantly grown in popularity in the reinforcement–learning field. In particular, they have been largely employed in motor control and robotic applications, thanks to their ability to cope with continuous state and action domains and partial observable problems. Policy gradient researches have been mainly focused on the identification of effective gradient directions and the proposal of efficient estimation algorithms. Nonetheless, the performance of policy gradient methods is determined not only by the gradient direction, since convergence properties are strongly influenced by the choice of the step size: small values imply slow convergence rate, while large values may lead to oscillations or even divergence of the policy parameters. Step–size value is usually chosen by hand tuning and still little attention has been paid to its automatic selection. In this paper, we propose to determine the learning rate by maximizing a lower bound to the expected performance gain. Focusing on Gaussian policies, we derive a lower bound that is second–order polynomial of the step size, and we show how a simplified version of such lower bound can be maximized when the gradient is estimated from trajectory samples. The properties of the proposed approach are empirically evaluated in a linear–quadratic regulator problem.

1 Paper Body

Policy gradient methods have established as the most effective reinforcement?learning techniques in robotic applications. Such methods perform a policy search to maximize the expected return of a policy in a parameterized policy class. The reasons for their success are many. Compared to several traditional reinforcement?learning approaches, policy gradients scale well to high?dimensional continuous state and action problems, and no changes to the algorithms are needed to face uncertainty in the state due to limited and noisy sensors. Furthermore, policy representation can be properly designed for the given task, thus allowing to incorporate domain knowledge into the algorithm useful to speed up the learning process and to prevent the unexpected execution of dangerous policies

that may harm the system. Finally, they are guaranteed to converge to locally optimal policies. Thanks to these advantages, from the 1990s policy gradient methods have been widely used to learn complex control tasks [1]. The research in these years has focused on obtaining good model?free estimators of the policy gradient using data generated during the task execution. The oldest policy gradient approaches are finite?difference methods [2], that estimate gradient direction by resolving a regression problem based on the performance evaluation of policies associated to different small perturbations of the current parameterization. Finite?difference methods have some advantages: they are easy to implement, do not need assumptions on the differentiability of the policy w.r.t. the policy parameters, and are efficient in deterministic settings. On the other hand, when used on real systems, the choice of parameter perturbations may be difficult and critical for system safeness. Furthermore, the presence of uncertainties may significantly slow down the convergence rate. Such drawbacks have been overcome by likelihood ratio methods [3, 4, 5], since they do not need to generate policy parameters variations and quickly converge even in highly stochastic systems. Several 1

studies have addressed the problem to find minimum variance estimators by the computation of optimal baselines [6]. To further improve the efficiency of policy gradient methods, natural gradient approaches (where the steepest ascent is computed w.r.t. the Fisher information metric) have been considered [7, 8]. Natural gradients still converge to locally optimal policies, are independent from the policy parameterization, need less data to attain good gradient estimate, and are less affected by plateaus. Once an accurate estimate of the gradient direction is obtained, policy parameters are updated by: ? t+1 = ? t + ?t ?? J ?=?t , where ?t ? R+ is the step size in the direction of the gradient. Although, given an unbiased gradient estimate, convergence to a local optimum can be guaranteed under mild conditions over the learning?rate values [9], their choice may significantly affect the convergence speed or the behavior during the transient. Updating the policy with large step sizes may lead to policy oscillations or even divergence [10], while trying to avoid such phenomena by using small learning rates determines a growth in the number of iterations that is unbearable in most real?world applications. In general unconstrained programming, the optimal step size for gradient ascent methods is determined through line?search algorithms [11], that require to try different values for the learning rate and evaluate the function value in the corresponding updated points. Such an approach is unfeasible for policy gradient methods, since it would require to perform a large number of policy evaluations. Despite these difficulties, up to now, little attention has been paid to the study of step? size computation for policy gradient algorithms. Nonetheless, some policy search methods based on expectation?maximization have been recently proposed; such methods have properties similar to the ones of policy gradients, but the policy update does not require to tune the step size [12, 13]. In this paper, we propose a new approach to compute the step size in policy gradient methods that guarantees an improvement at each step, thus avoiding oscillation and divergence issues. Starting from a lower bound to the difference of performance between two poli-

cies, in Section 3 we derive a lower bound in the case where the new policy is obtained from the old one by changing its parameters along the gradient direction. Such a new bound is a (polynomial) function of the step size, that, for positive values of the step size, presents a single, positive maximum (i.e., it guarantees improvement) which can be computed in closed form. In Section 4, we show how the bound simplifies to a quadratic function of the step size when Gaussian policies are considered, and Section 5 studies how the bound needs to be changed in approximated settings (e.g., model?free case) where the policy gradient needs to be estimated directly from experience.

