Using Response Surfaces and Expected Improvement to Optimize Snake Robot Gait Parameters

Matthew Tesch, Jeff Schneider, and Howie Choset

***Abstract*— Several categories of optimization problems suffer from expensive objective function evaluation, driving the need for smart selection of subsequent experiments. One such category of problems involves physical robotic systems, which often require significant time, effort, and monetary expenditure in order to run tests. To assist in the selection of the next experiment, there has been a focus on the idea of *response surfaces* in recent years. These surfaces interpolate the existing data and provide a measure of confidence in their error, serving as a low-fidelity surrogate function that can be used to more intelligently choose the next experiment. In this paper, we robustly implement a previous algorithm based on the response surface methodology with an expected improvement criteria. We apply this technique to optimize open-loop gait parameters for snake robots, and demonstrate improved locomotive capabilities.**

I. Introduction

Running an experiment on a robotic system can take significant time, effort, and money. Unfortunately some experiments, such as those involving complex environmental interactions, cannot be simulated offline with a high degree of fidelity and must be run on the robot to acquire meaningful results. If one is to optimize the performance of such a robotic system, it is important to choose each experimental evaluation carefully so as to minimize the number of trials required. Optimization techniques have been developed that carefully analyze the previous trials in order to generate the "best" next choice of experiment. These techniques strive for global, not local, optimization of the objective function in a small number of trials. Although proof of convergence to a global optimum is not always possible, a global search is still preferable to a local solution with stronger theoretical guarantees.

Problems involving expensive cost function evaluations are not limited to robotics. The aerospace industry often makes use of computational fluid dynamics simulations that can take hours to obtain a single data point [1]. The benefits of careful consideration of the next experiment far outweigh those of fast experiment selection. Another application involving careful experiment selection is pharmaceutical development [2], as it is desirous to reduce the number of human trials required. These ideas are closely related to active learning, where labeling a data point is an expensive operation and the goal is to build a good classifier with minimal training points via intelligent selection of these points, rather than optimization.

The system we focus our efforts on in this paper is a snake robot (Fig. 1), the most recent iteration of the mechanism presented in [3]. A set of expressive and useful open-loop locomotive motions, termed *gaits*, have been developed for this robot, based on a parameterized model [4]. Previously, these gaits were optimized by hand-tuning parameters to improve performance. This can lead to suboptimal solutions, and tends to not fully explore the parameter space. We desire to use optimization techniques to modify gait parameters to globally improve the effectiveness (here we focus on average speed) of these gaits, and thereby enhance the capabilities of the robot. Furthermore, we wish to optimize these gaits in a more autonomous fashion, without relying on human intuition.

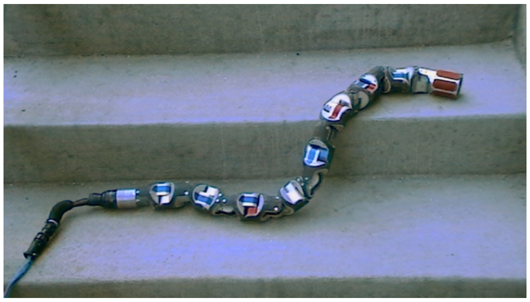


Fig. 1: A hyper-redundant snake robot, shown on an outdoor staircase.

Because of the robots highly dynamic and multimodal locomotion we have been unable to develop a quantitatively accurate model of the robot, and must rely on physical robot experiments to obtain useful results. However, many traditional learning methods depend on gradient information in order to operate, which is not generated via these physical experiments. This "black-box" nature of the objective function limits the choice of optimization method.

A survey of previous work provides promising approaches to optimizing a single expensive scalar objective function over an n-dimensional parameter space. Techniques such as the Nelder-Mead simplex search [5] or subgradient methods [6] have been used extensively, but do not use information from all the previously sampled data points to make the best possible function evaluation decisions. Therefore, these and other similar methods are not well suited to the expensive nature of our cost function evaluations.

A different approach is taken in the *response surface* literature [7]. These methods aim to explicitly use all of the information available to fit a surrogate for the actual objective function. Based on the previously sampled points, the best estimate of the true function is returned, along with the estimated error in this approximation. Various techniques have been used to generate response surfaces; stochastic processes are often the chosen tool [8], [9], [10]. In this paper, we focus on Gaussian process techniques (see [11] for a more complete treatment) in fitting response surfaces for the performance of snake robot gaits.

