An Approximate Bayesian Reinforcement Learning Approach Using Robust Control Policy and Tree Search

Toru Hishinuma, Kei Senda
Department of Aeronautics and Astronautics
Graduate School of Engineering
Kyoto University
hishinuma.toru.43n@st.kyoto-u.ac.jp, senda@kuaero.kyoto-u.ac.jp

Abstract

For autonomous robots, we propose an approximate model-based Bayesian reinforcement learning (MB-BRL) approach that reduces real-world samples within feasible computational efforts. Firstly, to find an approximate solution of an original undiscounted infinite horizon MB-BRL problem with a cost-free termination, we consider a finite horizon (FH) MB-BRL problem in which terminal costs are given by robust control policies. The resulting performance is better than or equal to the performance obtained with a robust method, while the resulting policy may choose an explorative behavior to get useful information about parametric model uncertainty for reducing real-world samples. Secondly, to obtain a feasible solution of the FH MB-BRL problem using simulation samples, we propose a combination of robust RL, Monte Carlo tree search (MCTS), and Bayesian inference. We show an idea of reusing previous MCTS samples for Bayesian inference at a leaf node. The proposed approach allows an agent to choose from multiple robust policies at a leaf node. Numerical experiments of a two-dimensional peg-in-hole task demonstrate the effectiveness of the proposed approach.

1 Introduction

Reinforcement learning (RL) is a promising framework for an autonomous robot in an uncertain environment (Sutton and Barto 1998). Since real-world samples in a robotic task are expensive in terms of time and labor, reducing real-world sample is often more important than reducing computational efforts (Kober, Bagnell, and Peters 2013).

Simulation samples obtained with a model can complement real-world samples. Additionally, if a parametric model is given using prior knowledge, then an agent can learn using a considerably fewer real-world samples. A naturally assumed parametric model is a differential equation model (DE-model), e.g. equations of motion. However, if a control policy is learned using an imprecise model, it may not work well for the real robot. When an agent cannot initially identify its sufficiently accurate model, it needs to consider model uncertainty (Kober, Bagnell, and Peters 2013).

As an approach to the issues on real-world samples and model uncertainty, we consider a model-based Bayesian RL (MB-BRL) framework (Duff 2002). We assume that an agent knows a parametric DE-model, while it does not know the exact values of some parameters in the model. In principle, an optimal history-dependent policy for the exploration-exploitation tradeoff in the future can be planned in advance. The MB-BRL framework has the potential to make the best use of both real-world samples and an assumed model structure, while one major issue is its relatively large amount of computation.

We consider a stochastic shortest path (SSP) setting, an undiscounted infinite horizon (IH) problem with a cost-free termination state. Since a continuous-time decision process is intractable, the learning process deals with its discrete-time version, while its control results follow a DE-model.

One important goal in RL is to achieve (near) optimality after sufficient exploration. In MB-BRL, for model identification, Bayesian DP (Strens 2000) and BOSS (Asmuth et al. 2009) drive exploration by random sampling and optimism, respectively. A constrained exploration approach, e.g. safe exploration (Moldovan and Abbeel 2012), also considers (near) optimality in the distant future. Although such an algorithm is appropriate for running a large number of episodes, it may not be suitable for a current episode alone.

Another perspective in RL is to ensure performance under uncertainty instead of (near) optimality. For this purpose, a robust (or risk-aware) method considers a policy valid for a set of models. Robustness is often opposed to exploration, e.g. robust DP (Nilim and El Ghaoui 2005) and BOSS. In a current episode alone, robustness is often reasonable. Although robust replanning (Bertuccelli, Wu, and How 2012) combines robust DP and online model identification, it still cannot choose an explorative behavior to get useful information about model uncertainty for reducing real-world samples in a current episode, e.g. the Listen action in the Tiger problem (Kaelbling, Littman, and Cassandra 1998). The reason is that such exploration is worthless for a stationary policy which is not changed by online identification. Thus, robustness without considering exploration is also insufficient.

A finite horizon (FH) approximation in MB-BRL has the potential for reducing real-world samples, while the major issues are its expensive computational efforts and the need for terminal costs/rewards. Monte Carlo tree search (MCTS) is a promising sample-based approach for a large-scale FH or discounted IH problem. BAMCP (Guez, Silver, and Dayan 2012), an extension of MCTS to MB-BRL,

gives terminal costs by simulation using one rollout policy. BAMCP solves a discounted IH MB-BRL, where the value of the rollout policy starting from a node at a certain depth can be finally negligible within its corresponding precision thanks to the discount factor. However, in case of no discount, BAMCP often cannot solve a SSP MB-BRL.

