Alternating Optimisation and Quadrature for Robust Reinforcement Learning

Supratik Paul
Department of Computer Science
University of Oxford
supratik.paul@cs.ox.ac.uk

Kamil Ciosek
Department of Computer Science
University of Oxford
kamil.ciosek@cs.ox.ac.uk

Michael A. Osborne
Department of Engineering Science
University of Oxford
mosb@robots.ox.ac.uk

Shimon Whiteson
Department of Computer Science
University of Oxford
shimon.whiteson@cs.ox.ac.uk

Abstract

Bayesian optimisation has been successfully applied to a variety of reinforcement learning problems. However, the traditional approach for learning optimal policies in simulators does not utilise the opportunity to improve learning by adjusting certain environment variables – state features that are randomly determined by the environment in a physical setting but are controllable in a simulator. This paper considers the problem of finding an optimal policy while taking into account the impact of environment variables. We present the alternating optimisation and quadrature algorithm which uses Bayesian optimisation and Bayesian quadrature to address such settings and is robust to the presence of significant rare events, which may not be observable under random sampling but have a considerable impact on determining the optimal policy. Our experimental results show that our approach learns better and faster than existing methods.

1 Introduction

A key consideration when applying reinforcement learning (RL) to a physical setting is the risk and expense of running trials, for example while learning the optimal policy for a robot. To address this, learning is often conducted in simulators. Although this is cheaper and safer than running physical trials, the computational cost of conducting each simulated trial can still be quite high. The challenge then is to develop algorithms that are sample efficient, i.e., that minimise the number of such trials. In such settings, Bayesian optimisation (BO) (Brochu et al., 2010) is an appealing approach because it is highly sample efficient and has been successfully applied to RL in multiple domains (Lizotte et al., 2007, Martinez-Cantin et al., 2007, 2009, Cully et al., 2015, Calandra et al., 2015).

However, the traditional approach to BO does not take advantage of an opportunity afforded by simulators: the ability to adjust certain environment variables, state features that cannot be controlled in a physical setting but are (randomly) determined by the environment. For example, when learning to fly a helicopter under different wind conditions (Koppejan and Whiteson, 2011), we typically cannot control the wind in physical trials but can easily do so in a simulator.

A naïve approach would be to simply randomly sample values for these environment variables in each trial, so as to estimate expected performance. However, this approach is not robust to significant rare events (SREs), i.e., it fails any time there are rare events that substantially affect expected performance. For example, some rare wind conditions may increase the risk of crashing the helicopter. Since crashes are so catastrophic, avoiding them is key to maximising expected performance, even
though the wind conditions contributing to the crash occur only rarely. In such cases, the naïve approach will not see such rare events often enough to learn an appropriate response.

In this paper, we propose a new approach called ALternating Optimisation and Quadrature (ALOQ) that is specifically aimed towards learning policies that are robust to these rare events. The main idea is to construct a Gaussian Process (GP) that models performance as a function of both the policy and the environment variables and then, at each timestep, to use BO and Bayesian quadrature (BQ) in turn to select a policy and environment setting, respectively, to evaluate.

We apply ALOQ to a number of test problems, as well as simulated navigation and robot control problems. Our results demonstrate that ALOQ learns better and faster than multiple baselines, including a naïve approach, ALOQ with certain components ablated, and an existing GP-based method for coping with environment variables.

2 Related Work

[Frank et al.] also consider the problems posed by SREs in RL. In particular, they propose an approach based on importance sampling for efficiently evaluating policies whose expected value may be substantially affected by rare events. While their approach is based on temporal difference (TD) methods, we take a BO-based policy search approach. Unlike TD methods, BO is well suited to settings in which sample efficiency is paramount and/or where assumptions (e.g., the Markov property) that underly TD methods, cannot be verified. BO has had empirical success in such settings (Lizotte et al., 2007; Martinez-Cantin et al., 2007, 2009; Cully et al., 2015; Calandra et al., 2015).

More importantly, [Frank et al.] assume prior knowledge of the SREs, such that they can directly alter the probability of such events during policy evaluation. By contrast, a key strength of ALOQ is that it requires only that a set of environment variables can be controlled in the simulator, without assuming any prior knowledge about whether SREs exist, or about the settings of the environment variables that might trigger them.