The response surface literature is concerned with how to use information provided by the response surface in order to select parameters for subsequent experiments, as summarized by Jones [12]. This selection process inherently involves

Matthew Tesch, Jeff Schneider, and Howie Choset are with the Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA. This work was supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program.

a trade-off between exploitation, a local search near the predicted maximum, and exploration, a global search to reduce uncertainty. Various approaches have been developed to balance this tradeoff, such as IEMAX [13], or maximizing the probability of improvement [14], [15]; however, these require careful hand-tuning to ensure good algorithm performance. Other approaches combine discrete stages of local and global search, and recently have shown success in improving the walking speed of a small humanoid robot [16]. In this paper, we instead choose to follow the lead of Jones et al.'s EGO algorithm [17], and maximize the expected improvement over the best experiment conducted so far. This automatically balances the trade-off between exploration and exploitation, informed by a rigorous statistical measure.

In this paper, we extend the previous usage cases of the EGO algorithm by demonstrating its first application that we are aware of to a physical robotic system. In doing so, we have discovered new motions that extend the capabilities of existing snake robots to locomote over obstacles and terrain that were previously impassable. We also show that the EGO algorithm can use an appropriate choice of response surface parameters to successfully handle noisy objective function evaluations. Finally, we propose the use of a combination of unsupervised techniques to improve the quality of the response surface fit.

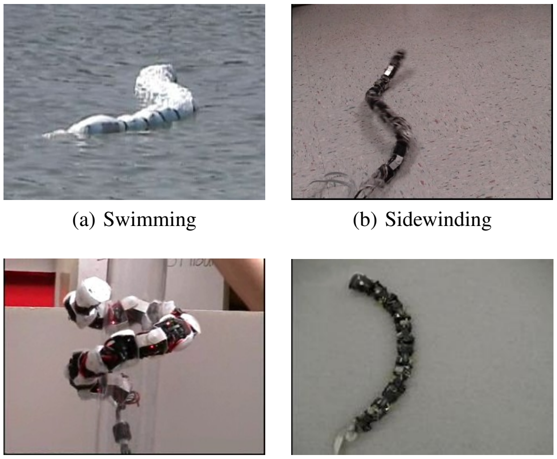
II. The snake robot system

Snake robots (Fig. 1) are hyper-redundant mechanisms which use internal shape changes to move through their environment. The snake robots developed in our lab are composed of a serial chain of actuators with 180° range of motion. The rotation axes are offset 90° about the central axis from module to module. These robots feature many degrees of freedom (usually we use 16 modules) and a small diameter of two inches. The benefits of this design include the ability to crawl through small, tightly packed spaces and to locomote through different environments with a varied set of gaits. However, many degrees of freedom require complex and coordinated controllers in order to generate useful motion. Therefore, previous work of Tesch et al. [4] has focused on developing a useful and general parameterized gait model for the snake. This open-loop controller determines the desired joint angles as a function of module number n and time t. The equation is given by

$$\alpha(n,t) \quad = \quad \begin{cases} \beta_{even} + A_{even}\sin(\frac{d\theta}{dn}n + \frac{d\theta}{dt}t), n = \text{even}, \\ \beta_{odd} + A_{odd}\sin(\frac{d\theta}{dn}n + \frac{d\theta}{dt}t + \delta), n = \text{odd}, \end{cases} \tag{1}$$

where the various constants (such as β, δ, and $\frac{d\theta}{dn}$) are parameters that are modified in order to vary the robot locomotion. These parameters are what we optimize over in order to generate new motions and capabilities. This seven parameter gait model is surprisingly general, and can produce such varied motions as sidewinding, rolling, swimming, and even climbing up a pole (Fig. 2).

Unfortunately, the gaits that have been found by hand-tuning these parameters are still limited in their capabilities. We are interested in increasing the speed of flat-ground locomotive gaits, improving climbing up sloped surfaces, and enabling crawling over unstructured terrain cluttered with obstacles. Using machine learning techniques to conduct a global search allows the discovery of new, less intuitive gaits that might not have been realized by hand-tuning the parameters. Although human intuition is useful, removing this inherent bias in experiment selection and replacing it with a more principled approach allows the discovery of previously unimagined solutions.