2

Preliminaries

A discrete?time continuous Markov decision process (MDP) is defined as a 6-tuple hS, A, P, R, ?, Di, where S is the continuous state space, A is the continuous action space, P is a Markovian transition model where P(s0 —s, a) defines the transition density between state s and s0 under action a, R : S ? A ? [0, R] is the reward function, such that R(s, a) is the expected immediate reward for the state-action pair (s, a) and R is the maximum reward value, ? ? [0, 1) is the discount factor for future rewards, and D is the initial state distribution. The policy of an agent is characterized by a density distribution ?(?—s) that specifies for each state s the density distribution over the action space A. To measure the distance between two policies we will use this norm: Z 0 k? ? ?k? = sup —? 0 (a—s) ? ?(a—s)—da, s?S

A

that is the superior value over the state space of the total variation between the distributions over the action space of policy ? 0 and ?. We consider infinite horizon problems where the future rewards are exponentially discounted with ?. For each state s, we define the utility of following a stationary policy ? as: "? # X ? t V (s) = E at ? ? ? R(st , at)—s0 = s . st ? P

It is known that V

?

t=0

solves the following recursive (Bellman) equation: Z Z V ? (s) = ?(a—s)R(s, a) + ? P (s0 —s, a)V ? (s0)ds0 da. A

S

2

Policies can be ranked by their expected discounted reward starting from the state distribution D: Z Z Z ? ? ? JD = D(s)V (s)ds) = dD (s) ?(a—s)R(s, a)dads, S

d?D (s)

P?

S

A

t

where = (1 ? ?) t=0 ? P r(st = s—?, D) is the ??discounted future state distribution for a starting state distribution D [5]. Solving an MDP means to find a policy ? ? that maximizes ? the expected long-term reward: ? ? ?

arg max??? JD . For any MDP there exists at least one deterministic optimal policy that simultaneously maximizes V ? (s), ?s ? S. For control purposes, it is better to consider action values Q? (s, a), i.e., the value of taking action a in state s and following a policy ? thereafter: Z Z ? 0 Q (s, a) = R(s, a) + ? P(s —s, a) ?(a0 —s0)Q? (s0 , a0)da0 ds0 . S

A

Furthermore, we define the advantage function: A? (s, a) = Q? (s, a) ? V ? (s), that quantifies the advantage (or disadvantage) of taking action a in state s instead of following policy ?. In for each state s, we define the advantage of a policy ? 0 over policy ? as R particular, ?0 0 ? A? (s) = A ? (a—s)A (s, a)da and, following [14], we define its expected value w.r.t. an initial R 0 0 state distribution ? as A??,? = S d?? (s)A?? (s)ds. We consider the problem of finding a policy that maximizes the expected discounted reward over a class of parameterized policies ?? = {?? : ? ? Rm }, where ?? is a compact representation of ?(a—s, ?). The exact gradient of the expected discounted reward w.r.t. the policy parameters [5] is: Z Z 1 d??? (s) ?? ?(a—s, ?)Q?? (s, a)dads. ?? J? (?) = 1?? S A The policy parameters can be updated by following the direction of the gradient of the expected discounted reward: ? 0 = ? + ??? J? (?). In the following, we will denote with k?? J? (?)k1 and k?? J? (?)k2 the L1? and L2?norm of the policy gradient vector, respectively.