[Williams et al.] consider a problem setting they call the design of computer experiments which is essentially identical to our setting. They also propose an approach that alternates between BO and BQ. However, their approach, which we discuss further in Section 3, is applicable only to discrete environment variables whereas ALOQ can handle both discrete and continuous ones. Furthermore, our experiments show that ALOQ is faster computationally; more robust to SREs; and, unlike the approach of Williams et al., outperforms a baseline that randomly samples the environment variable.

Finally, [Krause and Ong] also consider the problem of optimising performance in the presence of environment variables. However, whereas in our setting we optimise performance after marginalising out the environment variable, they address a contextual bandit setting in which the learned policy conditions on the environment variable.

3 Background

Given a (possibly noisy) black-box objective function \( f(x) : \mathbb{R}^d \rightarrow \mathbb{R} \), we assume that it has been drawn from a GP prior. A GP is completely specified by a mean function \( m(x) \) and a (positive definite) covariance function \( k(x, x') \) and has the property that any finite set of points can be expressed as a multivariate Gaussian distribution. The distribution of any point, given a set of observed points, can be expressed in closed form due to the conditional and marginalisation properties of the Gaussian distribution. For an introduction to GPs, see [Rasmussen and Williams, 2005].

The prior mean function of the GP is often assumed to be 0 for convenience. A popular choice for the covariance function is the class of stationary functions of the form \( k(x, x') = k(x - x') \), which may have some hyperparameters \( \zeta \). Observed data points \( D = \{ x_i, f(x_i) \}_{i=1}^N \) are used to update the posterior belief about the objective function through the posterior distribution \( p(f | D, \zeta) \propto p(f | f \theta) p(D | f, \zeta) \). We follow a full Bayesian approach and compute the marginalised posterior distribution \( p(f | D) \) by first placing a hyperprior distribution on \( \zeta \) and then marginalising it out from \( p(f | D, \zeta) \). In practice, an analytical solution for this is unlikely to exist so we estimate \( \int p(f | D, \zeta) p(\zeta | D) d\zeta \) using Monte Carlo quadrature, described next. For ease of notation, we drop \( \zeta \) from the conditioning set in the rest of the paper.
Monte Carlo quadrature estimates integrals of the form $\bar{f} = \int f(x)p(x)dx$, where $p(x)$ is the probability density of $x$. We simply sample $(x_1, x_2, ..., x_N)$ from $p(x)$ and estimate the integral as:

$$\bar{f} \approx \frac{1}{N} \sum_{i=1}^{N} f(x_i).$$

(1)

$N$ needs to be sufficiently large to ensure convergence to the true value of the integral, and hence this method should only be used if $\bar{f}$ is cheap to evaluate.

BO (O’Hagan [1991] Rasmussen and Ghahramani [2003] is an alternative to Monte Carlo quadrature for settings where $f$ is computationally expensive to evaluate. BO begins by taking a GP prior for $f(x)$, and then uses evaluations of the integrand (at selected nodes) to compute a posterior for $f$. This posterior can then be used to compute a univariate Gaussian posterior for the integral, $\bar{f}$, whose mean and variance can be computed analytically for particular choices of GP covariance and prior $p(x)$. If no analytical solution exists, we can approximate the mean and variance using Monte Carlo quadrature by drawing samples from the posterior for $f$.

BO addresses the problem of optimising $f(x)$ within some compact set $A$, i.e. finding $x^*$:

$$x^* = \arg\max_{x \in A \subset \mathbb{R}^d} f(x).$$

(2)

BO works by selecting for evaluation at each timestep the $x$ that maximises an acquisition function such as expected improvement (EI) (Mockus [1975] Jones et al. [1998]) or upper confidence bound (UCB) (Cox and John [1992] 1997). Defining $x^+$ as the current optimal evaluation, i.e., $x^+ = \arg\max_x, f(x)$, EI seeks to maximise the expected improvement over the current optimum:

$$\alpha_{EI}(x) = \mathbb{E}[f(x)] = \mathbb{E}[\max\{0, f(x) - f(x^+)\}]$$

(3)

By contrast, UCB does not depend on $x^+$ but still acknowledges the uncertainty in our estimate while computing the potential for improvement at any point:

$$\alpha_{UCB}(x) = \mu(x) + k \sigma(x),$$

(4)

where $\mu$ and $\sigma^2$ are the mean and variance, and $k$ controls the exploration-exploitation tradeoff.

