Environmentally Adaptive Control Policies for Expensive Systems

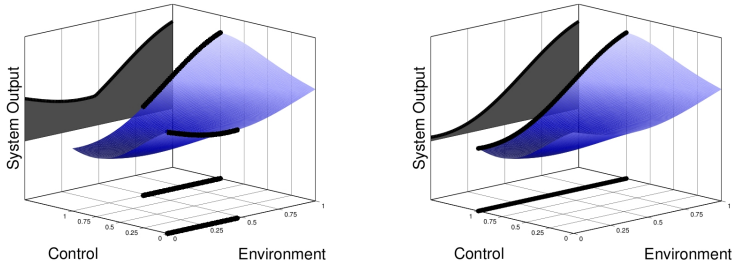
Matthew Tesch‡, Jeff Schneider, and Howie Choset

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213
‡ *mtesch@cs.cmu.edu*

Running experiments on physical robotic systems typically requires significant resources (personnel, time, and money), restricting the number that can be run during training. Additionally, these systems typically operate over a range of external conditions and can be highly non-linear. We focus on the problem of learning a globally optimal policy to adapt controllers for such systems based on the value of these external conditions in order to maximize expected performance. We propose two novel myopic search methods for this problem, and present results comparing these algorithms with various other approaches. Finally, we use these methods to train both simulated and physical snake robots to automatically adapt to changing terrain.

Many robotic systems are effectively a "black-box" — there is no known analytic process model or expression for performance, and it is infeasible to sample the control space densely enough to obtain such a model. In particular, we focus on systems for which evaluation of a single control parameter may take significant effort. Careful *experiment selection* is therefore necessary in order to minimize the number of evaluations required for selection of globally optimal control parameters.

Furthermore, during typical operation of such systems, the environment can change significantly. For example, a locomoting snake robot may move over gently up-sloped terrain, traverse a slightly bumpy horizontal area, and move downhill through many large obstacles. Consider a parameterized controller which can be tuned for each of these environments; an example of such a controller would be a set of gait parameters which close the loop on actuator position at a low level. Obviously, the same set of gait parameters would not be optimal method in each environment; rather one would expect the parameters for best locomotive performance to vary based on the environment. In this work, we seek to develop experiment selection



(a) (b)

Fig. 1: We are interested in problems for which the optimal control parameter changes significantly depending on the environmental conditions. Here we show a system objective function for a toy problem with 1-D environment and control spaces; the policy is shown below the function and the policy score is projected to the left. An optimal adaptive policy is illustrated in (a), presenting the best control parameter for every environment parameter. The constant policy shown in (b) results in a significantly lower overall score.

methods which enable intelligent generation of *control policies* which adapt to changes in the environment by selecting the best controller parameters given a particular environment (Fig. 1). This control policy then acts to close the loop at a high level (typically a relatively slow timescale), changing the parameters of the low-level (typically fast timescale) controller as significant changes in the environment are detected.

Related Work

A key idea in this work is the use of a *surrogate* to represent a function which is expensive to evaluate, and basing search methods on this cheap model. These ideas have been extensively explored in the global optimization community to minimize the number of evaluations required to globally maximize expensive functions.[1] Often these methods rely on stochastic processes to create a surrogate function;[2–4] following this lead we use Gaussian processes[5] as a function approximation method that generates an estimate of expected experiment output along with a measure of confidence in that estimate.

In these cases where function evaluations are costly (hours to days), careful sample choice is extremely important in order to best use the experimental budget. A number of heuristics and statistical methods have been derived to use information from the surrogate function to best choose sample locations (Jones[1] provides a survey of many existing methods). These metrics include the upper confidence bound of the predicted function,[6,7]

the probability of improvement,[8,9] and the expected improvement.[10,11] In particular, expected improvement has been shown to effectively trade off exploration and exploitation without requiring algorithm parameters to be carefully tuned. However, existing expensive global optimization methods are not directly applicable to the addition of environment parameters.