3

Policy Gradient Formulation

In this section we provide a lower bound to the improvement obtained by updating the policy parameters along the gradient direction as a function of the step size. The idea is to start from the general lower bound on the performance difference between any pair of policies introduced in [15] and specialize it to the policy gradient framework. Lemma 3.1 (Continuous MDP version of Corollary 3.6 in [15]). For any pair of stationary policies corresponding to parameters ? and ? 0 and for any starting state distribution ?, the difference between the performance of policy ??0 and policy ?? can be bounded as follows Z 1 ? 2 0 (1) J? (?) ? J? (?) ? d??? (s)A????0 (s)ds ? k??0 ? ?? k? kQ?? k? , 1?? S 2(1 ? ?)2 where kQ?? k? is the supremum norm of the Q?function: kQ?? k? =

sup

Q?? (s, a)

s?S,a?A

As we can notice from the above bound, to maximize the performance improvement, we need to ? find a new policy ??0 that is associated to large average advantage A???0,? , but, at the same time, is not too different from the current policy ?? . Policy gradient approaches provide search directions characterized by increasing advantage values and, through the step size value, allow to control the difference between the new policy and the target one. Exploiting a lower bound to the first order Taylor?s expansion, we can bound the difference between the current policy and the new policy, whose parameters are adjusted along the gradient direction, as a function of the step size ?. Lemma 3.2. Let the update of the policy parameters be ? 0 = ? + ??? J? (?). Then 0

T

?(a—s, ?) ? ?(a—s, ?) ???? ?(a—s, ?) ?? J? (?) + ?
where ?? = ??? J? (?). 3
2
inf c?(0,1)
!
m X ? 2 ?(a—s, ?) ??i ??j , ??i ??j ?+c?? 1 + I(i = j) i,j=1

By combining the two previous lemmas, it is possible to derive the policy performance improvement obtained following the gradient direction. Theorem 3.3. Let the update of the parameters be ? 0 = ? + ??? J? (?). Then for any stationary policy ?(a—s, ?) and any starting state distribution ?, the difference in performance between ?? and ??0 is lower bounded by: J? (? 0) ? J? (?) ? ? k?? J? (?)k22 !

Z Z m X ? 2 ?(a—s, ?) ?2 ??i ??j ?? + d? (s) inf Q?? (s, a)dads 1?? S ??i ??j ?+c?? 1 + I(i = j) A c?(0,1) i,j=1
Z
? kQ?? k? ?? ?(a—s, ?)T ?? J? (?) da ? ? sup 2 2(1 ? ?) s?S A ! !2
Z m
X ? 2 ?(a—s, ?) ??i ??j
+?2 sup sup da .
?? 1 + I(i = j) i ??j s?S A c?(0,1) ?+c?? i,j=1

The above bound is a forth?order polynomial of the step size, whose stationary points, being the roots of a third?order polynomial ax3 + bx2 + cx + d, can be expressed in closed form. It is worth to notice that, for positive values of ?, the bound presents a single stationary point that corresponds to a local maximum. In fact, since a, b ? 0 and d ? 0, the Descartes? rule of signs gives the existence and uniqueness of the real positive root. In the following section, we will show, in the case of Gaussian policies, how the bound in Theorem 3.3 can be reduced to a second?order polynomial in ?, thus obtaining a simpler closed-form solution for optimal (w.r.t. the bound) step size.

4

The Gaussian Policy Model

In this section we consider the Gaussian policy model with fixed standard deviation ? and the mean is a linear combination of the state feature vector ?(?) using a parameter vector ? of size m:

2 ! 1 1 a ? ? T ?(s) ?(a—s, ?) = ? exp ? . 2 ? 2?? 2 In the case of Gaussian policies, each second?order derivative of policy ?? can be easily bounded. Lemma 4.1. For any Gaussian policy ?(a—s, ?) ? N (? T ?(s), ? 2), the second order derivative of the policy can be bounded as follows:

2 ? ?(a—s, ?) —?i (s)?j (s)— m

??i ??j ? ?2?? 3 , ?? ? R , ?a ? A. This result allows to restate Lemma 3.2 in the case of Gaussian policies: T ?(a—s, ? 0) ? ?(a—s, ?) ? ??? ?(a—s, ?) ?? J? (?) ? ?

2 ?2 T —?? J? (?)— —?(s)— . 3 2??

In the following we will assume that features ? are uniformly bounded: Assumption 4.1. All the basis functions are uniformly bounded by M? : —?i (s)—¡ M? , ?s ? S, ?i = 1, . . . , m. Exploiting Pinsker?s inequality [16]

(which upper bounds the total variation between two distributions with their Kullback?Liebler divergence), it is possible to provide the following upper bound to the supremum norm between two Gaussian policies. Lemma 4.2. For any pair of stationary policies ?? and ??0 , so that ? 0 = ? +??? J? (?), supremum norm of their difference can be upper bounded as follows: k??0 ? ?? k? ?