4 Problem Setting & Method

We assume access to a computationally expensive simulator that takes as input a policy $\pi \in A$ and environment variable $\theta \in B$ and produces as output $f(\pi, \theta) \in \mathcal{R}$, where both $A$ and $B$ belong to some compact sets in $\mathbb{R}^d_A$ and $\mathbb{R}^d_B$, respectively. We also assume that $p(\theta)$, the probability distribution over $\theta$, is known. Defining $f_i = f(\pi_i, \theta_i)$, we assume an initial dataset $D_{1:t} = \{(\pi_1, \theta_1, f_1), (\pi_2, \theta_2, f_2), ..., (\pi_t, \theta_t, f_t)\}$. Our objective is to find an optimal policy $\pi^*$:

$$\pi^* = \arg\max_{\pi} \hat{f}(\pi) = \arg\max_{\pi} \mathbb{E}[f(\pi, \theta)].$$

(5)

Let us first consider a naïve approach that disregards $\theta$, applies BO directly to $\hat{f}(\pi) = f(\pi, \theta)$, with only one input $\pi$, and attempts to estimate $\pi^*$. Formally, this approach models $\hat{f}$ as a GP with a zero mean function and a suitable covariance function $k(\pi, \pi')$. For any given $\pi$, the variation in $\hat{f}$ due to different settings of $\theta$ is treated as observation noise. To estimate $\pi^*$, the approach applies BO, while sampling $\theta$ from $p(\theta)$ at each timestep. This naïve approach will almost surely fail, as it is unlikely to sample SREs often enough to learn a suitable response.

A better approach is to model $f(\pi, \theta)$, acknowledging both its inputs, as a GP: $f \sim GP(m, k)$. At timestep $t + 1$, we then simultaneously select $\pi_{t+1}$ and $\theta_{t+1}$: $(\pi_{t+1}, \theta_{t+1}) = \arg\max_{\pi, \theta} \alpha_{SO}(\pi, \theta)$. Here $\alpha_{SO}$ is a novel acquisition function, based on EI, that measures the expected improvement from the given $(\pi, \theta)$, while marginalising out $\theta$:

$$\alpha_{SO}(\pi, \theta) = \mathbb{E}_{f(\pi, \theta) \mid D_{1:t}} \left[ \max \left\{ 0, \hat{f}(\pi) \right\} \mid D_{1:t+1} - \hat{f}(\pi^+) \mid D_{1:t} \right],$$

(6)

Most notably, the posterior for the integral is closed form for combinations of a squared exponential covariance function and a prior that is Gaussian or a mixture of Gaussians. Many other combinations of covariance and prior also yield closed-form posteriors (Briol et al. 2015).
where $\pi^+$ is the best $\pi$ found up until timestep $t$. Unlike (5), the $\max$ operator together with a stochastic $f(\pi^+) \mid D_{1:t}$ makes $\alpha_{SO}(\pi, \theta)$ analytically intractable. We could approximate it using Monte Carlo sampling but the computational cost, increasing with $t$, makes this impractical.

Instead of simultaneously selecting $\pi$ and $\theta$, another approach is to select them in alternating fashion: first select $\pi$ using a BO acquisition function on $\bar{f}(\pi) \mid D$, then select $\theta$ using a quadrature acquisition function that conditions on the selected $\pi$. This is the approach taken by Williams et al. (2000). The BO acquisition function of their algorithm, which we refer to as WSN, is based on EI:

$$\alpha_{WSN}(\pi) = \mathbb{E}_{\pi, \theta}(\bar{f}(\pi) \mid D_{1:t}) = \mathbb{E}_{\pi}\left[\pi_{t+1} \mid D_{1:t}\right] = \max\{0, \bar{f}(\pi) - f(\pi^+) \mid D_{1:t}\}. \quad (7)$$