In the proposed approach, we firstly split an original SSP MB-BRL problem into two simplified problems: the first FH MB-BRL and the remaining SSP robust RL. The former obtains a history-dependent policy for exploration-exploitation tradeoff within the finite depth. The latter considers a stationary policy for performance guarantee. Combining these two easier problems, we deal with a FH MB-BRL problem in which terminal costs are given by robust stationary policies. For any depth, the resulting performance is better than or equal to the performance obtained with a robust method. Thus, an agent can employ the resulting policy, even if the depth is not sufficient for exploration due to computational resource limits. In addition, the resulting policy may choose an explorative behavior within the finite depth.

Secondly, to find a feasible solution of the approximate MB-BRL problem using a DE-model, we propose a combination of three simulation-based approaches: (i) a robust RL method learning a set of stationary policies employed for terminal costs, (ii) a MCTS method planning a history-dependent policy in the FH MB-BRL, and (iii) a new Bayesian inference method approximating a belief at a leaf node by reusing previous MCTS samples. As a result, the proposed combination is a transfer approach through the MB-BRL framework, in that real-world decisions are made by simulation samples. The proposed approach allows an agent to choose from multiple robust policies according to an approximate belief at a leaf node. Thus, it improves terminal costs in a FH approximation. This is especially important for a SSP MB-BRL.

2 Notation and Assumptions

Due to space constraints, for an introduction to a SSP Markov Decision Process (MDP), please refer to (Bertsekas and Tsitsiklis 1996). A Bayes-Adaptive MDP (BAMDP) is an extension of a MDP (Duff 2002). Let $\mathcal{S}$, $\mathcal{U}$, and Θ be sets of spaces, actions, and parameters, respectively. Let $p(s'|\theta, s, u)$ be a discrete-time transition probability model (TP-model), which is the probability of next state s' given current state s and action u in a MDP with uncertain constant parameter θ. We assume that an agent knows exactly not $p(s'|\theta, s, u)$ but its underlying DE-model parameterized by the same θ. Let $g(s, u, s')$ be a known cost function, which we assume to be independent of θ for simplicity. Let $h^t \equiv (b^0, s^0, u^0, \cdots, s^t)$ be a history at timestep t. Let b^t be a belief given h^t. Belief b^t follows the Bayes rule,

$$b^t(\theta) \propto b^0(\theta) \prod_{t'=0}^{t-1} p(s^{t'+1}|\theta, s^{t'}, u^{t'}).$$

The J-factor of policy π for s^t in a MDP with θ is

$$J^\pi_\theta(s^t) \equiv E\left[\sum_{t'=t}^{\infty} g(s^{t'}, u^{t'}, s^{t'+1}) \,\middle|\, \theta, s^t, \pi\right].$$

The J-factor of π for h^t in a BAMDP is

$$J^\pi(h^t) \equiv E\left[J^\pi_\theta(s^t)|b^t, s^t, \pi\right] = \int b^t(\theta) J^\pi_\theta(s^t) d\theta.$$

An optimal policy in a BAMDP depends on h^t or corresponding (b^t, s^t). Similarly, we also define the Q-factor.

If $J^\pi_\theta(s) < \infty$, $\forall s$, then π is proper for a MDP with θ (for short, proper for θ). A proper policy guarantees that an agent finally reaches a termination state from any other state.

3 Approximate MB-BRL Problem

Remaining SSR robust RL After observing h^D at depth D, the remaining problem starts from h^D or corresponding (b^D, s^D). If $J^\pi(b, s) < \infty$, $\forall s$, then π is proper for the support of b. Such a proper policy is a kind of robust controllers, in that it guarantees task achievement for a set of MDPs. To limit the search space, we consider the class of stationary policies. Let π^r_{bs} be a sub-optimal stationary policy for (b, s) that minimizes $J^\pi(b, s)$. For short, $J^r(b, s) \equiv J^{\pi^r_{bs}}(b, s)$. If $(b, s) \neq (b', s')$, then $\pi^r_{bs} \neq \pi^r_{b's'}$. Thus, it is important to choose appropriately a stationary policy for each (b, s). For simplicity of notation, we also use π^r_h and $J^r(h)$.