?M? k?? J? (?)k1 . ? 4

By plugging the results of Lemmas 4.1 and 4.2 into Equation (1) we can obtain a lower bound to the performance difference between a Gaussian policy ?? and another policy along the gradient direction that is quadratic in the step size ?. Theorem 4.3. For any starting state distribution ?, and any pair of stationary Gaussian policies T ?? ? N (? T ?(s), ? 2) and ??0 ? N (? 0 ?(s), ? 2), so that ? 0 = ? + ??? J? (?) and under Assumption 4.1, the difference between the performance of ??0 and the one of ?? can be lower bounded as follows: 2

J? (? 0) ? J? (?) ? ? k?? J? (?)k2

Z Z 2 1 T ? ? ?2 d??? (s) —?? J? (?)— —?(s)— Q?? (s, a)dads (1 ? ?) 2?? 3 S A ! ?M?2 2 ?? k?? J? (?)k1 kQ k? . + 2(1 ? ?)2 ? 2 Since the linear coefficient is positive and the quadratic one is negative, the bound in Theorem 4.3 has a single maximum attained for some positive value of ?. Corollary 4.4. The performance lower bound provided in Theorem 4.3 is maximized by choosing the following step size: ?? =

?

? 2??M?2

? (1 ? ?)2 2?? 3 k?? J? (?)k22 , 2 R R ? Q?? (s, a)dads k?? J? (?)k21 kQ?? k? + 2(1 ? ?) S d?? (s) —?? J? (?)—T —?(s)— A

that guarantees the following policy performance improvement 1 2 J? (? 0) ? J? (?) ? ?? k?? J? (?)k2 . 2

5

Approximate Framework

The solution for the tuning of the step size presented in the previous section depends on some constants (e.g., discount factor and the variance of the Gaussian policy) and requires to be able to compute some quantities (e.g., the policy gradient and the supremum value of the Q?function). In many real?world applications such quantities cannot be computed (e.g., when the state?transition model is unknown or too large for exact methods) and need to be estimated from experience samples. In this section, we study how the step size can be chosen when the gradient is estimated through sample trajectories to guarantee a performance improvement in high probability. For sake of easiness, we consider a simplified version of the bound in Theorem 4.3, in order to obtain a bound where the only element that needs to be estimated is the policy gradient ?? J? (?). Corollary 5.1. For any starting state distribution ?, and any pair of stationary Gaussian policies T ?? ? N (? T ?(s), ? 2) and ??0 ? N (? 0 ?(s), ? 2), so that ? 0 = ? + ??? J? (?) and under Assumption 4.1, the difference between the performance of ??0 and ?? is lower bounded by:

2 RM?2 k?? J? (?)k1 ? —A— 2 ? J? (? 0) ? J? (?) ? ? k?? J? (?)k2 ? ?2 + , 2 2?? 2(1 ? ?) (1 ? ?) ? 2 that is maximized by the following step size

value: ? 2 (1 ? ?)3 2?? 3 k?? J? (?)k2 ? ? ? = ?

2. ? 2?? + 2(1 ? ?)—A— RM?2 k?? J? (?)k1 Since we are assuming that the policy gradient ?? J? (?) is estimated through trajectory samples, the lower bound in Corollary 5.1 must take into consideration the associated approximation error. b ? J? (?) of Given a set of trajectories obtained following policy ?? , we can produce an estimate ? T the policy gradient and we assume to be able to produce a vector = [1 , . . . , m] , so that the i?th component of the approximation error is bounded at least with probability 1 ? ?:

b ? J? (?) ? i ? ?. P ??i J? (?) ? ? i 5

Given the approximation error vector , we can adjust the bound in Corollary 5.1 to produce a new m bound that holds at least with probability (1 ? ?) . In particular, to preserve the inequality sign, the estimated approximation error must be used to decrease the L2?norm of the policy gradient in the first term (the one that provides the positive contribution to the performance improvement) and to increase the L1?norm in the penalization term. To lower bound the L2?norm, we introduce the b ? J? (?) whose components are a lower bound to the absolute value of the policy gradient vector ? built on the basis of the approximation error : b ? J? (?) = max(—? b ? J? (?)— ? , 0), ? where 0 denotes the m?size vector with all zeros, and max denotes the component?wise maximum. b ? J? (?): Similarly, to upper bound the L1?norm of the policy gradient, we introduce the vector ? b ? J? (?) = —? b ? J? (?)— + . ? Theorem 5.2. Under the same assumptions of Corollary 5.1, and provided

that it is available a

b b policy gradient estimate ?? J? (?), so that P ??i J? (?) ? ??i J? (?) ? i ? ?, the difference m

between the performance of ??0 and ?? can be lower bounded at least with probability (1 ? ?) :