$\alpha_{WSN}$ cannot be computed analytically since $f(\pi^+) \mid D_{1:t}$ is a random variable. This is addressed by applying the identity $\mathbb{E}(I(\pi) \mid D_{1:t}) = \mathbb{E}[(I(\pi) \mid D_{1:t}, \bar{f}(\pi^+))]$ and using Monte Carlo sampling to approximate the inner expectation. This requires computing the joint distribution of $\{f(\pi_1), f(\pi_2), ..., f(\pi_t) \mid D_{1:t}\}$, which in turn requires performing predictions on $t \times N_\theta$ points, i.e., all the $t$ observed $\pi$’s paired with each of the $N_\theta$ support point for the environment variable. This is prohibitively expensive in practice, even for moderate $t$, as the computational complexity of GP predictions scales quadratically with the number of predictions. Moreover, as we highlight later in this section, WSN with a stationary covariance function, as the authors propose, is unsuited to modelling SREs as it cannot capture the different length scales that characterise such events. Finally, the formulation of WSN is such that it can only be applied to settings where $\theta$ is discrete and $f(\pi) \mid D = \sum_\theta p(\theta)f(\pi, \theta) \mid D$.

We propose a new, simpler, and more effective alternating scheme called Alternating Optimisation and Quadrature (ALOQ). Without loss of generality, we assume that the prior mean function is 0 and use a suitable stationary kernel as the covariance function $k$. Our estimate of $\pi^*$ is thus:

$$\hat{\pi}^* = \arg\max_\pi \mathbb{E}_{\pi}(f(\pi) \mid D_{1:t}). \quad (8)$$

Unlike WSN, we make no restrictive assumptions about $\theta$. While for discrete $\theta$ the estimate for $f(\pi) \mid D$ is straightforward, for continuous $\theta$ we apply Monte Carlo quadrature. Although this requires sampling a large number of $\theta$ and evaluating the corresponding $f(\pi, \theta) \mid D$, it is feasible since we evaluate $f(\pi, \theta) \mid D$ not from the expensive simulator, but from the computationally cheap GP.

As noted earlier, using the EI acquisition function is not practical for ALOQ. However, the UCB acquisition function is a natural choice since it bypasses the challenges with estimating $f(\pi^+) \mid D$ at each timestep. Our acquisition function for $\pi$ is thus:

$$\alpha_{ALOQ}(\pi) = \mu(\bar{f}(\pi) \mid D) + \kappa\sigma(\bar{f}(\pi) \mid D), \quad (9)$$

and at timestep $t + 1$ we set $\pi_{t+1} = \arg\max_\pi \alpha_{ALOQ}(\pi) \mid D_{1:t}$. Once $\pi_{t+1}$ has been selected, we follow Osborne et al. (2012) and seek to minimise the posterior variance of $\bar{f}(\pi_{t+1})$ to select $\theta_{t+1}$:

$$\theta_{t+1} = \arg\min_\theta \mathcal{V}(\bar{f}(\pi_{t+1}) \mid D_{1:t}, \pi_{t+1}, \theta). \quad (10)$$

Although the approach described so far actively selects $\pi$ and $\theta$ through BO and BQ, it is unlikely to perform well in practice. A key observation is that the presence of SREs, which we hope ALOQ will address, implies that the scale of $f$ varies considerably, e.g., from dangerous wind conditions to normal ones. This nonstationarity cannot be modelled with our stationary kernel. Therefore, to make ALOQ useful, we must transform the inputs to ensure stationarity of $f$. To do so, we transform the inputs along both $\pi$ and $\theta$ using Beta CDFs with parameters $(\alpha, \beta)$, as in Snoek et al. (2013).

Algorithm[1] presents an overview of ALOQ. $\zeta$ denotes the combined set of hyperparameters of $k$ and the parameters of the Beta CDFs. The corresponding hyperpriors are denoted by $p(\zeta)$.

5 Experimental Results

To evaluate ALOQ, we applied it to three types of problems: 1) artificial test functions, including those used by Williams et al. (2000), 2) a simulated navigation task, and 3) a simulated robot arm control task.

For each problem, we compare ALOQ to several baselines: 1) the naïve method described in Section 4, 2) the WSN method; 3) random quadrature, which is like ALOQ but samples $\theta$ randomly from
These are six and four dimensional problems, respectively, with two of the dimensions treated as discrete environmental variables with a total of 12 and 49 support points, respectively. Figures 1a and 1b plot $E_{\theta}[f(\hat{\pi}^*, \theta)]$ for the $\hat{\pi}^*$ estimated by each algorithm at each timestep. In Hartmann 6, there is no substantial performance difference between ALOQ, WSN, and random quadrature, and even the naive approach eventually approaches the optimum. In Branin, ALOQ, random quadrature, and unwarped ALOQ all perform substantially better than WSN.

