Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Probabilistic Inferences on Quadruped Robots: An Experimental Comparison

Jiahui Zhu, Chunyan Rong and Andre Rosendo
School of Information Science and Technology
ShanghaiTech University
Shanghai, China
{*zhujh1, rongchy, arosendo*}*@shanghaitech.edu.cn*

***Abstract*— Due to the reality gap, computer software cannot fully model the physical robot in its environment, with noise, ground friction, and energy consumption. Consequently, a limited number of researchers work on applying machine learning in real-world robots. In this paper, we use two intelligent black-box optimization algorithms, Bayesian Optimization (BO) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES), to solve a quadruped robot gait's parametric search problem in 10 dimensions, and compare these two methods to find which one is more suitable for legged robots' controller parameters tuning. Our results show that both methods can find an optimal solution in 130 iterations. BO converges faster than CMA-ES within its constrained range, while CMA-ES finds the optimum in the continuous space. Compared with the specific controller parameters of two methods, we also find that for quadruped robot's oscillators, the angular amplitude is the most important parameter. Thus, it is very beneficial for the quick parametric search of legged robots' controllers and avoids time-consuming manual tuning.**

***Index Terms*— Bayesian Optimization, Covariance Matrix Adaptation Evolution Strategy, Quadruped Locomotion.**

I. Introduction

Black-box problems usually present the following characteristics: A. It is expensive and inefficient to optimize a non-linear objective function. B. The objective function's gradients are not available or not useful. In recent years, there are many black-box optimization methods proposed by mathematicians, such as grid search (GS), random search (RS) and genetic algorithms (GA). GS is the simple most black-box optimization method. Due to the number of evaluations, it increases exponentially with the number of parameters, and this method is difficult to be used in high-dimensional black-box optimization problems [1]. The RS algorithm does not make any assumptions about the machine learning algorithm and it is only a useful baseline to evaluate how complex a black-box optimization problem is [2]. As for GA, the global optimal solution of the optimization problem can be obtained, but its programming implementation is relatively complex [3]. First, the problem needs to be coded, and then the problem needs to be decoded after finding the optimal solution. Those are all traditional methods and many researchers did experiments with them. In the year of 2012, Bayesian Optimization for machine learning is proposed by J. Snoek [4], and it is a powerful strategy for finding the extrema of objective functions that are expensive to evaluate. In the work of [5], BO is applied in a 2-dimensional bipedal locomotion problem compared to grid search and pure random search, and it finds that BO is a promising method for efficient optimization. In our previous work [6] we applied BO in a 3-dimensional quadruped locomotion parametric search problem, and we found that BO can still efficiently choose stable parameters for walking gaits within a few iterations.

In this paper, we used a different data-efficient optimization algorithm called Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [7] on our quadruped robot's locomotion, and compared BO and CMA-ES these two methods with the same evaluation criteria- a forward displacement within 10 s. Since we changed our original robot [6] structure and the morphological changes were significant, we need to re-optimize the gait with BO algorithm. This again proves that BO can find optimization parameters and has self-adaptations with the mechanical changes in robot's morphologies. In Section II, we present the two methods used in this work, BO and CMA-ES, and the experiment set-up of our quadruped robot $SmartAnt$, shown in Fig. 1. In Section III and IV, we present the experimental results and discuss these two methods with several comparisons, while in the last section, we conclude our work.

II. Method

A. Bayesian Optimization

BO is an optimization method that combines Gaussian processes (GPs) with Bayes statistical theory to predict the next best sampling point of an expensive cost function. It is based on previous observations and beliefs [8]. GPs are probabilistic nonlinear regressions and provide a method for modeling probability distributions over functions with mean function and covariance function. The entire process is expressed by a flowchart, as shown in Fig. 2.

