Policy Learning – a unified perspective with applications in robotics

Jan Peters[1,2], Jens Kober[1], and Duy Nguyen-Tuong[1]

[1] Max-Planck Institute for Biological Cybernetics,
Spemannstr. 32, 72074 Tübingen,
{jrpeters,kober,duy}@tuebingen.mpg.de,
WWW home page: http://kyb.mpg.de/ jrpeters
[2] University of Southern California,
Los Angeles, CA 90089, USA

Abstract. Policy Learning approaches are among the best suited methods for high-dimensional, continuous control systems such as anthropomorphic robot arms and humanoid robots. In this paper, we show two contributions: firstly, we show a unified perspective which allows us to derive several policy learning algorithms from a common point of view, i.e, policy gradient algorithms, natural-gradient algorithms and EM-like policy learning. Secondly, we present several applications to both robot motor primitive learning as well as to robot control in task space. Results both from simulation and several different real robots are shown.

1 Introduction

In order to ever leave the well-structured environments of factory floors and research labs, future robots will require the ability to aquire novel behaviors, motor skills and control policies as well as to improve existing ones. Reinforcement learning is probably the most general framework in which such robot learning problems can be phrased. However, most of the methods proposed in the reinforcement learning community to date are not applicable to robotics as they do not scale beyond robots with more than one to three degrees of freedom. Policy learning methods are a notable exception to this statement. Starting with the pioneering work of Gullapali, Franklin and Benbrahim [4, 8] in the early 1990s, these methods have been applied to a variety of robot learning problems ranging from simple control tasks (e.g., balancing a ball-on a beam [3], and pole-balancing [11]) to complex learning tasks involving many degrees of freedom such as learning of complex motor skills [8, 15, 21] and locomotion [10, 23, 13, 6, 25, 16, 7].

In this paper, we expand previous work on policy learning towards the direction of a unified framework for policy learning. For doing so, we discuss upper and lower bounds on policy improvements. From the lower bound, we derive a cost function which allows us to derive policy gradient approaches, natural policy gradient approaches as well as EM-like policy learning methods. Furthermore, we show several applications in the context of robot skill learning. These applications include both learning task-space control with reinforcement learning as well as motor primitive learning. Results of both real robots and simulation are being shown.

2 Policy Learning Approaches

As outlined before, we need two different styles of policy learning algorithms, i.e., methods for long-term reward optimization and methods for immediate improvement. We can unify this goal by stating a cost function

$$J(\boldsymbol{\theta}) = \int_{\mathbb{T}} p_{\boldsymbol{\theta}}(\boldsymbol{\tau})\, r(\boldsymbol{\tau})\, d\boldsymbol{\tau}, \tag{1}$$

where $\boldsymbol{\tau}$ denotes a path, e.g., $\boldsymbol{\tau} = [\mathbf{x}_{1:n}, \mathbf{u}_{1:n}]$ with states $\mathbf{x}_{1:n}$ and actions $\mathbf{u}_{1:n}$, $r(\boldsymbol{\tau})$ denotes the reward along the path, e.g., $r(\boldsymbol{\tau}) = \sum_{t=1}^{n} \gamma^t r_t$ and $p_{\boldsymbol{\theta}}(d\boldsymbol{\tau})$ denotes the path probability density $p_{\boldsymbol{\theta}}(d\boldsymbol{\tau}) = p(\mathbf{x}_1) \prod_{t=1}^{n-1} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)\, \pi(\mathbf{u}_t|\mathbf{x}_t; \boldsymbol{\theta})$ with a first-state distribution $p(\mathbf{x}_1)$, a state transition $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$ and a policy $\pi(\mathbf{u}_t|\mathbf{x}_t; \boldsymbol{\theta})$. Note, that $p_{\boldsymbol{\theta}}(\boldsymbol{\tau})\, r(\boldsymbol{\tau})$ is an improper distribution, i.e., does not integrate to 1. The policy $\pi(\mathbf{u}_t|\mathbf{x}_t; \boldsymbol{\theta})$ is the function which we intend to learn by optimizing its parameters $\boldsymbol{\theta} \in \mathbb{R}^N$. Many policy learning algorithms have started optimize this cost function, including policy gradient methods [1], actor-critic methods [24, 14], the Natural Actor-Critic [19, 20, 22] and Reward-Weighted Regression [18]. In the remainder of this section, we will sketch a unified approach to policy optimization which allows the derivation of all of the methods above from the variation of a single cost function. This section might appear rather abstract in comparison to the rest of the paper; however, it contains major novelties as it allows a coherent treatment of many previous and future approaches.