Figures 1c and 1d plot the per-step runtime of each algorithm, i.e., the time taken to process one data point. On both problems, WSN takes significantly longer than ALOQ or the other baselines, and shows a clear increasing trend. The sudden reduction in time near the end is a computational artefact due to cores and memory being freed up as some runs finish faster than others.

The slow runtime of WSN is as expected (see Section 5.1). However, its failure to outperform random quadrature and the naive method is surprising, as these are the test problems [Williams et al. 2000] use in their own evaluation. However, they never compared WSN to these (or any other) baselines. Consequently, they never validated the benefit of modeling $\theta$ explicitly, much less selecting it actively. In retrospect, these results make sense because neither test problem is characterised by significant rare events and so there is no a priori reason to predict that simpler methods will fail.

Consequently, these results underscore the fact that a meaningful evaluation requires a problem with significant rare events, as only such problems require more robust methods. To create such an evaluation, we formulated two test functions, F-SRE1 and F-SRE2, that are characterised by significant rare events. Figures 2a and 2b show contour plots of these functions, both of which have narrow bands of $\theta$ in which the scale of the rewards is much larger (to make the plots more readable, we downscale the regions corresponding to significant rare events by a factor of 10).

Figures 2c and 2d, which plot the performance of all methods on these two test problems, show that ALOQ substantially outperforms all the other algorithms. The final learned policy, i.e., $\hat{\pi}^*$, of each algorithm is shown as a vertical line in Figures 2a and 2b, along with $\pi^*$ (the true maximum). These lines illustrate that properly accounting for significant rare events can lead to learning qualitatively different policies.

5.1 Artificial Test Functions

We begin with the Hartmann 6 and Branin test functions used by Williams et al. [2000] to evaluate WSN. These are six and four dimensional problems, respectively, with two of the dimensions treated as discrete environmental variables with a total of 12 and 49 support points, respectively. Figures 1a and 1b plot $E_{\theta}[f(\hat{\pi}^*, \theta)]$ for the $\hat{\pi}^*$ estimated by each algorithm at each timestep. In Hartmann 6, there is no substantial performance difference between ALOQ, WSN, and random quadrature, and even the naive approach eventually approaches the optimum. In Branin, ALOQ, random quadrature, and unwarped ALOQ all perform substantially better than WSN.

$p(\theta)$ instead of choosing it actively; and 4) unwarped ALOQ, which does not perform Beta warping of the inputs. All results plotted below, except for the simulated navigation task, are the median of the three sets of experiments.

### Algorithm 1 ALOQ

**Input** An expensive black box function $f(\pi, \theta)$, an initial set of data points $D_{1:L}$, the number of function evaluations $L$, and an appropriate GP prior with hyperparameters $\zeta$ and corresponding hyperpriors $p(\zeta)$.

1. **for** $n = l + 1, l + 2, ..., L$ **do**
2. Draw a random sample $\{\zeta_1, \zeta_2, ..., \zeta_z\}$ from $p(\zeta|D_{1:n-1})$ using a Monte Carlo method
3. Compute the marginalised posterior distribution:
   \[
   p(f|D_{1:n-1}) = \int p(f|D_{1:n-1}, \zeta)p(\zeta|D_{1:n-1}) \approx \frac{1}{z} \sum_{i=1}^{z} p(f|D_{1:n-1}, \zeta_i)
   \]
4. Use Monte Carlo quadrature to estimate the mean and variance of $p(f|D_{1:n-1})$
5. Using the marginalised posterior distribution, select $\pi_n = \text{argmax}_{\pi} \alpha \text{ALOQ}(\tilde{f}(\pi)|D_{1:n-1})$
6. Using the marginalised posterior distribution and $\pi_n$, select:
   \[
   \theta_n|\pi_n = \text{argmin}_{\theta} \sigma^2(\tilde{f}(\pi_n)|D_{1:n-1}, \pi_n, \theta)
   \]
7. Evaluate $f_n = f(\pi_n, \theta_n)$ and set $D_{1:n} = D_{1:n-1} \cup (\pi_n, \theta_n, f_n)$
8. **end for**