Perhaps the most closely related work is that of contextual bandits,[12] an extension of the multi-armed bandit literature. The choice of which arm to pull (analogously, which experiment to run) in the contextual setting is informed by a context (the environment parameter in our work). The contextual bandit work presents a framework for learning an adaptive arm-selection policy, but assumes an online setting where the context is given to the algorithm; the algorithm must then minimize continuous regret over a series of experiments. In our work, we are interested in efficient training of a controller in an offline setting, where we can set the context (environment parameters) as well as the arm (control parameters); here we wish to minimize absolute expected regret over all contexts. Note that this learned controller could then be used later in an online setting.

Problem Definition

This work aims to find a control policy that optimally adapts to external variables, while minimizing the number of experiments done to perform the optimization. In order to formalize this problem, we define several terms and then more precisely state the goal of this work.

Definition 1 (Control Parameter). *The control parameter space X_c is a compact subset of $\mathbb{R}^{m_c}$. Each $x_c \in X_c$ represents a particular value of the system of interest that can be fully specified during normal operation. Some examples of control parameters include the value of a set of gains in a PID controller, the relative concentration of two reactants in an industrial process, or the prescribed dosage of a drug during drug development.*

Definition 2 (Environment Parameter). *The environment parameter space X_e is a compact subset of $\mathbb{R}^{m_e}$. This space contrasts with the control space in that values $x_e \in X_e$ cannot be controlled under normal real-world operation, but can be specified in laboratory trials. Furthermore, the value of x_e can be measured during normal operation. Therefore, these parameters represent continuous valued external factors of the system, such as terrain steepness for a locomoting system, wind strength and direction for a UAV, particulate size in an industrial process, or disease strain during drug development.*

Definition 3 (System Output). *The system output, denoted as* f*:* $X_e \times X_c \to \mathbb{R}$*, is a continuous, real-valued function of the environment and control parameters. This function represents the performance of the system, given some environmental conditions and some specified control parameter. Example system outputs include the speed of a locomoting system over a terrain, the efficiency of a mechanical process, or the turbulance of a wing design calculated from a wind tunnel or computational fluid dynamics experiment. For the methods we propose here, we assume that sampling this system output is time intensive or computationally expensive, and therefore there is a limit to the number of times this function can be evaluated.*

Definition 4 (Control Policy). *The control policy defined in this work is a mapping* $\gamma\colon X_e \to X_c$ *(not necessarily surjective), such that* $\gamma(x_e)$ *represents the control parameter set by* γ *in reaction to sensing environment parameter* x_e*.*

The policy score,

$$\mathcal{S}(\gamma) = \int_{X_e} \omega(x_e) f(x_e, \gamma(x_e))\, dx_e, \tag{1}$$

represents the expected performance of a policy for a given environment distribtion $\omega\colon X_e \to \mathbb{R}$*.*

The optimal policy γ^* is defined as $\operatorname{argmax} \mathcal{S}(\gamma)$. In Fig. 1, γ^* is shown projected onto the control-environment plane, and $\mathcal{S}(\gamma^*)$ is visualized on the system output-environment plane. Note that γ^* is independent of ω, because $\hat{\gamma}^*(x_e)$ can be independently determined for each $x_e \in X_e$. Although the weighting function can make a significant difference during experiment selection and policy comparison, the optimal policy is unaffected by the relative importance of different environments.

The goal is to find the highest scoring policy γ after a number of system output evaluations. In other words, the quantity of interest is not a single point, but a mapping from environment to control parameters. We break this problem into the subproblems of *policy generation* and *experiment selection*:

(1) **Policy Generation:** Given the results of n system output evaluations, choose the best estimate for γ^*.
(2) **Experiment Selection:** Choose the sequence of points $X = \{x^1, x^2, \ldots x^n\}$, $x^i \in X_e \times X_c$, where the choice of x^{k+1} is informed by $\{f(x^i) \mid i \leq k\}$, which maximizes the score of the policy produced by the chosen policy generation algorithm.

Proposed Methods

One of the difficulties in solving these problems lies in the fact that the true objective function we wish to maximize is $\mathcal{S}$, but as this involves an integral over the expensive f it is impossible to calculate for any γ, let alone optimize via standard optimization techniques. By using Gaussian process regression to find a surrogate $\hat{f}$ for f, we obtain the probability density function $p_x(y)$ of the predictive distribution at each $x \in X_e \times X_c$. These distributions allow us to calculate statistically meaningful quantities for a potential sample, such as the approximate expected improvement over the current predicted optimal policy's score if that experiment were run.