2.1 Bounds for Policy Updates

In this section, we will look at two problems in policy learning, i.e., an upper bound and a lower bound on policy improvements. The upper bound outlines why a greedy operator is not a useful solution while the lower bound will be used to derive useful policy updates.

Upper Bound on Policy Improvements. In the stochastic programming community, it is well-known that the greedy approach to policy optimization suffers from the major drawback that it can return only a biassed solution. This drawback can be formalized straighforwardly by showing that if we optimize $J(\boldsymbol{\theta})$ and approximate it by samples, e.g., by $\hat{J}_S(\boldsymbol{\theta}) = \sum_{s=1}^{S} p_{\boldsymbol{\theta}}(\boldsymbol{\tau}_s)\, r(\boldsymbol{\tau}_s) \approx J(\boldsymbol{\theta})$, we obtain the fundamental relationship

$$E\{\max_{\boldsymbol{\theta}} \hat{J}_S(\boldsymbol{\theta})\} \geq \max_{\boldsymbol{\theta}} E\{\hat{J}_S(\boldsymbol{\theta})\}, \tag{2}$$

which can be shown straightforwardly by first realizing the that the maximum is always larger than any member of a sample. Thus, a subsequent expectation will not change this fact nor the subsequent optimization of the lower bound. Thus, a policy which is optimized by doing a greedy step in parameter space is guaranteed to be biased in the presence of errors with a bias of $b_S(\boldsymbol{\theta}) = E\{\max_{\boldsymbol{\theta}} \hat{J}_S(\boldsymbol{\theta})\} - \max_{\boldsymbol{\theta}} E\{\hat{J}_S(\boldsymbol{\theta})\} \geq 0$. However, we can also show that the bias decreases over the number of samples, i.e., $b_S(\boldsymbol{\theta}) \geq b_{S+1}(\boldsymbol{\theta})$, and converges to zero for infinite samples, i.e., $\lim_{S\to\infty} b_S(\boldsymbol{\theta}) = 0$

[17]. This optimization bias illustrates the deficiencies of the greedy operator: for finite data any policy update is problematic and can result into unstable learning processes with oscillations, divergence, etc as frequently observed in the reinforcement learning community [2, 1].

Lower Bound on Policy Improvements. In other branches of machine learning, the focus has been on lower bounds, e.g., in Expectation-Maximization (EM) algorithms. The reasons for this preference apply in policy learning: if the lower bound also becomes an equality for the sampling policy, we can guarantee that the policy will be improved. Surprisingly, the lower bounds in supervised learning can be transferred with ease. For doing so, we look at the scenario (suggested in [5]) that we have a policy $\boldsymbol{\theta}'$ and intend to match the path distribution generated by this policy to the success weighted path distribution, then we intend to minimize the distance between both distributions, i.e., $D\left(p_{\boldsymbol{\theta}'}\left(\boldsymbol{\tau}\right)||p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right)r\left(\boldsymbol{\tau}\right)\right)$. Surprisingly, this results into a lower bound using Jensen's inequality and the convexity of the logarithm function. This results into

$$\log J(\boldsymbol{\theta}') = \log \int \frac{p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right)}{p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right)} p_{\boldsymbol{\theta}'}\left(\boldsymbol{\tau}\right) r\left(\boldsymbol{\tau}\right) d\boldsymbol{\tau}, \tag{3}$$