2

b

2 RM?2 ? ? J? (?) ? —A—

b

0 2 1 ? J? (?) ? J? (?) ? ? ?? J? (?) ? ? + , 2 2 2?? 2(1 ? ?) (1 ? ?) ? 2 that is maximized by the following step size value:

2 ?

b

(1 ? ?)3 2?? 3 ? ? J? (?) ? 2 ? b = ?

2 .

b

2 ? 2?? + 2(1 ? ?)—A— RM? ?? J? (?) 1

In the following, we will discuss how the approximation error of the policy gradient can be bounded. Among the several methods that have been proposed over the years, we focus on two well? understood policy?gradient estimation approaches: REINFORCE [3] and G(PO)MDP [4]/policy gradient theorem (PGT) [5]. 5.1

Approximation with REINFORCE gradient estimator

The REINFORCE approach [3] is the main exponent of the likelihood?ratio family. The episodic REINFORCE gradient estimator is given by: !! H N H X X X 1 RF n n l?1 n b ? J? (?) = ?? log ? (ak ; sk , ?) ? ? rl ? b , N n=1 k=1 l=1

where N is the number of H?step trajectories generated from a system by roll?outs and b ? R is a baseline that can be chosen arbitrary, but usually with the goal of minimizing the variance of the gradient estimator. The main drawback of REINFORCE is its variance, that is strongly affected by the length of the trajectory horizon H. The goal is to determine the number of trajectories N in order to obtain the desired accuracy of the gradient estimate. To achieve this, we exploit the upper bound to the variance of the episodic REINFORCE gradient estimator introduced in [17] for Gaussian policies. Lemma 5.3 (Adapted from Theorem 2 in [17]). Given a Gaussian policy ?(a—s, ?) ?

N ? T ?(s), ? 2 , under the assumption of uniformly bounded rewards and basis functions (Assumption 4.1), we have the following upper bound to the variance of the i?th component of the episodic b ? J?RF (?): REINFORCE gradient estimate ? i

R2 M 2 H 1 ? ? H 2 ? RF b ? J? (?) ? V ar ? . i 2 N ? 2 (1 ? ?) 6

The result in the previous Lemma combined with the Chebyshev?s inequality allows to provide a high?probability upper bound to the gradient approximation error using the episodic REINFORCE gradient estimator.

Theorem 5.4. Given a Gaussian policy ?(a—s, ?) ? N ? T ?(s), ? 2 , under the assumption of uniformly bounded rewards and basis functions (Assumption 4.1), using the following number of H?step trajectories: 2 R2 M?2 H 1 ? ? H N= , 2 ?2i ? 2 (1 ? ?) b ? J?RF (?) generated by REINFORCE is such that with probability 1 ? ?: the gradient estimate ? i

b ??i J?RF (?) ? ??i J? (?) ? i . 5.2

Approximation with G(PO)MDP/PGT gradient estimator

Although the REINFORCE method is guaranteed to converge at the true gradient at the fastest possible pace, its large variance can be problematic in practice. Advances in the likelihood ratio gradient estimators have produced new approaches that significantly reduce the variance of the estimate. Focusing on the class of ?vanilla? gradient estimator, two main approaches have been proposed: policy gradient theorem (PGT) [5] and G(PO)MDP [4]. In [6], the authors show that, while the algorithms b ? J?G(PO)MDP (?). For this b ? J?P GT (?) = ? look different, their gradient estimate are equal, i.e., ? reason, we can limit our attention to the PGT formulation: !! H H H X X X 1 b ? J?P GT (?) = ? ?? log ? (ank ; snk , ?) ? l?1 rln ? bnl , N n=1 k=1 l=k bnl

where ? R have the objective to reduce the variance of the gradient estimate. Following the procedure used to bound the approximation error of REINFORCE, we need an upper bound to the variance of the gradient estimate of PGT that is provided by the following lemma (whose proof is similar to the one used in [17] for the REINFORCE case).