(a) Swimming

(b) Sidewinding

(c) Climbing a pole

(d) Rolling laterally

Fig. 2: The generality of the gait model is shown by the varied motions it can produce.

III. Gaussian processes for function regression

Gaussian processes (GPs) are used for many applications; they can form a probability distribution over functions, interpolate between known values, or fit a function to unknown points. In this paper, we are interested in using GPs to produce a *response surface*, or an estimate of our objective function (locomotion performance) as a function of the gait model parameters.

Each Gaussian process is associated with a covariance function which describes the general behavior of the true function we wish to model. Many covariance functions are parameterized by *hyper-parameters* that can be tuned to better reflect the nature of the true function. In addition to fitting various smooth and periodic functions, covariance functions can also be used to model noisy observations by adding a Gaussian noise term to the covariance function. Some examples of the effects of different covariance functions on regression are shown in Fig. 3.

Although Gaussian processes can be used to fit a variety of functions, this technique still relies on a reasonable choice of covariance function and a good selection of hyper-parameters for that covariance function. One can judge the quality of a covariance function and its hyper-parameters by calculating the marginal likelihood of the training data given the GP fit, as in [11]. Unfortunately, this likelihood must then be maximized in a sub-problem during each iteration of the overall optimization. This hyperparameter optimization sub-problem often involves a search over a multi-modal likelihood function, with large areas that are essentially flat. This quality of the likelihood function causes simple search techniques, such as conjugate gradient, to often terminate early, or reach an incorrect local optimum. As suboptimal choices for hyper-parameters can lead to extremely poor response surface fits and consequently poor algorithm performance, one of the main contributions made in this paper is outlining a method to robustly select the best covariance function and hyper-parameters. This selection process occurs with minimal human intervention and without any knowledge

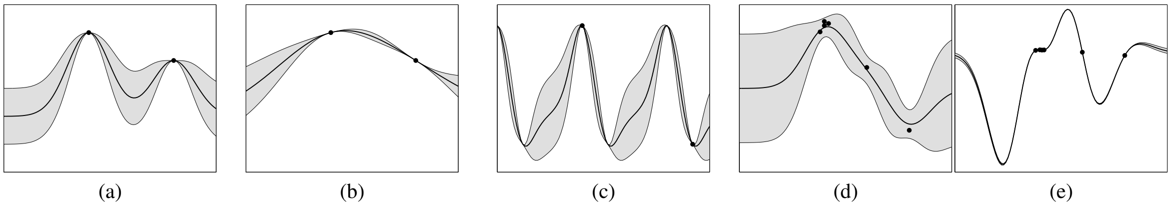


Fig. 3: The first three images show the posterior of a GP (with a one-σ bound) sampled at two data points, with various covariance functions. **(a):** The "squared exponential" covariance function that describes smooth functions, with hyper-parameters set to a medium length scale. **(b):** The same covariance function, with the hyper-parameters adjusted to describe a longer length scale. **(c):** A periodic covariance function. In **(d)** and **(e)**, we show how a covariance function that takes noise into account explicitly, **(d)**, does not overfit noisy data, as does the covariance function which assumes noiseless sampled data points, **(e)**. The no-noise model overfit the data so badly that the range of the function is thirty times that of the noisy model's fit.

of the true function behavior. These improvements allow GP function regression to be reliably and easily used to generate response surfaces that can be used in high-level algorithms.

IV. Experiment selection metrics

Given a reasonable response surface, the question remains of how to use this information in order to best select the next experiment. This choice of experiment involves a tradeoff between exploiting the best known regions and exploring in areas that are largely unknown. At one extreme, an approach would be to choose the current maximum of the response surface; this method is not guaranteed to converge, even to a local optimum. At the opposite extreme, one might seek to maximize the information gain (described in [18]), which serves to reduce the uncertainty in the function estimation. Although this method will eventually find the global optimum through dense sampling of the space, it wastes precious function evaluations on improving the function estimate in low, uninteresting regions of the objective function.