$$\geq \int p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right) r\left(\boldsymbol{\tau}\right) \log \frac{p_{\boldsymbol{\theta}'}\left(\boldsymbol{\tau}\right)}{p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right)} d\boldsymbol{\tau} \propto -D\left(p_{\boldsymbol{\theta}'}\left(\boldsymbol{\tau}\right)||p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right)r\left(\boldsymbol{\tau}\right)\right), \tag{4}$$

where $D\left(p_{\boldsymbol{\theta}'}\left(\boldsymbol{\tau}\right)||p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right)\right) = \int p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right) \log(p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right)/p_{\boldsymbol{\theta}'}\left(\boldsymbol{\tau}\right))d\boldsymbol{\tau}$ is the Kullback-Leibler divergence, i.e., a distance measure for probability distributions. With other words, we have the lower bound $J(\boldsymbol{\theta}') \geq \exp\left(D\left(p_{\boldsymbol{\theta}'}\left(\boldsymbol{\tau}\right)||p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right)r\left(\boldsymbol{\tau}\right)\right)\right)$, and we can minimize

$$J_{\mathrm{KL}} = D\left(p_{\boldsymbol{\theta}'}\left(\boldsymbol{\tau}\right)||p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right)r\left(\boldsymbol{\tau}\right)\right) = \int p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right) r\left(\boldsymbol{\tau}\right) \log \frac{p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right) r\left(\boldsymbol{\tau}\right)}{p_{\boldsymbol{\theta}'}\left(\boldsymbol{\tau}\right)} d\boldsymbol{\tau} \tag{5}$$

without the problems which have troubled the reinforcement learning community when optimizing the upper bound as we are guaranteed to improve the policy. However, in many cases, we might intend to punish divergence from the previous solution. In this case, we intend to additionally control the distance which we move away from our previous policy, e.g., minimize the term $J_+ = -D\left(p_{\boldsymbol{\theta}}\left(\boldsymbol{\tau}\right)||p_{\boldsymbol{\theta}'}\left(\boldsymbol{\tau}\right)\right)$. We can combine these into a joint cost function

$$J_{\mathrm{KL}+} = J_{\mathrm{KL}} + \lambda J_+, \tag{6}$$

where $\lambda \in \mathbb{R}^+$ is a positive punishment factor with $0 \leq \lambda \leq J(\boldsymbol{\theta})$. Note that the exchange of the arguments is due to the fact that the Kullback-Leibler divergence is unsymmetric. This second term will play an important rule as both baselines and natural policy gradients are a directly result of it. The proper determination of λ is non-trivial and depends on the method. E.g., in policy gradients, this becomes the baseline.

2.2 Resulting Approaches for Policy Learning

We now proceed into deriving three different methods for lower bound optimization, i.e., policy gradients, the natural actor-critic and reward-weighted regression. All three of these can be derived from this one perspective.

Policy Gradients Approaches. It has recently been recognized that policy gradient methods [2, 1] do not suffer from the drawbacks of the greedy operator and, thus, had a large revival in recent years. We can derive policy gradient approaches straightforwardly from this formulation using the steepest descent of the first order taylor extension

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \alpha(\boldsymbol{\nabla} J_{\mathrm{KL}} - \lambda \boldsymbol{\nabla} J_{+}) \tag{7}$$

$$= \boldsymbol{\theta} + \alpha \int p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) (r(\boldsymbol{\tau}) - \lambda) \boldsymbol{\nabla} \log p_{\boldsymbol{\theta}'}(\boldsymbol{\tau}) \, d\boldsymbol{\tau}, \tag{8}$$

where α is a learning rate. This is only true as for the first derivative $\boldsymbol{\nabla} D(p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) || p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})) = \boldsymbol{\nabla} D(p_{\boldsymbol{\theta}'}(\boldsymbol{\tau}) || p_{\boldsymbol{\theta}}(\boldsymbol{\tau}))$. The punishment factor from before simply becomes the baseline of the policy gradient estimator. As $\boldsymbol{\nabla} \log p_{\boldsymbol{\theta}'}(\boldsymbol{\tau}) = \sum_{t=1}^{n-1} \boldsymbol{\nabla} \log \pi(\mathbf{u}_t | \mathbf{x}_t; \boldsymbol{\theta})$, we obtain the straightforward gradient estimator also known as REINFORCE, policy gradient theorem or GPOMDP, for an overview see [1]. The punishment term only constrains the variance of the policy gradient estimate and vanishes as $\boldsymbol{\nabla} J_{\mathrm{KL}+} = \boldsymbol{\nabla} J_{\mathrm{KL}}$ for infinite data. However, this policy update can be shown to be rather slow [9, 19, 20, 22].