**Output** $\pi^* = \text{argmax}_{\pi} \tilde{f}(\pi_i) \mid D_{1:L}, i = 1, 2, ..., L$
In particular, ALOQ is close to $\pi^*$ in both cases and, while random quadrature is in the neighbourhood of $\pi^*$ in F-SRE1, it is completely misled in F-SRE2. This is because in F-SRE1 observing a limited number of SREs through random sampling guides it towards the true optimum without being unduly influenced by the rest of the surface, as $f(\pi)$ excluding SREs is quite flat. In F-SRE2, the maximum of $f(\pi)$ excluding SREs is located at the minimum of the full function and observing only a few SREs is unable to guide the search away from the minimum. In both cases, WSN is far from $\pi^*$.

Figures 2e and 2f plot in log-scale the per-step runtime of each algorithm on F-SRE1 and F-SRE2. As before, WSN is far slower than the other methods. In fact, for F-SRE2, it was not computationally feasible to run WSN beyond 75 data points.

5.2 Simulated Navigation Problem

To evaluate ALOQ on a more realistic and challenging problem, we used the V-REP simulator (Rohmer et al., 2013) to simulate a helicopter flying from a starting point to a predefined goal with the objective of reaching it in the shortest possible time. While the shortest path is a straight line, the task is complicated by the presence of stochastic crosswinds with an approximately 2% chance of blowing the helicopter onto an obstacle parallel to its path. The cost of a crash is high enough to warrant avoiding it even though the probability of such an event is low. The optimal course is to go leeward of the obstacle such that the effect of the crosswind is completely negated, while passing as close to the obstacle as possible to reduce the time taken.

Since the time taken to run a simulated trial on V-REP is relatively high, we could only perform five independent runs for each algorithm. For the same reason, it is impractical to numerically determine the expected cost of any given path. Hence, any path with a chance of leading to a crash is assumed to be suboptimal and assigned a cost of 150. Due to its poor performance on the test functions, we did not test the naïve approach in this setting. We also could not apply WSN here because the wind velocity, which follows an exponential distribution, is continuous.

Figure 3a shows a screenshot of a simulation in progress. The helicopter flies to the target, denoted by a black dot, and can either follow a path in the dangerous section between the obstacle and the source of the wind (as it does in this case by following the dark blue line), or follow a longer but safer path that goes around the obstacle, an example of which is the light blue line. The path is determined
Figure 2: Contour plot of the SRE functions (values in SRE regions have been reduced by a factor of 10) as well as performance and runtime (note the log scale) of all methods on these problems.

by sliding the two points, denoted by the green and yellow spheres, along the axis parallel to the source of the wind and the obstacle. It is possible to set a path that collides with the obstacle. Thus, the learning algorithm must learn to avoid both the obstacle and the dangerous region.

Figure 3b shows the cost (lower is better) of the path selected after each timestep for each method. After an initial exploration phase, ALOQ settles on a safe path, while random quadrature is misled into taking a dangerous one, and unwarped ALOQ does not converge to any specific path.

5.3 Robotic Arm Simulator

For our final experiment, we evaluated ALOQ’s performance on a robot control problem implemented in a kinematic simulator. The goal is to configure each of the three controllable joints of a robot arm such that the tip of the arm gets as close as possible to a predefined target point. In the first setting, the arm is placed in front of a wall whose distance away is a stochastic environment variable. At some distances, some joint angles yield a collision with the wall, which incurs a large cost. Minimising cost entails getting as close to the target as possible, without passing through the region where the wall may be present.
Figure 3: Visualisation, performance, and runtime of each method on the simulated navigation problem.

Figure 4a shows the expected cost (lower is better) of the arm configurations after each timestep for each method. ALOQ and random quadrature greatly outperform the other methods, including WSN. Furthermore, ALOQ substantially outperforms random quadrature, achieving a final cost that is approximately three times lower.

Figure 4b shows the learned arm configurations, as well as the policy that would be learned by ALOQ if there was no wall (No Wall). The shaded region represents the possible locations of the wall. Note that this extends well beyond the origin but is not shown in the plot. This plot illustrates that ALOQ learns a qualitatively different policy than random quadrature, with the former much closer to the target location. Furthermore, while all the algorithms learn to avoid the wall, active selection of $\theta$ allows ALOQ to do so more quickly: smart quadrature allows it to more efficiently observe rare events and accurately estimate their boundary.