To improve the quality of search, we need to use the estimated function value and the uncertainty of this estimate. To this end, algorithms such as IEMAX [13] have been developed, which pick a next point based on a weighted sum of the estimated function and its error. A more statistically rigorous metric is the *probability of improvement* [14], [15], given by the integral of the tail of the predictive distribution over the maximum value of the objective found so far, plus some threshold. Unfortunately, both of these methods have parameters that must be tuned to obtain the best performance.

The approach taken in this paper is that proposed by Jones et. al. [17]. The metric that is maximized for point selection is the *expected improvement* (EI) over the best previous function evaluation. The response surface provides a distribution Y_x over objective function values at a point in the parameter space x, where $p(y_x)$ represents the probability density of the value $y_x \in Y_x$. Given y_{max} as the best previous function evaluation (e.g., the fastest locomotive speed from previous experiments), the EI can be calculated as

$$EI(x) = \int_{y_{max}}^{\infty} (y_x - y_{max}) p(y_x) dy_x. \qquad (2)$$

The first term in this integral represents the improvement over the previous best sampled point, and the second term represents the probability of that improvement.

The key benefits of this method are that there are no parameters to tune – the tradeoff between exploration and exploitation is inherently handled – and that this metric selects the point which is statistically most advantageous to sample next, at least for a myopic search. When optimizing a function, you desire the next experiment to improve the most upon the current maximum; thus the quantity you actually wish to maximize when selecting a new point is the expected improvement. Furthermore, expected improvement has been empirically shown to work well and to efficiently optimize unknown black-box functions [17], [19].

V. Proposed algorithm

The algorithm we apply to optimize the robot gaits is based on Jones et. al.'s EGO algorithm [17]. After an initial sampling, we iterate the selection process for each subsequent experiment as follows, until convergence is reached or the budget for experiments is exhausted:

1) Fit a response surface to the previously sampled data points.
2) Search over the response surface for the test point which maximizes expected improvement over the previous best experimental result.
3) Run this experiment.
4) Repeat.

We have identified two problems that occur during implementation of this algorithm on noisy robotic systems. First, hyper-parameters often get trapped in local maxima of the likelihood, producing poor quality surface fits. Second, the naïve computation of the expected improvement for noisy functions is emprically unsuccessful. In our proposed algorithm, we have combined new and existing solutions to these problems, and outlined a framework for incorporating these techniques that reduces the need for human supervision during optimization. Furthermore, we have successfully tested this algorithm on the snake robots in our lab. Below, we discuss the individual techniques we use to overcome these problems, and summarize the proposed algorithm.

A. Initial experiment design and random searches

It is impractical to attempt to fit a useful response surface to a very few data points. Instead, one should choose an initial distribution of points to sample before the response surface is constructed. One common approach for choosing this sampling distributions is selecting a *Latin hypercube* (LH) design [17], [19], [20]. At a high level, points in an LH design are guaranteed to be well distributed along each dimension of the parameter space .When using LH sampling, one must also determine how many initial points are chosen. We have found that although a range of values work comparably well, 5 points seem to be sufficient for a two or three dimensional parameter space.

We propose a large, intensive search for the maximum likelihood covariance function hyper-parameters of the initial response surface. This search is worthwhile, because a quality fit will carry through in the form of better formative experiment selections, and a better initial hyper-parameter setting for future iterations. To conduct this search, we use a Matlab package associated with [11], which implements a conjugate-gradient method to minimize the hyper-parameters' negative log likelihood. This method requires a starting point in hyper-parameter space for this minimization, and as it is a local optimization method it is prone to local minima. To overcome this problem, we propose running a conjugate gradient search for each of many randomly picked initial hyper-parameters, as suggested in [21]. We use 50 random points for searching 3 to 5 dimension hyper-parameter spaces.

After each new data point has been sampled, the hyper-parameters should again be optimized. To avoid a large-scale search, we seed one conjugate gradient search with the hyper-parameters use to fit the previous iteration's response surface, and also choose a small number of random hyper-parameters (5-10) to seed other conjugate gradient searches. These random selections explore the space and increase the chances of escaping local minima. It is not sufficient to only use the previous response surface's hyper-parameters, because these sometimes lie outside the global maxima's basin of attraction in the likelihood function after subsequent data points are added.