Lemma 5.5. Given a Gaussian policy ?(a—s, ?) ? N ? T ?(s), ? 2 , under the assumption of uniformly bounded rewards and basis functions (Assumption 4.1), we have the following upper bound b ? J?P GT (?): to the variance of the i?th component of the PGT gradient estimate ? i

b ? J?P GT (?) ? V ar ? i

R2 M?2

2

N (1 ? ?) ? 2

H 1 ? ? 2H 2H H1?? + H? ? 2? . 1 ? ?2 1??

As expected, since the variance of the gradient estimate obtained with PGT is smaller than the one with REINFORCE, also the upper bound of the PGT variance is smaller than REINFORCE one. In particular, while the variance with REINFORCE grows linearly with the time horizon, using PGT the dependence on the time horizon is significantly smaller. Finally, we can derive the upper bound for the approximation error of the gradient estimated of PGT.

Theorem 5.6. Given a Gaussian policy ?(a—s, ?) ? N ? T ?(s), ? 2 , under the assumption of uniformly bounded rewards and basis functions (Assumption 4.1), using the following number of H?step trajectories:

H R2 M?2 1 ? ? 2H 2H H1?? N= 2 + H? ? 2? 2 1 ? ?2 1?? ?i ? 2 (1 ? ?) b ? J?P GT (?) generated by PGT is such that with probability 1 ? ?: the gradient estimate ? i

b

??i J?P GT (?) ? ??i J? (?) ? i .

7

?const

?t =

?0 t

1e ? 07 1e ? 06 1e ? 05 1e ? 04 1e ? 03 1e ? 05 1e ? 04

?

?

0.50 itmax itmax 17138 1675 ? itmax itmax 24106

0.75 itmax itmax 8669 697 ? itmax itmax 7271

1.00 itmax itmax 5120 499 ? itmax itmax 3279

1.25 itmax itmax 3348 ? ? itmax ? 1838

? 1.50 itmax 23651 2342 ? ? itmax ? 1172

1.75 itmax 17516 1714 ? ? itmax ? 813

2.00 itmax 13480 1287 ? ? itmax ? 598

5.00 21888 2163 ? ? ? ? ? 1

7.50 9740 849 ? ? ? ? ? 58

Table 1: Convergence speed in exact LQG scenario with ? = 0.95. The table reports the number of iterations required by the exact gradient approach, starting from ? = 0, to learn the optimal policy parameter ?? = ?0.6037 with an accuracy of 0.01, for different step?size values. Three different set of experiments are shown: constant step size, decreasing step size, and the step size proposed in Corollary 4.4. The table contains itmax when no convergence happens in

30, 000 iterations, and ? when the algorithm diverges (? ¡ ?1 or ? ¿ 0). Best performances are reported in boldface. 10, 000 RF PGT

it 822 29, 761

? ?0.0030 ?0.2176

Number of trajectories 100, 000 it ? 51, 731 ?0.3068 63, 985 ?0.4013

500, 000 it ? 75, 345 ?0.4088 83, 983 ?0.4558

Table 2: Convergence speed in approximate LQG scenario with ? = 0.9. The table reports, starting from ? = 0 and fixed ? = 1, the number of iterations performed before the proposed step size ? b becomes 0 and the last value of the policy parameter. Results are shown for different number of trajectories (of 20 steps each) used in the gradient estimation by REINFORCE and PGT.

6

Numerical Simulations and Discussion

In this section we show results related to some numerical simulations of policy gradient in the linear?quadratic Gaussian regulation (LQG) problem as formulated in [6]. The LQG problem is