First FH MB-BRL Given $J^r(h)$ as a terminal cost, the J-factor of π in the first FH MB-BRL with depth D is

$$J^\pi_D(h^t) \equiv E\left[J^r(h^D) + \sum_{t'=t}^{D-1} g(s^{t'}, u^{t'}, s^{t'+1}) \,\middle|\, b^t, s^t, \pi\right].$$

This problem considers all behaviors within depth D, including both explorative behaviors and robust behaviors. Let π^f_D be a sub-optimal history-dependent policy that minimizes $J^\pi_D(h^t)$. Similarly, we here define the Q-factor.

Combination of two problems Let π^p_D be a sub-optimal policy that uses π^f_D until depth D and subsequently switches to $\pi^r_{h^D}$ for observed h^D. In terms of the J-factors in a BAMDP, π^p_D is better than or equal to a robust method. In fact, when $D = 1$,

$$\begin{aligned} J^{\pi^p_1}_1(h^0) &= \min_u E\left[g(s^0, u, s') + J^r(h^0, u, s')|b^0, s^0, u\right] \\ &\leq E\left[g(s^0, u, s') + J^r(h^0, u, s')|b^0, s^0, \pi^r_{h^0}\right] \\ &\leq E\left[g(s^0, u, s') + J^{\pi^r_{h^0}}(h^0, u, s')|b^0, s^0, \pi^r_{h^0}\right]. \end{aligned}$$

The first line is the J-factor of π^p_1, the second the J-factor of robust replanning of π^r_h, and the third the J-factor of $\pi^r_{h^0}$. By induction, the same inequalities hold for $D > 1$.

4 Sample-based Approximation

For simplicity, we assume that discretizations of continuous time and $\mathcal{U}$ are given. We employ discretization grids over $\mathcal{S}$ and Θ for function approximation, while simulation samples are basically obtained using a DE-model. We choose the grid over Θ according to the robustness of a policy, described in

Algorithm 1 Implementation outline

1: Assume a DE-model and choose D.
2: In simulation, construct $\mathcal{M}$ by robust RL.
3: In simulation, learn $\pi^f_D(\mathcal{M})$ by MCTS with Bayesian inference at leaf nodes.
4: In a real environment, use $\pi^p_D(\mathcal{M})$.

Section 5. The grid over $\mathcal{S}$ is based on a standard idea in continuous RL.

Sample-based robust RL and Bayesian inference require significantly more samples than one episode of MCTS. Firstly, since it is unrealistic to learn π^r_h for all possible h, we consider constructing a finite set of stationary policies, $\mathcal{M}$, before MCTS. An agent chooses the best policy in $\mathcal{M}$ for h. In the first FH MB-BRL, we replace $J^r(h)$ with $\min_{\pi\in\mathcal{M}} J^\pi(h)$. Let $\pi^r_h(\mathcal{M})$, $\pi^f_D(\mathcal{M})$, and $\pi^p_D(\mathcal{M})$ be the modified sub-optimal policies. Secondly, we propose a Bayesian inference method reusing previous MCTS samples to approximate a belief at a leaf node in MCTS simulation. The implementation outline is presented in Algorithm 1.

Robust RL The minimum requirement for $\mathcal{M}$ is to include at least one policy proper for Θ. This is necessary for the convergence of the proposed approach. To find a robust deterministic stationary policy, we use a simple combination of Q-learning and parameter sampling from a belief, called robust Q-learning. Although this algorithm has no guarantee to find a policy proper for Θ, it finds such a policy in some domains, including the task in Section 5. After constructing $\mathcal{M}$, we calculate $J^\pi_\theta(s)$ for $\pi\in\mathcal{M}$ and discretized (θ, s).

When robust Q-learning fails to find a policy proper for Θ, we must consider another algorithm or a larger class of policies. A policy proper for Θ is a stochastic combination of policies proper for subsets of parameters such that the sum of the subsets is equal to Θ. However, the performance is usually poor. An acceptable policy proper for Θ and effective construction of $\mathcal{M}$ in a general case are future issues.

MCTS We introduce the UCT (Kocsis and Szepesvári 2006) and the parameter sampling from a belief at a root node, as with BAMCP. For Bayesian inference at leaf nodes, we add the visit counter of (θ, h, u, s') based on the grids over $\mathcal{S}$ and Θ, denoted by $v^{\theta hu}_{s'}$. Let v be the set of $v^{\theta hu}_{s'}$. Instead of rollout simulation, we give a terminal cost at a leaf node by $\mathrm{Leaf}(h, v, \mathcal{M})$, described later. This cost function is also used when MCTS simulation reaches depth D.

