Learning Flexible and Reusable Locomotion Primitives for a Microrobot

Brian Yang
UC Berkeley

Grant Wang
UC Berkeley

Roberto Calandra
UC Berkeley

Daniel S. Contreras
UC Berkeley

Sergey Levine
UC Berkeley

Kristofer S. J. Pister
UC Berkeley

Abstract

The design of gaits for robot locomotion can be a daunting process which requires significant expert knowledge and engineering. This process is even more challenging for robots that do not have an accurate physical model, such as compliant or micro-scale robots. Data-driven gait optimization provides an automated alternative to analytical gait design. In this paper, we propose a novel approach to efficiently learn a wide range of locomotion tasks with walking robots. This approach formalizes locomotion as a contextual policy search task to collect data, and subsequently uses that data to learn multi-objective locomotion primitives that can be used for planning. As a proof-of-concept we consider a simulated hexapod modeled after a recently developed microrobot, and we thoroughly evaluate the performance of this microrobot on different tasks and gaits. Our results validate the proposed controller and learning scheme on single and multi-objective locomotion tasks. Moreover, the experiments show that without any prior knowledge about the robot used (e.g., dynamics model), our approach is capable of learning locomotion primitives within 250 trials and subsequently using them to successfully navigate through a maze.

1 INTRODUCTION

Substantial progress has been made in recent years towards the development of fully autonomous microrobots [20, 25]. However, the design and implementation of gaits for enabling locomotion at the sub-centimeter scale still remains a non-trivial task. Completing more complicated locomotion tasks like navigating complex environments is even more challenging. These issues become exacerbated when dealing with legged locomotion, where even walking straight is still an active area of study for normal-sized robots. In this paper, we present a novel approach for the autonomous optimization of locomotion primitives and gaits. While locomotion on larger-scale robots has been thoroughly investigated, transferring many of these proven approaches to the millimeter scale poses many unique challenges. One such obstacle is the lack of access to sufficiently accurate simulated models at the millimeter scale. Even simulation environments designed to simulate dynamics at this scale are generally unequipped for usage in robotics contexts. Additionally, working with microrobots can place severe limitations on the number of iterations as trials become much more time-consuming and expensive to run. While microrobot locomotion has been addressed in the past, much of the work is primarily concerned with the mechanical

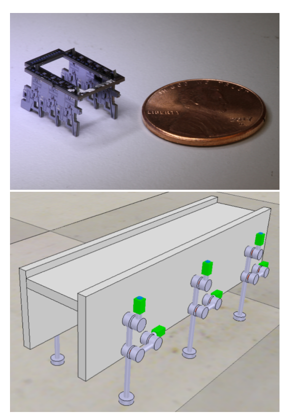


Figure 1: The six-legged micro walker considered in our study (top) and its simulation (bottom).

design and manufacturing of microrobots. Accomplishing more sophisticated locomotion tasks on the sub-centimeter scale remains an open area for research. Analytical implementations of various gait behaviors have worked on microrobots [7, 8], but these solutions can become unwieldy for robots with higher DOF such as legged walkers (e.g., our micro-hexapod). Data-driven automatic gait optimization is a viable alternative to analytical gait design and optimization, but using these techniques can be challenging due to the high number of trials that might be necessary to perform in order to learn viable gaits. Our primary contribution is introducing a novel approach that allows to efficiently learn gaits and motor primitives from scratch without the need for prior knowledge (e.g., a dynamics model). This is accomplished by collecting data on various motor primitives using contextual policy search and using those evaluations to reformulate the problem into a multi-objective optimization task, giving us a model that can map any set of parameters to a predicted trajectory. Using this model, we are able to optimize our parameters on various trajectories for subsequent use in path planning. Our approach is not tied to microrobots only, and can be used for any walking robot. To evaluate our approach, we used a simulated hexapod microrobot modeled after a recently developed microrobot [6]. We first validated existing techniques on a curriculum of progressively more difficult tasks including learning single-objective, contextual, and multi-objective gaits. Then, we evaluated our approach by learning motor primitives from 250 trials and subsequently used them to successfully navigate through a maze.

2 RELATED WORK

While sufficient for simple controllers with few parameters, manually tuning controller parameters can require both an immense amount of domain expertise and time. As such, automatic gait optimization is an important research field which has been studied with a wide variety of approaches in both the single-objective [22, 5, 15, 13, 23, 16, 17, 2] and multi-objective setting [4, 16, 17, 24]. While sufficient for simple controllers with very few parameters, manually tuning controller parameters can require both an immense amount of domain expertise and time. Evolutionary algorithms have been successfully used to train quadrupedal robots [5, 16], but this approach often require thousands of experiments before they can produce good results, which is unfeasible on fragile microrobots.