Figure 4: Performance and learned configurations of each method on the robot simulator.

6 Conclusions

This paper proposed ALOQ, a novel approach to using Bayesian optimisation and quadrature to perform sample-efficient reinforcement learning in a way that is robust to the presence of significant rare events. We empirically evaluated ALOQ on several test functions, a simulated navigation problem, and a robotic arm simulator. Our results demonstrated that ALOQ significantly outperforms the WSN algorithm, which tries to address a similar problem. Further, ALOQ is far more computationally efficient and is also able to handle continuous environment variables.

From a theoretical perspective, ALOQ does not take into account the predicted mean, but only the predicted variance when selecting the next environment variable. This is a consequence of using a GP prior with the variance minimisation objective. Although we have shown that this works well, we believe that an acquisition function that explicitly takes into account the mean could further improve performance and we aim to explore this in future work.
In addition, we hope to develop an extension to ALOQ that is robust to errors in estimating \( p(\theta) \). Doing so could ease the transition of policies learned in simulation to the real world, a famously difficult challenge. We hope to test such an extension on real robots.

**Acknowledgements**

We would like to thank Jean-Baptiste Mouret and Federico Allocati for providing the robotic arm simulator used in the experiments, and Tom Rainforth and Yannis M. Assael for the many helpful discussions. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement #637713).

**References**

Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., and Sejdinovic, D. (2015). Probabilistic Integration: A Role for Statisticians in Numerical Analysis? *arXiv:1512.00933*.

Brochu, E., Cora, V. M., and Freitas, N. D. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Calandra, R., Seyfarth, A., Peters, J., and Deisenroth, M. (2015). Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*.

Cox, D. D. and John, S. (1992). A statistical method for global optimization. In *Systems, Man and Cybernetics, 1992., IEEE International Conference on*.

Cox, D. D. and John, S. (1997). Sdo: A statistical method for global optimization. In *Multidisciplinary Design Optimization: State-of-the-Art*.

Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, 521.

Frank, J., Mannor, S., and Precup, D. (2008). Reinforcement learning in the presence of rare events. In *Proceedings of the 25th international conference on Machine learning*. ACM.

Jones, D., Schonlau, M., and Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4).

Koppejan, R. and Whiteson, S. (2011). Neuroevolutionary reinforcement learning for generalized control of simulated helicopters. *Evolutionary Intelligence*, 4.

Krause, A. and Ong, C. S. (2011). Contextual gaussian process bandit optimization. In *Advances in Neural Information Processing Systems*.

Lizotte, D. J., Wang, T., Bowling, M., and Schuurmans, D. (2007). Automatic gait optimization with gaussian process regression. In *IJCAI*.

Martinez-Cantin, R., de Freitas, N., Brochu, E., Castellanos, J., and Doucet, A. (2009). A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, 27(2).

Martinez-Cantin, R., de Freitas, N., Doucet, A., and Castellanos, J. (2007). Active policy learning for robot planning and exploration under uncertainty. In *Robotics: Science and Systems*.

Močkus, J. (1975). On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*.

O’Hagan, A. (1991). Bayes–hermite quadrature. *Journal of Statistical Planning and Inference*, 29(3).

Osborne, M., Garnett, R., Ghahramani, Z., Duvenaud, D. K., Roberts, S. J., and Rasmussen, C. E. (2012). Active learning of model evidence using bayesian quadrature. In *Advances in Neural Information Processing Systems*.
Rasmussen, C. E. and Ghahramani, Z. (2003). Bayesian monte carlo. *Advances in neural information processing systems*, 15.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Rohmer, E., Singh, S. P. N., and Freese, M. (2013). V-rep: a versatile and scalable robot simulation framework. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*.

Snoek, J., Swersky, K., Zemel, R. S., and Adams, R. P. (2013). Input warping for bayesian optimization of non-stationary functions. *Advances in Neural Information Processing Systems Workshop on Bayesian Optimization*.

Williams, B. J., Santner, T. J., and Notz, W. I. (2000). Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica*, 10(4).