Bayesian inference Since the likelihood in the Bayes rule, $p(s'|\theta, s, u)$, is not known exactly, we consider estimating it using simulation samples. Based on the grids over $\mathcal{S}$ and Θ, we approximate $p(s'|\theta, s, u)$ and $b^t(\theta)$ by multinomial distributions. Let $\{s_0, s_1, \cdots, s_N\}$ be the set of discrete states. Let c be a condition to specify a part of a history of continuous-states. Let $p^{\theta cu}_j \equiv p(s_j|\theta, c, u)$ be the probability of next discrete-state s_j given (θ, c, u). In general, a more detailed condition improves the accuracy of Bayesian inference, while it becomes more difficult to get samples satisfying the condition. For example, a history of discrete-states is a more detailed condition than a current discrete-state alone. This is because a continuous-state transition is Markovian, while its discretization is not necessarily Markovian. Note that the most detailed condition is a current continuous-state, which is based on the assumed DE-model.

We propose reusing previous MCTS samples. Let $v^{\theta cu}_{s_j}$ be the sum of $v^{\theta hu}_{s_j}$ such that h satisfies c. For short, $p^{\theta cu} \equiv [p^{\theta cu}_j]^N_{j=0}$ and $v^{\theta cu} \equiv [v^{\theta cu}_{s_j}]^N_{j=0}$. Let $e \equiv [e_j]^N_{j=0}$ be a hyperparameter. Given Dirichlet prior $\mathrm{Dir}(p^{\theta cu}|e)$ and sample set $v^{\theta cu}$, the posterior distribution of $p^{\theta cu}$ is $\mathrm{Dir}(p^{\theta cu}|e+v^{\theta cu})$. Then, the posterior mean of $p^{\theta cu}$ is

$$\Pr(s_j|\theta, c, u, v^{\theta cu}) = \frac{e_j + v^{\theta cu}_{s_j}}{\sum^N_{j'=0}(e_{j'} + v^{\theta cu}_{s_{j'}})}.$$

As $|v^{\theta cu}| \to \infty$, the posterior mean converges, if ignoring the approximation errors by the multinomial distribution. Using this estimator, we approximate the Bayes rule by

$$\Pr(\theta|h^t, v) \propto b^0(\theta) \prod_{t'=0}^{t-1} \Pr(s^{t'+1}|\theta, c^{t'}, u^{t'}, v^{\theta c^{t'} u^{t'}}).$$

As $|v^{\theta c^{t'} u^{t'}}| \to \infty$ for all $t' < t$, $\Pr(\theta|h^t, v)$ also converges. Finally, we give the terminal cost function in MCTS by

$$\mathrm{Leaf}(h^t, v, \mathcal{M}) \equiv \min_{\pi\in\mathcal{M}} \sum_\theta \Pr(\theta|h^t, v) J^\pi_\theta(s^t).$$

As $|v| \to \infty$, $\mathrm{Leaf}(h, v, \mathcal{M})$ converges to $\min_{\pi\in\mathcal{M}} J^\pi(h)$ for all h, if $\mathcal{M}$ incudes at least one policy proper for Θ. Here, a transition to a leaf node is a non-stationary multi-armed bandit problem discussed in (Kocsis and Szepesvári 2006). The remaining proof of the convergence of the proposed approach is the same.

Since this terminal cost chooses from multiple policies according to (b, s) explicitly, it is better than or equal to the J-factor of a rollout policy that does not consider b explicitly.

5 Numerical Experiment

Definition We demonstrate the effectiveness of $\pi^p_D(\mathcal{M})$ using the 2-dimensional peg-in-hole task, which captures the essence of an assembly task by a robot. The goal of the task is to insert a square peg into a similar sized hole. As a DE-model, we use the ODE, an open-source simulation library of rigid body dynamics (Smith 2008). To discuss the proposed approach, we think of numerical experimental behaviors of resulting policies as real-world samples.

Due to space constraints, for details about the task definition, please refer to (Senda and Tani 2014). It is also shown that the ODE accurately predicts the results of the hardware experiments. The discrete-time transition is from the time when the peg starts moving to the time when it becomes stationary. Then, the state in a MDP is specified by the peg's position, attitude, force, and torque. In this paper, we use the same state grid, while we change the number of each discretized state variable from [5,5,5,4,3,3] to [10,7,11,3,5,5].