A more data-efficient approach used in the past to learn gaits for snake and bipedal robots is Bayesian optimization [13, 23, 2]. Bayesian optimization has also been applied to contextual policy search in the context of robot manipulation [14]. Our contribution builds off of this work by applying and extending the contextual framework to learning movement trajectories and path planning. Another extension of Bayesian optimization related to our work is Multi-objective Bayesian optimization, which has also been previously applied in the context of robotic locomotion [24]. However, past work is only concerned with using multi-objective optimization to balance the trade-off between various competing goals. Our main contribution demonstrates an entirely novel application of multi-objective optimization to learning motor primitives that does not involve the trade-off between the various goals, but instead uses a multi-objective model to learn over an area of possible trajectories for path planning.

3 BACKGROUND

Central Pattern Generators Central pattern generators (CPGs) are neural circuits found in nearly all vertebrates, which produce periodic outputs without sensory input [27]. CPGs are also a common choice for designing gaits for robot locomotion [9]. We chose to use CPGs for our controller because they are capable of reproducing a wide variety of different gaits simply by manipulating the relative coupling phase biases between oscillators. This allows us to easily produce a variety of gait patterns without having to manually program those behaviors. This makes them well suited for our microrobot, where the processing power is limited.

One of the foremost benefits of using a CPG controller is a drastic reduction in the number of parameters $\boldsymbol{\theta}_i$ we need to optimize. Overall, the 4 parameters that we consider during the optimization are $\boldsymbol{\theta} = [\omega, R_l, R_r, X]$ where ω is the frequency of the oscillators and X is the phase difference between each of the vertical-horizontal oscillator pairs. In order to allow for directional control, R_l and R_r are the amplitudes of the left and right side oscillators respectively.

Bayesian Optimization Even with a complete CPG network, some amount of parameters tuning is necessary to obtain efficient locomotion. To automate the parameters tuning, we use Bayesian optimization (BO), an approach often used for global optimization of black box functions [12, 10, 2]. We formulate the tuning of the CPG parameters as the optimization

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) , \tag{1}$$

where $\boldsymbol{\theta}$ are the CPG parameters to be optimized w.r.t. the objective function of choice f (e.g., walking speed, which we investigate in Section 5.2). At each iteration, BO learn a model $\tilde{f} : \boldsymbol{\theta} \rightarrow f(\boldsymbol{\theta})$ from the dataset of the previously evaluated parameters and corresponding objective values measured $\mathcal{D} = \{\boldsymbol{\theta}, f(\boldsymbol{\theta})\}$. Subsequently, the learned model $\tilde{f}$ is used to perform a "virtual" optimization through the use of an acquisition function which control the trade-off between exploration and exploitation. Once the model is optimized, the resulting set of parameters $\boldsymbol{\theta}^*$ is finally evaluated on the real system, and together with the corresponding measurement $f(\boldsymbol{\theta}^*)$ is added to the dataset before starting a new iteration. A common model used in BO for the underlying objective, and the one that we consider, are Gaussian processes [19]. For more information regarding BO, we refer the readers to [10, 21].

Multi-objective Bayesian Optimization A special case of the optimization task of Equation (1) is multi-objective optimization [1]. Often times in robotics, there are multiple conflicting objectives that needs to be optimized simultaneously, resulting in design trade-offs (e.g., walking speed vs energy efficiency which we investigate in Section 5.3). When multiple objectives are taken into consideration, there is no longer necessarily a single optimum solution, but rather the goal of the optimization became to find the set of Pareto optimal solutions [18], which also takes the name of Pareto front (PF). Formally, the PF is the set of parameters that are not dominated, where a set of parameters $\boldsymbol{\theta}_1$ is said to dominate $\boldsymbol{\theta}_2$ when

$$\begin{cases} \forall i \in \{1, \ldots, n\} : & f_i(\boldsymbol{\theta}_1) \leq f_i(\boldsymbol{\theta}_2) \\ \exists j \in \{1, \ldots, n\} : & f_j(\boldsymbol{\theta}_1) < f_j(\boldsymbol{\theta}_2) \end{cases} \tag{2}$$