Policy Generation

For the policy generation problem, we can use $\hat{f}$ to select the policy which maximizes our best prediction of the score. In low dimensions, $\hat{\gamma}^*$ can be estimated via a dense sampling of the (relatively) cheap surrogate $\hat{f}$; in higher dimensions one might perform global optimization (e.g., with a simulated annealing or genetic algorithm) on a chosen policy parameterization, using $\hat{f}$ to cheaply evaluate policies. As this work mainly focuses on experiment selection methods, we assume a policy generation method based on maximization of (1) via dense sampling (replacing f with $\hat{f}$).

Experiment Selection

Standard local and global optimization methods directly applied to optimizing system output perform poorly. This is because these methods are not optimizing the true objective; rather they focus on improving knowledge in environments with high system output results, and ignore environments with low system output results. This causes the resulting policy to be very weak in "difficult" environments (ones with low system output results), lowering the overall score. However, we have included results from the popular global search algorithm EGO[11] for comparison. At each iteration, this algorithm selects the parameters for the next experiment as those which maximize the expected improvement over the best previous experiment.

Our first proposed algorithm for experiment selection adapts the idea of expected improvement to the explicit separation of X_e and X_c. Instead of measuring improvement over the best evaluation of f so far, we consider the improvement over the maximum predicted value of f for the *same environment* as the test point. This gives the *unbiased expected improvement* (UEI) of a point $x^t = (x_e^t, x_c^t)$:

$$\mathrm{UEI}(x^t) = \omega(x_e^t) \int_{y^*_{x_e^t}}^{\infty} (y - y^*_{x_e^t}) p_x^t(y)\, dy, \text{ where } y^*_{x_e^t} = \max_{x_c \in X_c} (\hat{f}(x_c, x_e^t)) \quad (2)$$

UEI reduces the bias of standard expected improvement towards good environments. However, UEI only considers improvement of the policy score at one environment. To measure true expected improvement of the policy score from sampling a point x^t, the expectation must be computed over the predictive distribution $p_x^t(y)$, where each potential y involves regression of a new surrogate $\hat{f}_y$ conditioned on the addition of this potential sampled value. This second proposed method is termed the *expected policy score improvement*, or EPSI:

$$\mathrm{EPSI}(x^t) = \int_{-\infty}^{\infty} p_x^t(y) \int_{X_e} \omega(x_e) \max\left(\hat{f}_y(x_e, \gamma_y^*(x_e)) - \hat{f}(x_e, \gamma^*(x_e)), 0\right) dx_e\, dy \quad (3)$$

Using this method generates a more complete estimate of the effect of sampling a point on the policy score. Of course, computing a large numeric integral can also take significantly more time, and the quality of the solution can vary based on the resolution of the integral.

Experimental Results and Concluding Remarks

We evaluated the performance of the two proposed algorithms on analytic test functions as well as on simulated and physical snake robots. The three test functions of varying complexity were created specifically so that a static control policy would not be effective over the entire space.

To measure the score of a particular algorithm, we non-deterministically generate an initial set of k points through guaranteed-coverage sampling method, such as a latin hypercube,[13] and then sequentially select $n - k$ more points, evaluating the system output after each choice. The predicted $\hat{f}$ is used after each evaluation to generate a policy, which is scored via a numeric integral on system output. This entire process was repeated 10 times for each algorithm, the results averaged, and standard errors calculated to provide a rigorous comparison of methods. The algorithms were written in MATLAB, and used the open-source Gaussian Processes for Machine Learning package provided by Rasmussen and Williams.[14]

A summary of these results is shown in Fig. 2; more thorough comparisons[15] are omitted for brevity. Standard global optimization methods initially have good performance, but the policy score tends to stagnate

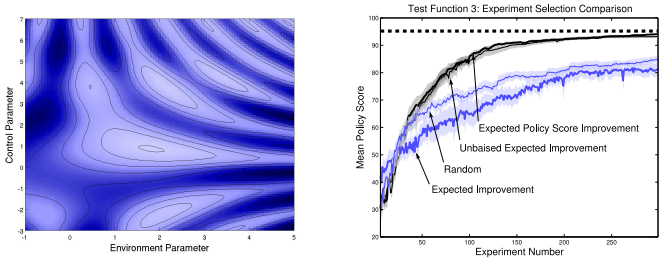


Fig. 2: **(Left):** One of the three analytic test functions used for algorithm comparison. **(Right):** Scores of policies learned for this test function (average over 10 trials); the dotted black line represents the best possible policy score for that function. Policies were generated and scored after each new experiment (x-axis represents experiment number).