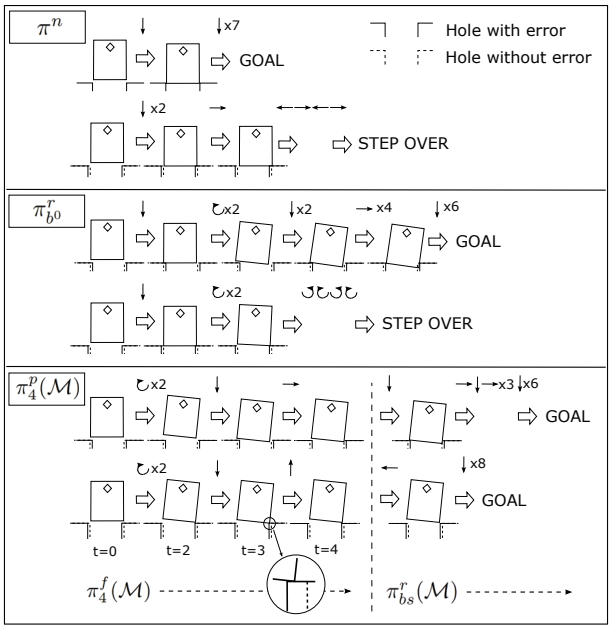

Figure 1: Behaviors of π^n, $\pi^r_{b^0}$, and $\pi^p_4(\mathcal{M})$.

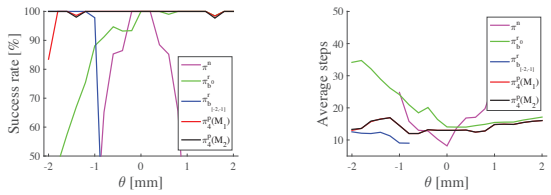

Figure 2: Results for $\theta \in \Theta$.

A discrete action is to move the peg up, down, left, right, clockwise, or counterclockwise. Note that this approximation is used only to simplify the learning process, while the resulting policies are applied to the ODE. The cost function is 1 in all states, except for the termination states. To discourage transitions toward the outside of $\mathcal{S}$, additional cost 10^5 is incurred. In this paper, we assume that the uncertain parameter in the task, θ, is the horizontal hole position error alone. We give initial belief b^0 by the uniform distribution on $\Theta \equiv [-2, 2]$ [mm]. Note that the value of θ is unknown to the agent, fixed in each episode, and changed every episode.

Results Each resulting policy is applied to the ODE and evaluated with 50000 episodes. A success episode is to achieve the task within 50 steps. A failed episode is to exceed 50 steps or to transition outside of $\mathcal{S}$. The average steps are calculated using only success episodes. Each resulting policy is asymmetric due to the small difference of the Q-factors obtained with a sample-based method.

Let π^n be an optimal policy for a MDP with $\theta = 0$ [mm]. As Figure 1 shows, π^n tries to insert the peg without considering the hole error, where the dotted line indicates the hole with $\theta = 0$ [mm]. In Figure 2, the success rate of π^n implies that nearly ± 0.5 [mm] error spoils a policy. Then, we choose $\Delta\theta = 0.5$ [mm] as the grid width over Θ.

Table 1: Total results over Θ.

Policy	Success	Outside	50-step over	Average
π^n	40.5 [%]	8.9 [%]	50.6 [%]	14.4
$\pi^r_{b^0}$	88.4	3.9	7.7	18.7
$\pi^p_2(\mathcal{M})$	98.6	0.0	1.4	14.6
$\pi^p_3(\mathcal{M})$	98.7	0.0	1.3	14.5
$\pi^p_4(\mathcal{M})$	99.4	0.2	0.4	14.1
$\pi^p_4(\mathcal{M})$	99.8	0.2	0.0	14.1

Let π^r_b be a policy learned by robust Q-learning, where θ is sampled from given b, and s^0 is randomly initialized. Although $\pi^r_{b^0}$ is more robust than π^n, it is unacceptable for Θ yet. Let $b_{[-2,-1]}$ be a certain distribution over $[-2, -1]$ [mm]. In Figure 2, the success rate of $\pi^r_{b_{[-2,-1]}}$ for Θ is worse than $\pi^r_{b^0}$, while the average steps of $\pi^r_{b_{[-2,-1]}}$ for $\theta \in [-2, -1]$ [mm] are better. This result shows the importance of choosing a stationary policy for each (b, s).