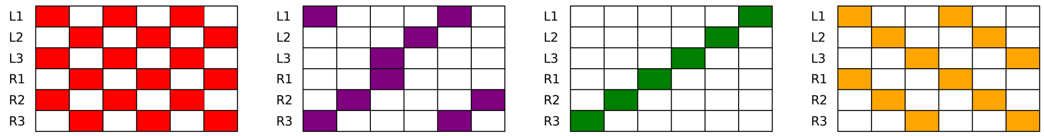
Intuitively, if $\boldsymbol{\theta}_1 \succ \boldsymbol{\theta}_2$, then $\boldsymbol{\theta}_1$ is preferable to $\boldsymbol{\theta}_2$ as it never performs worse, but at least in one objective function it performs strictly better. However, different dominant variables are equivalent in terms of optimality as they represent different trade-offs.

Multi-objective optimization can often be difficult to perform as it might require a significant amount of experiments. This is especially true with our microrobot where large number of experiments can wear-and-tear the robot. As a result, the number of evaluations allowed to find the Pareto set of solutions is limited. Luckily for us, there exist extensions of BO which address multi-objective optimization. In particular, the multi-objective Bayesian optimization algorithm that we consider is ParEGO [11]. The main intuition of ParEGO is that at every iteration, the multiple objectives can be randomly scalarized into a single objective (via the augmented Tchebycheff function), which is subsequently optimized as in the standard Bayesian optimization algorithm (by creating a response surface, and then optimizing its acquisition function). For more information about multi-objective Bayesian optimization we refer the reader to [26].

Contextual Bayesian Optimization Another special case of the optimization task of Equation (1), is contextual optimization. In contextual optimization, we assume that there are multiple correlated, but slightly different, tasks which we want to solve, and that they are identified by a context variable $\boldsymbol{c}$. An example might be walking on inclined slopes, where the contextual variable is the angle of the slope. The contextual optimization can hence be formalized as

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta}, \boldsymbol{c}) , \tag{3}$$

where for each context $\boldsymbol{c}$, a potentially different set of parameters $\boldsymbol{\theta}^*$ exists. The main advantage compared to treating each task independently is that, in contextual optimization, we can exploit the correlation between the tasks to generalize, and as a result quickly learn how to solve a new context. Specifically, in this paper we consider contextual Bayesian optimization (cBO) [14] which extends the classic BO framework from Section 3. Contextual Bayesian Optimization learns a joint model $\tilde{f} : \{\boldsymbol{\theta}, \boldsymbol{c}\} \rightarrow f(\boldsymbol{\theta})$, but now, at every iteration the acquisition function is optimized with a constrained optimization where the context $\boldsymbol{c}$ is provided by the environment. However, because the model jointly model the context-parameter space, experience learned in one context can be generalized to similar contexts. By utilizing cBO, we will show in Section 5 that our microrobot can learn to walk (and generalize) to different environmental contexts such as walking uphill and curving.



(a) Dual Tripod (b) Ripple (c) Wave (d) Four-Two

Figure 2: Contact/swing patterns for different gaits.

4 LEARNING LOCOMOTION PRIMITIVES FOR PATH PLANNING

We now present our novel approach to learn motor primitives for path planning. This approach relies on the possibility of re-using the evaluations collected using cBO to convert the task into a multi-objective optimization problem. We specifically consider a cBO task where we want to optimize the parameters $\boldsymbol{\theta}$ to reach different target positions $\boldsymbol{c} = [\Delta x_{\text{des}}, \Delta y_{\text{des}}]$ (this setting is evaluated in Section 5.6). The objective function in this case can be defined as the Euclidean distance

$$f = \sqrt{(\Delta x_{\text{des}} - \Delta x_{\text{obs}})^2 + (\Delta y_{\text{des}} - \Delta y_{\text{obs}})^2} \tag{4}$$

where $\Delta x_{\text{obs}}, \Delta y_{\text{obs}}$ are the actual positions measured after evaluating a set of parameters. The cBO model would map $\tilde{f} : [\boldsymbol{\theta}, \Delta x_{\text{des}}, \Delta y_{\text{des}}] \to f(\boldsymbol{\theta})$. However, in order to compute f it would need to measure $\Delta x_{\text{obs}}, \Delta y_{\text{obs}}$, effectively generating data of the form

$$[\boldsymbol{\theta}, \Delta x_{\text{des}}, \Delta y_{\text{des}}] \to [\Delta x_{\text{obs}}, \Delta y_{\text{obs}}, f(\boldsymbol{\theta})] \tag{5}$$