characterized by a transition model st+1 ? N st + at , ? 2 , Gaussian policy at ? N ? ? s, ? 2 and quadratic reward rt = ?0.5(s2t + a2t). The range of state and action spaces is bounded to the interval [?2, 2] and the initial state is drawn uniformly at random. This scenario is particularly instructive since it allows to exactly compute all terms involved in the bounds. We first present results in the exact scenario and then we move toward the approximated one. Table 1 shows how the number of iterations required to learn a near?optimal value of the policy parameter changes according to the standard deviation of the Gaussian policy and the step?size value. As expected, very small values of the step size allow to avoid divergence, but the learning process needs many iterations to reach a good performance (this can be observed both when the step size is kept constant and when it decreases). On the other hand, larger step?size values may lead to divergence. In this example, the higher the policy variance, the lower is the step size value that allows to avoid divergence, since, in LQG, higher policy variance implies larger policy gradient values. Using the step size ?? from Corollary 4.4 the policy gradient algorithm avoids divergence (since it guarantees an improvement at each iteration), and the speed of convergence is strongly affected by the variance of the Gaussian policy. In general, when the policy are nearly deterministic (small variance in the Gaussian case), small changes in the parameters lead to large distances between the policies, thus negatively affecting the lower bound in Equation 1. As we can notice from the expression of ?? in Corollary 4.4, considering policies with high variance (that might be a problem in real?world applications) allows to safely take larger step size, thus speeding up the learning process. Nonetheless, increasing the variance over some threshold (making policies nearly random) produces very bad policies, so that changing the policy parameter has a small impact on the performance, and as a result slows down the learning process. How to identify an optimal variance value is an interesting future research direction. Table 2 provides numerical results in the approximated settings, showing the effect of varying the number of trajectories used to estimate the gradient by

REINFORCE and PGT. Increasing the number of trajectories reduces the uncertainty on the gradient estimates, thus allowing to use larger step sizes and reaching better performances. Furthermore, the smaller variance of PGT w.r.t. REINFORCE allows the former to achieve better performances. However, even with a large number of trajectories, the approximated errors are still quite large preventing to reach very high performance. For this reason, future studies will try to derive tighter bounds. Further developments include extending these results to other policy models (e.g., Gibbs policies) and to other policy gradient approaches (e.g., natural gradient). 8

2 References

[1] Jan Peters and Stefan Schaal. Policy gradient methods for robotics. In Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pages 2219?2225. IEEE, 2006. [2] James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. Automatic Control, IEEE Transactions on, 37(3):332?341, 1992. [3] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3-4):229?256, May 1992. [4] Jonathan Baxter and Peter L. Bartlett. Infinite-horizon policy-gradient estimation. Journal of Artificial Intelligence Research, 15:319?350, 2001. [5] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. Advances in neural information processing systems, 12(22), 2000. [6] Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. Neural Networks, 21(4):682?697, 2008. [7] Sham Kakade. A natural policy gradient. Advances in neural information processing systems, 14:1531?1538, 2001. [8] Jan Peters and Stefan Schaal. Natural actor-critic. Neurocomputing, 71(7):1180?1190, 2008. [9] Herbert Robbins and Sutton Monro. A stochastic approximation method. The Annals of Mathematical Statistics, pages 400?407, 1951. [10] P. Wagner. A reinterpretation of the policy oscillation phenomenon in approximate policy iteration. Advances in Neural Information Processing Systems, 24, 2011. [11] Jorge J Mor?e and David J Thuente. Line search algorithms with guaranteed sufficient decrease. ACM Transactions on Mathematical Software (TOMS), 20(3):286?307, 1994. [12] J. Kober and J. Peters. Policy search for motor primitives in robotics. In Advances in Neural Information Processing Systems 22 (NIPS 2008), Cambridge, MA: MIT Press, 2009. [13] Nikos Vlassis, Marc Toussaint, Georgios Kontes, and Savas Piperidis. Learning model-free robot control by a monte carlo em algorithm. Autonomous Robots, 27(2):123?130, 2009. [14] S.M. Kakade. On the sample complexity of reinforcement learning. PhD thesis, PhD thesis, University College London, 2003. [15] Matteo Pirotta, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. Safe policy iteration. In Sanjoy Dasgupta and David McAllester, editors, Proceedings of the 30th International Conference on Machine Learning (ICML-13), volume 28, pages 307?315. JMLR Workshop and Conference Proceedings, May 2013. [16] S.

Pinsker. Information and Information Stability of Random Variable and Processes. HoldenDay Series in Time Series Analysis. Holden-Day, Inc., 1964. [17] Tingting Zhao, Hirotaka Hachiya, Gang Niu, and Masashi Sugiyama. Analysis and improvement of policy gradient estimation. Neural Networks, 26(0):118 ? 129, 2012.

9