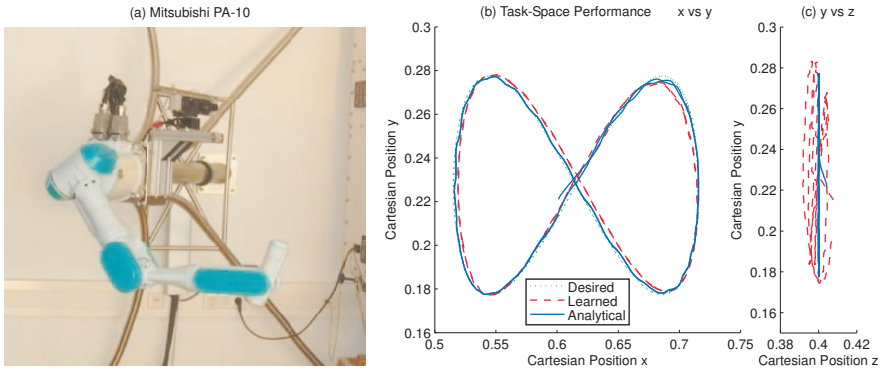

Fig. 1. (a) Mitsubishi PA-10 robot arm with seven degrees of freedom used in the experiments in this paper. (b) This figure illustrates the task performance of both the analytical and the learned resolved velocity control laws. Here, the green dotted line shows the desired trajectory which the robot should follow, the red dashed line is the performance of the real-time learning control law while the blue solid line shows the performance of the resolved velocity control law. Note, that while the online learning solution is as good as the analytical solution, it still yields comparable performance without any pre-training of the local control laws before the online learning (Nevertheless, the predictors were pre-trained).

Natural Policy Gradient Approaches. Suprisingly, the speed update can be improved significantly if we punish higher order terms of J_{+}, e.g., the second term of the taylor

expansion yields

$$\boldsymbol{\theta}' = \operatorname{argmax}_{\boldsymbol{\theta}'}(\boldsymbol{\theta}' - \boldsymbol{\theta})^T(\boldsymbol{\nabla} J_{\mathrm{KL}} - \lambda \boldsymbol{\nabla} J_+) - \frac{1}{2}\lambda(\boldsymbol{\theta}' - \boldsymbol{\theta})^T \boldsymbol{\nabla}^2 J_+(\boldsymbol{\theta}' - \boldsymbol{\theta}) \quad (9)$$

$$= \lambda \left(\boldsymbol{\nabla}^2 J_+\right)^{-1} (\boldsymbol{\nabla} J_{\mathrm{KL}} - \lambda \boldsymbol{\nabla} J_+) = \lambda \mathbf{F}^{-1} g_1, \quad (10)$$

where $\mathbf{F} = \boldsymbol{\nabla}^2 D\left(p_{\boldsymbol{\theta}}(\boldsymbol{\tau}) || p_{\boldsymbol{\theta}'}(\boldsymbol{\tau})\right) = \boldsymbol{\nabla}^2 D\left(p_{\boldsymbol{\theta}'}(\boldsymbol{\tau}) || p_{\boldsymbol{\theta}}(\boldsymbol{\tau})\right) = \boldsymbol{\nabla}^2 J_+$ is also known as the Fisher information matrix and the resulting policy update $\mathbf{g}_2$ is known as the Natural Policy Gradient. Surprisingly, the second order term has not yet been expanded and no Natural second-order gradient approaches are known. Thus, this could potentially be a great topic for future research.