We can now re-use the data generated from this contextual optimization to learn a motor primitive model in the form $g : \boldsymbol{\theta} \to [\Delta x_{\text{obs}}, \Delta y_{\text{obs}}]$. The purpose of this learned model g is now to provide an estimate of the final displacement obtained for a set of parameters independently from the optimization process that generated it. Once such a model is learned, we can use it to compute parameters that lead to the desired displacement $\Delta x^*_{\text{obs}}, \Delta y^*_{\text{obs}}$ by optimizing the parameters w.r.t. the output of the model

$$\boldsymbol{\theta}^* = \text{arg max}_{\boldsymbol{\theta}}\, z(g(\boldsymbol{\theta}))\,, \tag{6}$$

where z is a scalarization function of our choice (e.g., the Euclidean distance). This is equivalent to learning a continuous function that generates motor primitives from the desired displacement. It should be noted that this optimization is performed on the model g and therefore does not require any physical interaction with the robot. Moreover, we can optimize the parameters over a series of multiple displacements to obtain a path planning optimization. In Section 5.7, when performing path planning using the learned motor primitives we will employ a simple shooting method optimization which randomly samples multiple candidate parameters and selects the best outcome.

5 EXPERIMENTAL RESULTS

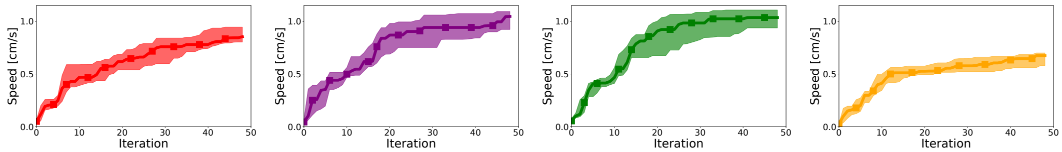
In this section we discuss our controller implementation as well as the performance of our simulated microrobot on various locomotion tasks. Videos of the various locomotion tasks are available at `https://sites.google.com/site/learninglocomotorprimitives/`.

5.1 Controller Implementation

We built our controller following the setup described in Section 3, using a network of 12 coupled phase oscillators (one per motor). In order to translate the output of each of the oscillators into motor actuation, we calculate the oscillator outputs for each vertical-horizontal motor pair using the piecewise function

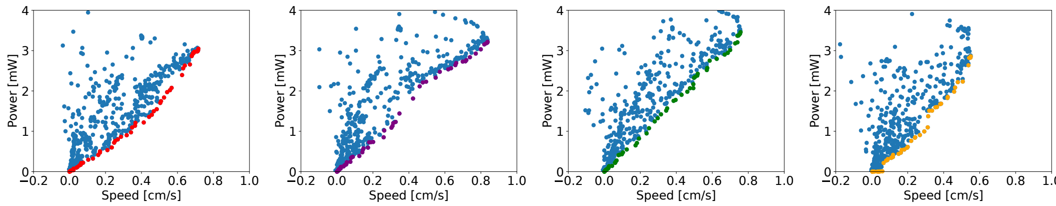
$$\begin{cases} x_i + r_i cos(\phi_i), x_j + r_j cos(\phi_j) & \text{if } \phi_i > \pi, \phi_j > \pi\,, \\ x_i + r_i, x_j + r_j cos(\phi_j) & \text{if } \phi_i \leq \pi, \phi_j > \pi\,, \\ x_i + r_i, x_j + r_j & \text{if } \phi_i \leq \pi, \phi_j \leq \pi\,, \\ x_i + r_i cos(\phi_i), x_j - r_j & \text{if } \phi_i \leq \pi, \phi_j > \pi\,, \end{cases} \tag{7}$$

where the ith oscillator outputs to its respective vertical motor and the jth oscillator outputs to its respective horizontal motor. This allows us to discard the parts of the oscillator output that are not



(a) Dual Tripod (b) Ripple (c) Wave (d) Four-Two

Figure 3: Learning curve for the four gaits (median and 65th percentile). We can see how, for all the gaits, BO learns to walk from scratch within 50 iterations. After the optimization, Wave and Ripple are the fastest gaits at $\sim 1\,\mathrm{cm/s}$.