Firstly, we discuss the results of $\pi^p_D(\mathcal{M}_1)$, where $\mathcal{M}_1$ includes $\pi^r_{b^0}$ and its mirrored policy with respect to the center line of the hole with $\theta = 0$ [mm]. For Bayesian inference at leaf nodes, we give condition c by a history of discrete-states. At $t = 0$, for all θ, all of the discrete-state transitions are deterministic due to the initial distance between the peg and hole. Since b^1 is not substantially updated, $\pi^r_{b^0}$ and $\pi^p_1(\mathcal{M}_1)$ are the same. This result shows that $\pi^p_D(\mathcal{M})$ with insufficiently deep D has robust performance at worst. Table 1 shows that the total results of $\pi^p_D(\mathcal{M}_1)$ are improved as D increases. These results demonstrate that the proposed approch is better than or equal to a robust method alone.

Secondly, we compare $\pi^p_4(\mathcal{M}_1)$ and $\pi^p_4(\mathcal{M}_2)$, where $\mathcal{M}_2$ includes $\pi^r_{b^0}$, $\pi^r_{b_{[-2,-1]}}$, and their mirrored policies. As Figure 1 shows, the peg's right corner is in contact with the hole only if $\theta \in [-2, -1]$ [mm]. Then, b^3 and b^4 are concentrated within $[-2, -1]$ [mm], and $\pi^p_4(\mathcal{M}_2)$ chooses $\pi^r_{b_{[-2,-1]}}$ at (b^4, s^4). In Figure 2, the success rate for $\theta \sim -2$ [mm] is improved (compare the red and black lines). This result shows the effectiveness of Bayesian inference to choose from multiple policies at a leaf node.

6 Conclusion

To reduce real-world samples within feasible computational efforts, we have considered the FH MB-BRL in which terminal costs are given by robust stationary policies, and have proposed the combination of the simulation sample-based approaches: robust RL, MCTS, and Bayesian inference.

For more complex domains, the three sample-based approaches need to be implemented using more advanced techniques. Firstly, a robust extension of DQN (Mnih et al. 2015) seems promising, though we need more discussion. Secondly, combining MCTS with deep neural networks (Silver et al. 2016) is useful. Thirdly, Bayesian inference at leaf nodes reusing MCTS samples needs more extensions, because it is a novel idea. Some interesting techniques are seen in ABC methods (Marin et al. 2012).

7 Acknowledgments

A part of this work has been financially supported by a grant-in-aid for Scientific Research from the Ministry of Education, Science, Culture, and Sports of Japan.

References

Asmuth, J.; Li, L.; Littman, M. L.; Nouri, A.; and Wingate, D. 2009. A bayesian sampling approach to exploration in reinforcement learning. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 19–26.

Bertsekas, D. P., and Tsitsiklis, J. N. 1996. *Neuro-Dynamic Programming*. Athena Scientific, 1st edition.

Bertuccelli, L. F.; Wu, A.; and How, J. P. 2012. Robust adaptive markov decision processes: Planning with model uncertainty. *IEEE Control Systems* 32(5):96–109.

Duff, M. O. 2002. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. Ph.D. Dissertation, University of Massachusetts Amherst.

Guez, A.; Silver, D.; and Dayan, P. 2012. Efficient bayes-adaptive reinforcement learning using sample-based search. In *Advances in Neural Information Processing Systems*, 1025–1033.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1):99–134.

Kober, J.; Bagnell, J. A.; and Peters, J. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32(11):1238–1274.

Kocsis, L., and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *European Conference on Machine Learning*, 282–293.

Marin, J.-M.; Pudlo, P.; Robert, C. P.; and Ryder, R. J. 2012. Approximate bayesian computational methods. *Statistics and Computing* 1–14.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529.

Moldovan, T. M., and Abbeel, P. 2012. Safe exploration in markov decision processes. In *Proceedings of the 29th International Conference on Machine Learning*, 1711–1718.

Nilim, A., and El Ghaoui, L. 2005. Robust control of markov decision processes with uncertain transition matrices. *Operations Research* 53(5):780–798.

Senda, K., and Tani, Y. 2014. Autonomous robust skill generation using reinforcement learning with plant variation. *Advances in Mechanical Engineering* 6:276264.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489.

Smith, R. 2008. Open dynamics engine. Available at http://www.ode.org/.

Strens, M. 2000. A Bayesian framework for reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, 943–950.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.