EM-Policy Learning. In a very special case, we can solve for the optimal policy parameters, e.g, for policy which are linear in the log-derivatives such as

$$\boldsymbol{\nabla} \log \pi(\mathbf{u}_t | \mathbf{x}_t; \boldsymbol{\theta}) = \mathbf{A}(\mathbf{x}_t, \mathbf{u}_t)\boldsymbol{\theta} + \mathbf{b}(\mathbf{x}_t, \mathbf{u}_t), \quad (11)$$

it is straightforward to derive an EM algorithm such as

$$\boldsymbol{\theta}' = \alpha^{-1}\beta, \quad (12)$$

$$\boldsymbol{\alpha} = \int p_{\boldsymbol{\theta}}(\boldsymbol{\tau})(r(\boldsymbol{\tau}) - \lambda) \sum_{t=1}^{n} \mathbf{A}(\mathbf{x}_t, \mathbf{u}_t) d\boldsymbol{\tau}, \quad (13)$$

$$\boldsymbol{\beta} = \int p_{\boldsymbol{\theta}}(\boldsymbol{\tau})(r(\boldsymbol{\tau}) - \lambda) \sum_{t=1}^{n} b(\mathbf{x}_t, \mathbf{u}_t) d\boldsymbol{\tau}. \quad (14)$$

This type of algorithms can result into very fast policy updates if applicable. It does not require a learning rate and is guaranteed to converge to at least a locally optimal solution.

2.3 Sketch of the Resulting Algorithms

Thus, we have developed two different classes of algorithms, i.e., the Natural Actor-Critic and the Reward-Weighted Regression.

Natural Actor-Critic. The Natural Actor-Critic algorithms [19, 20] instantiations of the natural policy gradient previously described with a large or infinite horizon n. They are considered the fastest policy gradient methods to date and "the current method of choice" [1]. They rely on the insight that we need to maximize the reward while keeping the loss of experience constant, i.e., we need to measure the distance between our current path distribution and the new path distribution created by the policy. This distance can be measured by the Kullback-Leibler divergence and approximated using the Fisher information metric resulting in a natural policy gradient approach. This natural policy gradient has a connection to the recently introduced compatible function approximation, which allows to obtain the Natural Actor-Critic. Interestingly, earlier

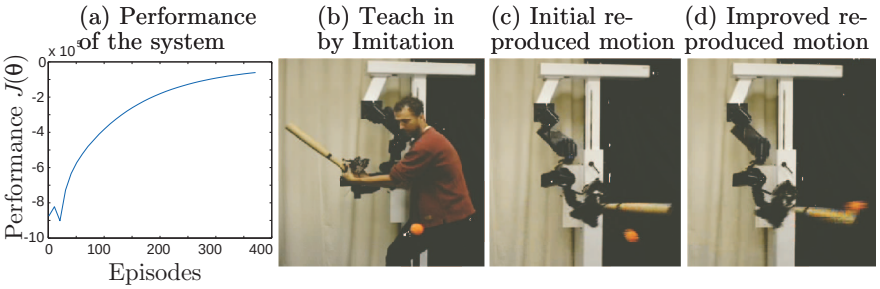

Fig. 2. This figure shows (a) the performance of a baseball swing task when using the motor primitives for learning. In (b), the learning system is initialized by imitation learning, in (c) it is initially failing at reproducing the motor behavior, and (d) after several hundred episodes exhibiting a nicely learned batting.

Actor-Critic approaches can be derived from this new approach. In application to motor primitive learning, we can demonstrate that the Natural Actor-Critic outperforms both finite-difference gradients as well as 'vanilla' policy gradient methods with optimal baselines.

Reward-Weighted Regression. In contrast to Natural Actor-Critic algorithms, the Reward-Weighted Regression algorithm [18] focuses on immediate reward improvement, i.e., $n = 1$, and employs an adaptation of the expectation maximization (EM) policy learning algorithm for reinforcement learning as previously described instead of a gradient based approach. The key difference here is that when using immediate rewards, we can learn from our actions directly, i.e., use them as training examples similar to a supervised learning problem with a higher priority for samples with a higher reward. Thus, this problem is a reward-weighted regression problem, i.e., it has a well-defined solution which can be obtained using established regression techniques. While we have given a more intuitive explanation of this algorithm, it corresponds to a properly derived maximization-maximization (MM) algorithm which maximizes a lower bound on the immediate reward similar to an EM algorithm. Our applications show that it scales to high dimensional domains and learns a good policy without any imitation of a human teacher.