(a) Dual Tripod (b) Ripple (c) Wave (d) Four-Two

Figure 4: Performance measured for the four gaits, and the corresponding PFs. ParEGO is able to quickly explore the PF for each of our four gaits.

consistent with the physical constraints of the physical robot, since the actual leg actuators cannot partially retract. We choose to mutually couple all six of the vertical oscillators (with a coupling weight of 4 to ensure quick convergence on stable limit cycles). Each of the horizontal oscillators are also coupled with their respective vertical oscillator in order to encapsulate the dynamics of each leg. We chose to implement four different gaits with the CPG – tripod, ripple, wave, and four-two (see Figure 2). For a more detailed description of these gaits we refer the reader to [3]. We use the same frequency and phase difference for the whole network in order to reduce the number of parameters and speed up the rate of convergence. We use two separate parameters for amplitude, each controlling the left and right set of legs respectively. This choice of parameters allows us to control the turning of the robot which is necessary for path planning and corrections for not walking straight.

5.2 Learning to Walk Straight

We optimized the four gaits considered (i.e., dual tripod, ripple, wave, and four-two) using as our objective function the walking speed of the robot (measured as the distance traveled after $1\,\mathrm{s}$). Since some gaits result in curved motions, we also penalized the speed objective with a term proportional to the drift from the axis of locomotion. The optimization used the 4 parameters outlined in Section 3 and was repeated 20 times for each of the gaits. In Figure 3, we show the median and 65th percentiles of the best solution obtained so far in the trials. The results show that the optimizer was able to learn to walk from scratch within 50 iterations. Moreover, it can be also noted that the optimized wave and ripple are the fastest gaits at $\sim 1\,\mathrm{cm/s}$.

5.3 Multi-objective Gait Optimization

In the previous experiment we only considered walking speed as our objective. However, for practical gait design, energy efficiency is another objective of great interest, particularly when it comes to designing gaits for a microrobot with real energy restrictions. For this reason, we now consider a multi-objective optimization setting and compare the different gait w.r.t. both walking speed, and energy consumption. The energy consumption of the robot was computed by measuring the forces exerted by each of the 12 motors along the axis of actuation and calculating the power used to actuate the motors.

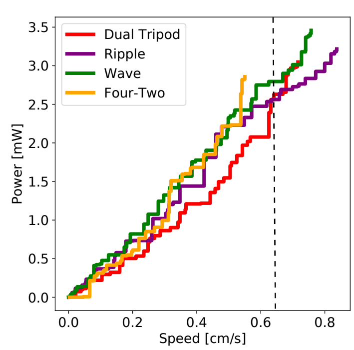


Figure 5: Comparison of the PFs obtained for the different gaits.

Since the retraction of the legs is spring powered, the energy input in the cycle is only during motor extension. Hence, we only consider the cost of extending the legs. With the mass of the robot and the time of each trial being held constant, this allows us to quantify the energy efficiency of a gait and estimate the cost of transport.

We optimized again the four gaits with the same parameters as used previously but this time using multi-objective Bayesian optimization with a budget of 50 iterations. In Figure 4 we can see the performance measured and Pareto fronts obtained for the different gaits. To better compare the PF from the different gaits, we also visualized just the PFs together in Figure 5. From these results, we can see how the tripod gait dominates the other gaits for speed $< 0.6\,\mathrm{cm/s}$, while Ripple dominates when the speed is $> 0.6\,\mathrm{cm/s}$, hence giving a clear indication of which gait is preferable under different circumstances.

5.4 Discovering New Gaits with Multi-objective Optimization

In addition to optimizing the four nature-inspired gaits, we also tested multi-objective optimization on the walker without constraining to using predefined gaits. To parametrize the oscillator couplings, we thus discretized each gait into intervals of constant length. Within each of these intervals, we assume that each leg steps exactly once, keeping each of the oscillators in the CPG in phase with each other. This allows us to parametrize gaits by assigning each leg a point during each interval where it begins stepping. While this parametrization excludes certain gaits that cannot be expressed in this form, we leave the study of more sophisticated gait parameterizations for gait discovery to future works.