B. Multiple covariance functions

Poor quality regression can also be caused by using a covariance function which does not describe the function or is needlessly complex, where complexity refers to the number of hyper-parameters and variety of functions it is able to fit. We propose the selection of a set of covariance functions. When fitting a response surface, the hyper-parameter likelihood is maximized for each covariance function in the set; the function with the highest likelihood is chosen. In our experiments, we found using two covariance functions was usually enough – an isometric squared exponential with process noise (three hyper-parameters), and a squared exponential with variable length scales in each direction and process noise ($d + 2$ hyper-parameters for regression in $\mathbb{R}^d$). We encourage the use of prior knowledge to choose appropriate covariance functions; for example, if the true function is periodic, periodic covariance functions should be added to the set.

This adaptive approach improves upon using a single, user-chosen covariance function through the entire optimization. It eliminates the need for an expert in the loop, and allows the complexity of the covariance function to adapt to that of the data. Finally, as the likelihood of a complex covariance function is distributed over a larger space of hyper-parameters than that of a simple covariance function, this approach selects the simplest adequate model, reducing overfitting concerns.

C. Leave-one-out optimization

Fitting a response surface by maximizing the marginal likelihood of Guassian process hyper-parameters can lead to overfitting the data, especially when the sampled data points were returned from a noisy process. Rather than maximizing the marginal likelihood, we instead maximize the leave-one-out likehood, as described in [11]. This provides robustness to overfitting, and generally increases the quality of fit of the surface.

D. Expected improvement with noisy functions

The variance predicted by the response surface generated by a noisy covariance function is the sum of the variance from the process noise (the explicit noise term) and the variance from the uncertainty of the estimate of the underlying noiseless function. As expected improvement is large in areas with large variance or high function estimates, and the magnitude of the variance is lower bounded everywhere by the process noise variance, an experiment selection bias occurs towards higher areas of the function estimate.

To compensate for this bias, we subtract the process noise from the total variance, leaving only the variance from the uncertainty of the true function estimate. This is empirically found to result in a better value for the expected improvement, and leads to a more balanced, effective search.

E. Resulting implementation

The resulting implementation still contains the basic structure of the original EGO algorithm. However, the improvements that have been made significantly improve the algorithm's performance on physical systems. Our proposed algorithm begins with a space-filling Latin hypercube selection of initial experiments. When these experiments are completed, a large scale leave-one-out likelihood maximization occurs over multiple covariance functions and from many initial hyper-parameters. The results of the search are a covariance function and hyper-parameters that describe a response surface that fits the initial sampled points. After the initial function fit, the following steps are repeated until convergence or until the available number of experiments has been exhausted.

1) Select the parameters which result in the maximum expected improvement. Subtract the variance of any process noise from the response surface error estimate.
2) Run an experiment at the selected set of parameters.
3) Fit a response surface to the updated set of sampled data points. Maximize the leave-one-out likelihood, over a space of multiple covariance functions and a few randomly sampled, locally optimized points in hyper-parameter space.

Using random restarts to improve the chances of finding the global maximum is a common technique taken in the machine learning community. Using a leave-one-out approach is also fairly standard. These both improve the implementation, but alone did not cause this algorithm to perform well for actual robotic systems. The use of a set of covariance functions and the improvement for noisy covariance functions also were important. When these features were added, the algorithm became viable for physical systems.

VI. Results

In order to test our algorithm on the robot, we chose to optimize locomotion speed over various terrain (see Fig. 4). Rather than optimize over the entire gait parameter space, we fixed some of these parameters to obtain a reduced subset of qualitatively similar gaits

Using hand-tuned sidewinding parameters, the robot would tip when attempting to climb up a slope. However, our algorithm quickly found a number of improved solutions. The optimized set of gait parameters, discovered after only 26 trials (Fig. 5(a)), resulted in a swinging motion that served to balance the snake on the slope during the climb