Policy Learning by Weighting Exploration with Rewards. A recent development is the policy learning by weighting exploration with rewards or PoWER method [12]. In this case, we attempt to extend the previous work of the reward-weighted regression from the immediate reward case to longer horizons. When using the reward-weighted regression, we suffer from a multitude of artificial local plateaus and will not converge to the optimal solution. However, the insight that state-dependent exploration rates result into this algorithm. Again, an EM algorithm is obtained and turns out to be highly efficient in the context of learning Kendama [12].

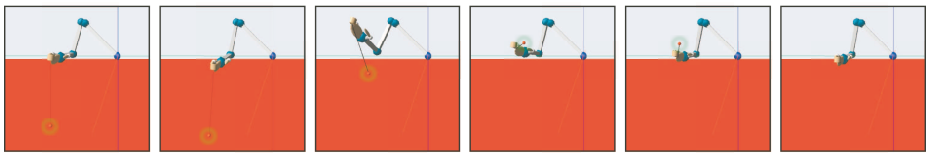

Fig. 3. This figure illustrates the successfully learned motion of the Kendama trial. For achieving this motion, motor primitives with external feedback had to be learned. Only an imitation from a human trial recorded in a VICON setup and, subsequently, reinforcement learning allowed to learn this motion reliably.

3 Robot Application

The general setup presented in this paper can be applied in robotics using analytical models as well as the presented learning algorithms. The applications presented in this paper include motor primitive learning and operational space control.

3.1 Learning Operational Space Control

Operational space control is one of the most general frameworks for obtaining task-level control laws in robotics. In this paper, we present a learning framework for operational space control which is a result of a reformulation of operational space control as a general point-wise optimal control framework and our insights into immediate reward reinforcement learning. While the general learning of operational space controllers with redundant degrees of freedom is non-convex and thus global supervised learning techniques cannot be applied straightforwardly, we can gain two insights, i.e., that the problem is locally convex and that our point-wise cost function allows us to ensure global consistency among the local solutions. We show that this can yield the analytically determined optimal solution for simulated three degrees of freedom arms where we can sample the state-space sufficiently. Similarly, we can show the framework works well for simulations of the both three and seven degrees of freedom robot arms as presented in Figure 1.

3.2 Motor Primitive Improvement by Reinforcement Learning

The main application of our long-term improvement framework is the optimization of motor primitives. Here, we follow essentially the previously outlined idea of acquiring an initial solution by supervised learning and then using reinforcement learning for motor primitive improvement. For this, we demonstrate both comparisons of motor primitive learning with different policy gradient methods, i.e., finite difference methods, 'vanilla' policy gradient methods and the Natural Actor-Critic, as well as an application of the most successful method, the Natural Actor-Critic to T-Ball learning on a physical, anthropomorphic SARCOS Master Arm, see Figure 2.

Another example for applying policy learning to the motor primitive frame is the children's game Kendama [12]. Here, we have managed to learn a good policy again

from a human demonstration which fails to bring the ball into the cup. Subsequently, we have learned how to improve our policy with the PoWER method [12] and have managed to learn a good motor primitive-based control policy. The results are shown in Figure 3.

4 Conclusion

In conclusion, in this paper, we have presented a general framework for learning motor skills which is based on a thorough, analytically understanding of robot task representation and execution. We have introduced a general framework for policy learning which allows the derivation of a variety of novel reinforcement learning methods including the Natural Actor-Critic and the Reward-Weighted Regression algorithm. We demonstrate the efficiency of these reinforcement learning methods in the application of learning to hit a baseball with an anthropomorphic robot arm on a physical SARCOS master arm using the Natural Actor-Critic, and in simulation for the learning of operational space with reward-weighted regression.