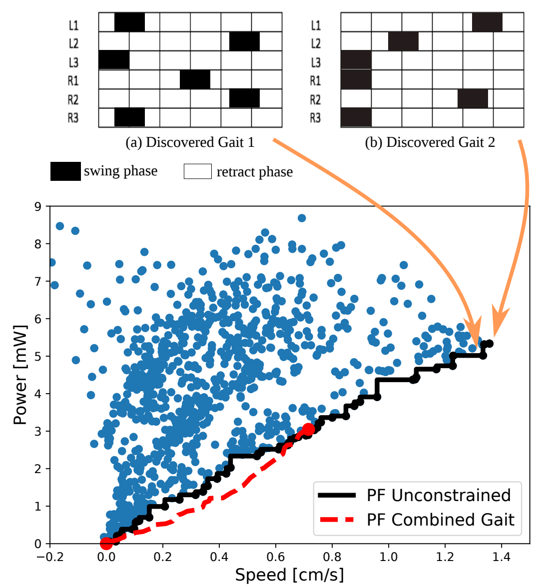


Figure 6: PF of the unrestrained gait optimization versus the best performance of the four nature-inspired gaits. The faster solutions outperform the fastest nature-inspired gaits, albeit with more energy expenditure. However, the inability of the optimizer to match the performance of the gaits at lower speeds within 1250 trials shows that the gait parametrization can help limit the search space to find better solutions easier. *(top)* Pattern for two of the discovered gaits.

The resulting multi-objective optimization task had eight parameters (frequency, phase difference between horizontal and vertical motors, and the six gait coupling parameters). Due to the higher parameter dimensionality than before, and because this training was not intended for on-line training, we ran the optimization for 250 iterations in order to allow a more comprehensive exploration of the optimization space. We also repeated the optimization five times for a total of 1250 trials. In Figure 6 we can see the Pareto front for the resulting gaits. We found that the fastest discovered gaits were actually able to outperform the four nature-inspired gaits implemented by a substantial margin. Even while penalizing curved paths, the fastest discovered gait outperformed Ripple (the fastest nature-inspired gait we found) by almost 50%. However, for low-speed gaits, the nature inspired gaits out-perform the gaits produced by the unconstrained optimization, indicating the optimization did not yet fully converged to the optimal PF.

5.5 Learning to Walk on Inclined Surfaces

We now consider the case of contextual optimization and specifically the task of gait optimization for slopes with different inclinations. We framed learning to walk on inclined terrain as a contextual policy search, where the angle of the inclination is the context. In this experiment we decided to use Dual Tripod for our gait with mostly the same open parameters as the previous experiments. We only used one parameter to represent the amplitude for the entire network in order to keep the number

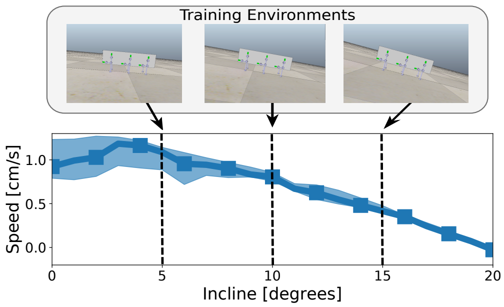
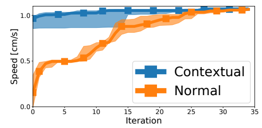
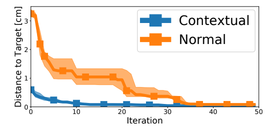


Figure 7: Performance of the contextual policy (median and 65th percentile) for a wide range of inclines. The policy was trained only at 5, 10 and 15 degrees, but it was capable of generalizing smoothly to unseen inclinations.



(a) Inclined surface.



(b) Curved trajectory.

Figure 8: Comparison between the optimization performance of a contextual optimizer and a normal optimizer for two different tasks: (a) walking on inclines (b) walking curved trajectories. In both cases, the contextual optimizer can leverage prior experiments to obtain high-performing gaits in fewer experiments.

of parameters low with the addition of a contextual variable, leaving us with 3 parameters and 1 contextual parameter. To respect real world constraints, where testing randomly sampled incline angles over a continuous interval can be excessively time-consuming, we chose at training time to perform experiments only from a small number of inclines: 5, 10, and 15 degrees.