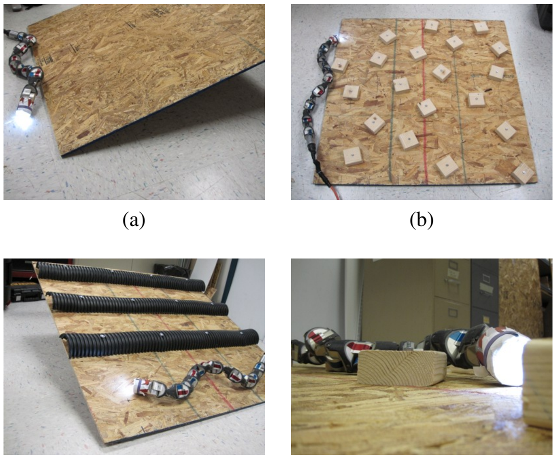
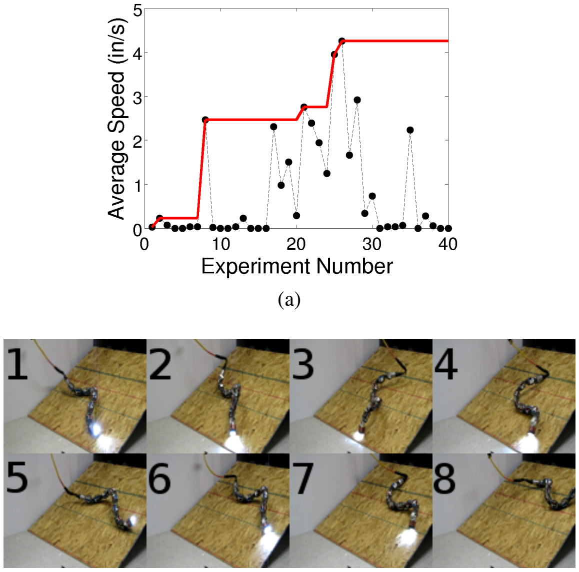


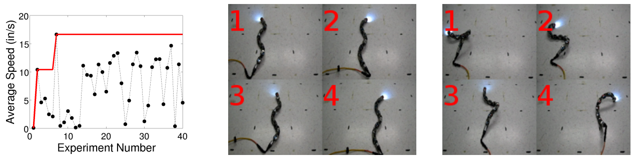
Fig. 4: Various obstacles were built on which to optimize open-loop gait performance. Note that each of these can either be inclined or flat or the ground. **(a):** A simple wood slope. **(b):** Irregularly spaced small square obstacles. **(c):** Three black pipes, split in half along the center. **(d):** Note the scale; the robot, even when lifting its body a moderate amount, cannot pass over the obstacles.



(b)

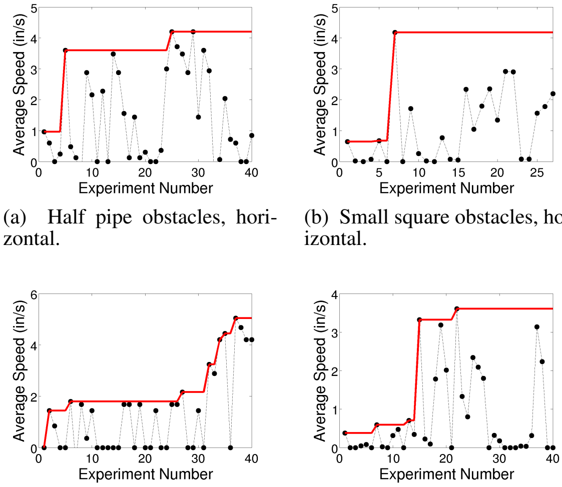
Fig. 5: **(a):** The sampled speed from each experiment conducted during the optimization. The solid line represents the best value found so far, or how many search iterations are required to obtain a certain level of performance, and is a useful visual comparison of results. **(b):** Still shots from the optimized climbing motion of the robot. In order to keep its balance, the robot "swings" from a horizontal position to one aligned with the slope of the incline.

(Fig. 5(b)). This result not only demonstrates the efficacy of the proposed algorithm, but also extended the capabilities of the snake robot. More importantly, the new set of parameters was found without expert knowledge regarding parameter selection; the only human interaction was resetting the snake and measuring the distance traveled.



(a) (b) (c)

Fig. 6: **(a):** The sampled speed from each experiment conducted during the optimization. The solid line represents the best value found so far. The previous best speed was approximately 5 inches per second; the new motion has a speed more than triple this figure. **(b):** The previous hand-tuned sidewinding motion. **(c):** The optimized gait; although it used the same fixed parameter values as the sidewinding gait, it is qualitatively different than the existing sidewinding motion. Note that although this optimized gait has an increased amplitude, simply increasing amplitude from the previous hand-tuned gait will produce a less effective gait; this increase must be coordinated with a change in the other parameters.