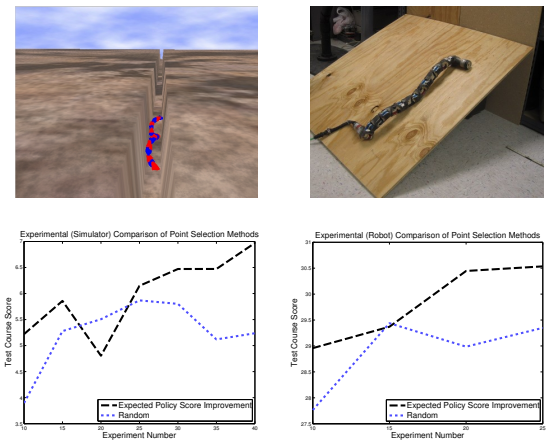


Fig. 3: **(Top):** Experimental setup; system performance in a measure of locomotive energy efficiency for a snake robot crawling through a crevice (simulated robottests) and climbing up an incline (physical robot tests). **(Bottom):** Performance of policies generated from points selected randomly versus using EPSI for the robot tests.

quickly because these methods are optimizing f rather than $\mathcal{S}$. Random point selection also performs suboptimally, showing that it is important to carefully select experiments. UEI and EPSI both perform a better global search; the latter outperforms the former only slightly, indicating that UEI provides a simpler and quicker method which produces similar results.

As such a complete analysis could not be run on physical systems due to the expensive nature of system output evaluation and the inability to

compute a true policy score, we instead set up a range of environmental conditions in a "test course", and then used the above algorithms to generate policies which were scored on this test course. These policies map environment parameters (slope and crevise width) into a 2-D gait parameter control space.[16] The EPSI algorithm was compared to random point selection; the results are shown in Fig. 3.

Although demonstrated here on snake robots, this framework is applicable to a rich set of problems both within and outside the field of robotics. We have described two approaches for experiment selection, proposed a simple policy generation method, and demonstrated the efficacy of these algorithms on analytic test functions and a physical snake robot. Future work involves application to other systems, restricted environment selection during training, and derivation of theoretical performance bounds.

References

1. D. R. Jones, *Journal of Global Optimization* **21**, 345 (2001).
2. R. G. Regis and C. A. Shoemaker, *INFORMS Journal on Computing* **19** (2007).
3. H.-M. Gutmann, *Journal of Global Optimization* **19** (1999).
4. K. Holmström, *Journal of Global Optimization* **41** (2008).
5. C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, 2006).
6. A. W. Moore and J. Schneider, *Advances in Neural Information Processing Systems* , 1066 (1996).
7. D. Cox and S. John, A statistical method for global optimization, in *1992 IEEE International Conference on Systems, Man, and Cybernetics*, (Ieee, 1992).
8. H. J. Kushner, *Journal of Basic Engineering* **86**, 97 (1964).
9. A. Žilinskas, *Journal of Global Optimization* **2**, 145(June 1992).
10. J. Mockus, V. Tiesis and A. Zilinskas, *Towards Global Optimization* **2**, 117 (1978).
11. D. R. Jones, M. Schonlau and W. J. Welch, *Journal of Global Optimization* **13** (1998).
12. T. Lu, D. Pál and M. Pál, *13th International Conference on Artificial Intelligence and Statistics* (2010).
13. M. D. McKay, R. J. Beckman and W. J. Conover, *Technometrics* **21**, 239 (1979).
14. C. E. Rasmussen and C. K. I. Williams, Gaussian Processes for Machine Learning.
15. M. Tesch, J. Schneider and H. Choset, Adapting Control Policies for Expensive Systems to Changing Environments, in *International Conference on Intelligent Robots and Systems*, (IEEE/RSJ, San Francisco, 2011).
16. M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz and H. Choset, *Advanced Robotics* **23**, 1131(June 2009).