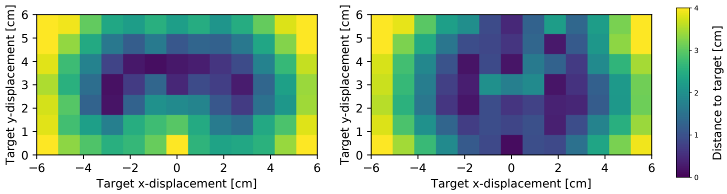
After optimizing the gaits for these three inclines over 50 iterations, we studied how the contextual optimizer is able to generalize across the context space by testing the performance of the contextual policy for a wide range of inclines. In Figure 7 we can see that the policy performs well on intermediary inclines and seems to smoothly interpolate between the training inclines as is desirable. The gradual decrease in performance as the inclines get steeper can be attributed to the increasing physical difficulty for climbing up steeper inclines. We also compared cBO against using standard BO to train the robot for an untested incline. As shown in Figure 8a, the contextual optimization was able to converge on optimal performance significantly faster than normal Bayesian optimization. This result demonstrates the ability of the contextual optimization to use data accumulated in other contexts to quickly reach optimal gaits in unseen contexts.

5.6 Learning to Curve

Another useful task that can be framed as contextual optimization is learning motor primitives to walk curves for use in path planning. We used the same parameters as in Section 5.2 and the contextual parameters in this case were the target displacements along both the x and y axes from the point of origin. In order to train particular trajectories, we selected five evenly spaced target points along the front quadrant of the field of vision. Since the primary objective was to reach the desired destination, we chose to use the distance of the final position to the target position as our sole objective function. We found that over 10 repetitions, the walker was able to accurately move and turn towards all of the target points within 250 iterations. We also compared the performance of cBO against standard BO on a previously unseen target position $(4\cos\pi/16, 4\sin\pi/16)$. We found that, as in the case of inclinations, the contextual policy was able to learn the optimal parameters for a novel trajectory within very few iterations.

5.7 Learning Motor Primitives for Path Planning

In the previous experiment we learned motor primitives capable of walking curved trajectories. While the model handled trajectories near and between the targets quite well, the performance on trajectories well within the physical capabilities of the robot but not in proximity to the targets left much to be desired, as shown in Figure 9. We now demonstrate how our approach presented in Section 4 can be used to significantly improve the movement accuracy (compared to cBO using the same data), as well as how such motor primitives can be used to perform path planning. First, we reused the data from the previous experiment in order to reformulate the task as a multi-objective optimization as described in Section 4. Then, we used our trained model to sample 10,000 trajectories by randomly sampling from the parameter space. Out of all these trajectories, we selected the one with the smallest expected error subject to not walking through the wall. Evaluating the resulting sequence of motor primitives on the



(a) cBO (b) Our approach

Figure 9: Comparison of the performances of cBO and our approach for learning motor primitives (using the same data). With the robot having an initial position of $(0, 0)$, we evaluated the error between the desired position (indicated by the element of the grid) and the reached position. Darker color indicates better target accuracy. While cBO accurately learned trajectories near the training targets, it did not generalize well to unseen targets. In contrast, our approach had a more comprehensive coverage as it was able to leverage better information about the environment to improve generalization.

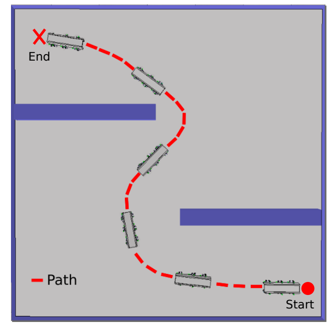


Figure 10: Path constructed using the locomotion primitives learned with our approach.

real system (i.e., the simulator) demonstrated that the expected trajectory was capable of navigating the maze.

6 CONCLUSIONS

Designing controllers for locomotion is a daunting task. In this paper, we demonstrated on a simulated microrobot that this process can be significantly automated using BO. Our main contributions are two-fold: 1) we introduced a coherent curriculum of increasing challenging tasks, which we use to evaluate our microrobot and the parametrization of its controller using existing BO techniques. 2) we presented a new approach that enables walking robots to efficiently learn motor primitives from scratch. By using the data collected from contextual optimization we reformulate the problem into a multi-objective optimization task, and learn a model that can map any set of parameters to a predicted trajectory. This model can subsequently be used for path planning. Our experimental results demonstrate using this approach the microrobot is able to successfully navigate through a maze.

References

[1] J. Branke, K. Deb, K. Miettinen, and R. Słowiński. *Multiobjective optimization: interactive and evolutionary approaches*, volume 5252. Springer, 2008. doi: 10.1007/978-3-540-88908-3.

[2] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence (AMAI)*, 76(1):5–23, 2015. ISSN 1573-7470. doi: 10.1007/s10472-015-9463-9.