(a) Half pipe obstacles, horizontal.

(b) Small square obstacles, horizontal.

(c) Half pipe obstacles, sloped.

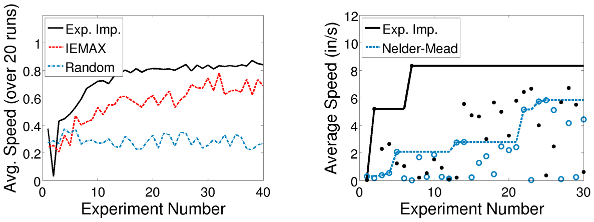
(d) Small square obstacles, sloped.

Fig. 7: Results from optimization tests on two sets of obstacles. Each obstacles was set flat on the ground for a first round of optimization tests; a second set of tests was conducted for each obstacle when set at an angle. For each test, the sampled objective function value is plotted for each experiment conducted, and a solid line indicates the best sampled value so far.

As comparing the newly discovered climbing gait with one hand-tuned for flat ground is misleading, we also optimized the speed of locomotion across flat ground. In this experiment, a significant improvement was found early in the experimental trials. This early success is likely due to the fact that the sidewinding parameter subspace has large regions which have not been adequately explored. When hand tuning a sidewinding motion, the performance usually decreases as an independent parameter is moved outside of a small range. By removing the human bias when searching the parameter space, other areas of this space are visited, significantly altering the motion of the robot.

The motion of the hand-tuned sidewinding gait is depicted in Fig. 6(b), while Fig. 6(c) shows the motion resulting from the newly optimized parameters. The first exciting aspect of this result is the fact that we achieve three times the speed of a gait hand-tuned for the same purpose. The second exciting aspect is that the resulting motion is a previously undiscovered method of locomotion; it is qualitatively different from the existing sidewinding gait. This new motion is also more dynamic than the existing sidewinding gait, relying on momentum rather than solely kinematic forces in order to locomote.

We tested this algorithm on several additional problems; the objective in each case was to obtain the highest average speed across the obstacles shown in Fig. 4. As our previous gaits could not move over the obstacles at all, any success represents a new capability of the snake. The results, shown in Fig. 7, demonstrate that our proposed algorithm finds



(a) (b)

Fig. 8: **(a):** Each of three experiment selection metrics (expected improvement, IEMAX, and random) was used for gait speed optimization of a simulated snake robot. The function evaluations at each iteration were averaged over 20 optimization runs. This averaged value is plotted to compare performance of each algorithm. **(b):** Algorithm performance comparison on the physical robot. Our expected improvement algorithm (solid line) was tested against a Nelder-Mead search (dashed line), which shows a slower rate of improvement.

parameters that successfully locomote the snake in each case. Although the results so far have been based on optimizing a parameter subspace composed of sidewinding gaits, we are not limited to this space and have also optimized various other gaits on the snake robot, as well as an eight dimensional parameter space in simulation.

At first glance, one concern is the "jumpy" nature of the optimization process. It is important to realize that this is expected; even though this algorithm does not attempt to accurately model the entire objective function, it must still sample poor quality regions of the space to be certain of the correct overall trend of the function. A more steady trend is seen in aggregates of multiple trials, such as in Fig. 8(a).

Finally, this algorithm is useless if a simpler algorithm could outperform it. Therefore, we have run extensive comparisons to other algorithms in simulation. These results, such as those in Fig. 8(a), demonstrate that our expected improvement based algorithm outperforms other optimization approaches. Furthermore, we have also compared algorithm performance on the actual robot. In Fig. 8(b), we show that our approach also reaches a maximum faster than a Nelder-Mead search, as our algorithm searches globally rather than constraining itself to a local search.

VII. Conclusion

In this paper, we have shown that maximizing expected improvement using a response surface is an effective tool for choosing subsequent experiments in optimization problems involving real robots. This allows us to optimize existing gaits for snake robots, and endows the robot with new gaits and capabilities. Furthermore, we have discussed modification to the standard expected improvement-based algorithm EGO which improve the algorithm's performance, and allow it to work with noisy functions. In addition, we present optimized results for our existing gait model which allow the robot to move faster than ever before, and climb over terrain which previously would have been an impasse.