References

1. Douglas Aberdeen. POMDPs and policy gradients. In *Proceedings of the Machine Learning Summer School (MLSS)*, Canberra, Australia, 2006.
2. Douglas A. Aberdeen. *Policy-Gradient Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Australian National Unversity, 2003.
3. H. Benbrahim, J. Doleac, J. Franklin, and O. Selfridge. Real-time learning: A ball on a beam. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Baltimore, MD, 1992.
4. Hamid Benbrahim and Judy Franklin. Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems*, 22:283–302, 1997.
5. Peter Dayan and Geoffrey E. Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 9(2):271–278, 1997.
6. Gen Endo, Jun Morimoto, Takamitsu Matsubara, Jun Nakanishi, and Gordon Cheng. Learning cpg sensory feedback with policy gradient for biped locomotion for a full-body humanoid. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, PA, 2005.
7. T. Geng, B. Porr, and F. Wörgötter. Fast biped walking with a reflexive neuronal controller and real-time online learning. *Submitted to Int. Journal of Robotics Res.*, 2005.
8. V. Gullapalli, J. Franklin, and H. Benbrahim. Aquiring robot skills via reinforcement learning. *IEEE Control Systems Journal, Special Issue on Robotics: Capturing Natural Motion*, 4(1):13–24, February 1994.
9. S. A. Kakade. Natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14, Vancouver, CA, 2002.
10. H. Kimura and S. Kobayashi. Reinforcement learning for locomotion of a two-linked robot arm. In *Proceedings of the Europian Workshop on Learning Robots (EWLR)*, pages 144–153, Brighton, UK, 1997.
11. H. Kimura and S. Kobayashi. Reinforcement learning for continuous action using stochastic gradient ascent. In *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)*, volume 5, pages 288–295, Madison, Wisconsin, 1998.

12. Jens Kober and Jan Peters. Reinforcement learning of perceptual coupling for motor primitives. In *Submitted to the European Workshop on Reinforcement Learning (EWRL)*, 2008.
13. Nate Kohl and Peter Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, May 2004.
14. V. Konda and J. Tsitsiklis. Actor-critic algorithms. *Advances in Neural Information Processing Systems 12*, 2000.
15. Noriaki Mitsunaga, Christian Smith, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Robot behavior adaptation for human-robot interaction based on policy gradient reinforcement learning. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1594–1601, Edmonton, Canada, 2005.
16. Takeshi Mori, Yutaka Nakamura, Masa aki Sato, and Shin Ishii. Reinforcement learning for cpg-driven biped robot. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 623–630, San Jose, CA, 2004.
17. J. Peters. The bias of the greedy update. Technical report, University of Southern California, 2007.
18. J. Peters and S. Schaal. Learning operational space control. In *Proceedings of Robotics: Science and Systems (RSS)*, Philadelphia, PA, 2006.
19. J. Peters, S. Vijayakumar, and S. Schaal. Reinforcement learning for humanoid robotics. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, Karlsruhe, Germany, September 2003.
20. J. Peters, S. Vijayakumar, and S. Schaal. Natural actor-critic. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 280–291. springer, 2005.
21. Jan Peters, Sethu Vijayakumar, and Stefan Schaal. Natural actor-critic. In *Proceedings of the European Machine Learning Conference (ECML)*, Porto, Portugal, 2005.
22. Silvia Richter, Douglas Aberdeen, and Jin Yu. Natural actor-critic for road traffic optimisation. In B. Schoelkopf, J.C. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems*, volume 19, Cambridge, MA, 2007. MIT Press.
23. M. Sato, Y. Nakamura, and S. Ishii. Reinforcement learning for biped locomotion. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, Lecture Notes in Computer Science, pages 777–782. Springer-Verlag, 2002.
24. 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. Mueller, editors, *Advances in Neural Information Processing Systems (NIPS)*, Denver, CO, 2000. MIT Press.
25. Russ Tedrake, Teresa Weirui Zhang, and H. Sebastian Seung. Learning to walk in 20 minutes. In *Proceedings of the Yale Workshop on Adaptive and Learning Systems*, New Haven, CT, 2005. Yale University, New Haven.