[3] R. Campos, V. Matos, and C. Santos. Hexapod locomotion: A nonlinear dynamical systems approach. In *Annual Conference of IEEE Industrial Electronics Society (IECON)*, pages 1546–1551, Nov 2010. doi: 10.1109/IECON.2010.5675454.

[4] G. Capi, M. Yokota, and K. Mitobe. A new humanoid robot gait generation based on multiobjective optimization. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 450–454, 2005. doi: 10.1109/AIM.2005.1501032.

[5] S. Chernova and M. Veloso. An evolutionary approach to gait learning for four-legged robots. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2562–2567. IEEE, 2004.

[6] D. S. Contreras, D. S. Drew, and K. S. Pister. First steps of a millimeter-scale walking silicon robot. In *International Conference on Solid-State Sensors, Actuators and Microsystems (TRANSDUCERS)*, pages 910–913. IEEE, 2017.

[7] T. Ebefors, J. U. Mattsson, E. Kälvesten, and G. Stemme. A walking silicon micro-robot. In *Proc. Transducers' 99*, pages 1202–1205, 1999.

[8] S. Hollar, A. Flynn, C. Bellew, and K. S. J. Pister. Solar powered 10 mg silicon robot. In *IEEE International Conference on Micro Electro Mechanical Systems (MEMS)*, pages 706–711. IEEE, 2003.

[9] A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural networks*, 21(4):642–653, 2008.

[10] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[11] J. Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1): 50–66, January 2006.

[12] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86:97, 1964.

[13] D. J. Lizotte, T. Wang, M. Bowling, and D. Schuurmans. Automatic gait optimization with Gaussian process regression. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 944–949, 2007.

[14] J. H. Metzen, A. Fabisch, and J. Hansen. Bayesian optimization for contextual policy search. In *IROS Workshop on Machine Learning in Planning and Control of Robot Motion*, 2015.

[15] C. Niehaus, T. Röfer, and T. Laue. Gait optimization on a humanoid robot using particle swarm optimization. In *Proceedings of the Second Workshop on Humanoid Soccer Robots in conjunction with the*, 2007.

[16] M. Oliveira, L. Costa, A. Rocha, C. Santos, and M. Ferreira. Multiobjective optimization of a quadruped robot locomotion using a genetic algorithm. In *Soft Computing in Industrial Applications*, volume 96, pages 427–436. Springer, 2011. ISBN 978-3-642-20504-0. doi: 10.1007/978-3-642-20505-7_38.

[17] M. Oliveira, V. Matos, C. P. Santos, and L. Costa. Multi-objective parameter CPG optimization for gait generation of a biped robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3130–3135, 2013.

[18] V. Pareto. *Manuale di Economia Politica*, volume 13. Societa Editrice, 1906.

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

[20] K. Saito, K. Iwata, Y. Ishihara, K. Sugita, M. Takato, and F. Uchikoba. Miniaturized Rotary Actuators Using Shape Memory Alloy for Insect-Type MEMS Microrobot. *Micromachines*, 7 (4), 2016. ISSN 2072-666X. doi: 10.3390/mi7040058.

[21] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[22] R. Tedrake, T. W. Zhang, and H. S. Seung. Stochastic policy gradient reinforcement learning on a simple 3d biped. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2849–2854. IEEE, 2004.

[23] M. Tesch, J. Schneider, and H. Choset. Using response surfaces and expected improvement to optimize snake robot gait parameters. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1069–1074. IEEE, 2011.

[24] M. Tesch, J. Schneider, and H. Choset. Expensive multiobjective optimization for robotics. In *International Conference on Robotics and Automation (ICRA)*, pages 973 – 980, 2013.

[25] D. Vogtmann, R. S. Pierre, and S. Bergbreiter. A 25 mg magnetically actuated microrobot walking at > 5 body lengths/sec. In *IEEE International Conference on Micro Electro Mechanical Systems (MEMS)*, pages 179–182, Jan 2017. doi: 10.1109/MEMSYS.2017.7863370.

[26] T. Wagner, M. Emmerich, A. Deutz, and W. Ponweiser. On expected-improvement criteria for model-based multi-objective optimization. In *Parallel Problem Solving from Nature (PPSN) XI*, pages 718–727. Springer, 2010.

[27] J. Yu, M. Tan, J. Chen, and J. Zhang. A survey on CPG-inspired control models and system implementation. *IEEE Transactions on Neural Networks and Learning Systems*, 25(3):441–456, mar 2014. doi: 10.1109/tnnls.2013.2280596.