Although expected improvement would seem to be the "correct" metric to maximize over a response surface, there are still areas for improvement. For example, if the response surface is not actually a reasonable surrogate, it can choose a series of points that are very slow to converge to a global maximum. Furthermore, the calculation of expected improvement in the presence of noise is not yet fundamentally sound. Although our implementation works well empirically, a rigorous method to measure expected improvement in the presence of noisy samples should be developed. Another area for improvement is to modify the metric to be non-myopic. We wish to find a statistical quantity that makes optimal choices given multiple remaining steps. Unfortunately, non-myopic solutions are often intractable, and so approximate extensions are envisioned.

Finally, we will expand these methods to the field of multi-objective optimization. In particular, we will search more than one objective function, perhaps speed and energy. However, rather than find a single optimum representing some weighted combination of these objectives, we will search for a *pareto set* of points which are all optimal in some sense – they are all *non-dominated* points.

Overall, we believe this work introduces an exciting and promising optimization technique to the robotics field, demonstrating its efficacy on a highly dynamic and multi-modal platform.

References

[1] A. Jameson and J. Vassberg, "Computational fluid dynamics for aerodynamic design: Its current and future impact," *Fluid Dynamics*, vol. 538, 2001.

[2] J. DiMasi, "The price of innovation: new estimates of drug development costs," *Journal of Health Economics*, vol. 22, pp. 151–185, Mar. 2003.

[3] C. Wright, A. Johnson, A. Peck, Z. McCord, A. Naaktgeboren, P. Gianfortoni, M. Gonzalez-Rivero, R. Hatton, and H. Choset, "Design of a modular snake robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2609–2614, IEEE, Oct. 2007.

[4] M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz, and H. Choset, "Parameterized and Scripted Gaits for Modular Snake Robots," *Advanced Robotics*, vol. 23, pp. 1131–1158, June 2009.

[5] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, Jan. 1965.

[6] N. Z. Shor, K. C. Kiwiel, and A. Ruszcayski, *Minimization methods for non-differentiable functions*. New York: Springer-Verlag, 1985.

[7] G. E. P. Box and N. R. Draper, *Empirical model-building and response surfaces*. Wiley, 1987.

[8] R. G. Regis and C. A. Shoemaker, "A Stochastic Radial Basis Function Method for the Global Optimization of Expensive Functions," *INFORMS Journal on Computing*, vol. 19, no. 4, 2007.

[9] H.-M. Gutmann, "A Radial Basis Function Method for Global Optimization," *Journal of Global Optimization*, vol. 19, 1999.

[10] K. Holmström, "An adaptive radial basis algorithm (ARBF) for expensive black-box global optimization," *Journal of Global Optimization*, vol. 41, no. 3, 2008.

[11] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[12] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, 2001.

[13] A. W. Moore and J. Schneider, "Memory-based stochastic optimization," *Advances in Neural Information Processing Systems*, pp. 1066–1072, 1996.

[14] H. J. Kushner, "A new method for locating the maximum point of an arbitrary multipeak curve in the presence of noise.," *Journal of Basic Engineering*, vol. 86, pp. 97–106, 1964.

[15] A. Žilinskas, "A review of statistical models for global optimization," *Journal of Global Optimization*, vol. 2, pp. 145–153, June 1992.

[16] T. Hemker, M. Stelzer, O. von Stryk, and H. Sakamoto, "Efficient Walking Speed Optimization of a Humanoid Robot," *The International Journal of Robotics Research*, vol. 28, pp. 303–314, Feb. 2009.

[17] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, no. 4, 1998.

[18] R. M. Murray, "A Mathematical Introduction to Robotic Manipulation," 1994.

[19] A. Sóbester, S. J. Leary, and A. J. Keane, "On the Design of Optimization Strategies Based on Global Response Surface Approximation Models," *Journal of Global Optimization*, vol. 33, no. 1, pp. 31–59, 2005.

[20] M. D. McKay, R. J. Beckman, and W. J. Conover, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, vol. 21, no. 2, pp. 239 – 245, 1979.

[21] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for regression," *Advances in Neural Information Processing Systems*, 1996.