*This work was supported by the National Natural Science Foundation of China, project number 61850410527, and the Shanghai Young Oriental Scholars project number 0830000081.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

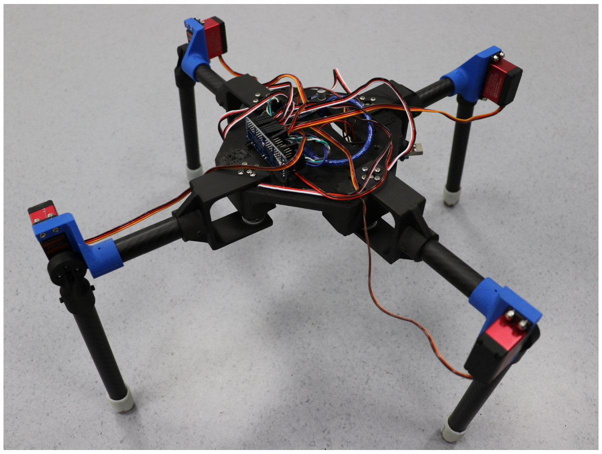


Fig. 1. Our quadruped robot $SmartAnt$ in this experiment. It shows that $SmartAnt$'s overall size is 410 × 410 × 250 mm. The proximal link is 160 mm and the distal link is 230 mm while the body's size is 150 × 150 mm. The weight of the total robot is only 1.25 kg including the battery.

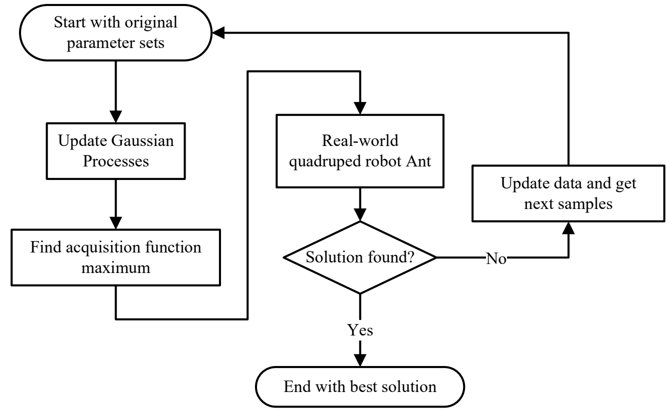


Fig. 2. Flowchart of Bayesian Optimization algorithm. BO first uses a Gaussian process to model a surrogate model of the input. Here the inputs are the parameters of each servo's oscillators. Then acquisition function finds the next probable better samples and we execute them in our real-world quadruped robot $SmartAnt$ to see how these parameters behave. Next, we put the new input into GPs and update the model. During this loop, BO will find the optimal solution to our problem.

We use the $Matérn$ kernel as our GPs' covariance kernel function [9]:

$$k_{Matern}(d) = \sigma_p^2 \frac{2^{1-v}}{\Gamma(\nu)}(\frac{\sqrt{2\nu d}}{l})^v K_\nu(\frac{\sqrt{2\nu d}}{l}) \qquad (1)$$

where d is the Euclidean distance between two inputs, $\Gamma(\nu)$ and K_ν denotes the Gamma function and the Bessel function of order ν. Meanwhile, ν is a parameter with positive value 5/2, l shows the characteristic length-scale, and both of them are hyper-parameters in this function, since they are supposed to be fixed before the function is used as a kernel. It also takes noise into account, which is denoted by σ_p^2. $Matérn$ kernel has higher flexibility for it is suitable for both smooth and sharp areas [10].

For the acquisition function [11], we choose the probability of improvement (PI) with a trade-off parameter $\xi \geq 0$:

$$PI(x) = \phi(\frac{\mu(x) - f(x^+) - \xi}{\sigma(x)}) \qquad (2)$$

where $x^+ = argmax_{x_i \in x_{1:t}} f(x_i)$, $\phi(\cdot)$ is the normal cumulative distribution function. Kushner [12] recommended a schedule for ξ, so that it started fairly high early in the optimization, to drive exploration, and decreased toward zero as the algorithm continued.

You can find more details about BO in our previous work [6] because we had a detailed description of it.

B. Covariance Matrix Adaptation Evolution Strategy

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is a stochastic method for continuous domain optimization of non-linear, non-convex functions. It is one kind of sample-efficient evolution algorithm that could adapt the covariance matrix of a multivariate normal distribution and generate isotropic search points conveniently. Usually, CMA-ES performs the following three steps at each generation [7]:

1) Sampling a Multivariate Normal Distribution: The basic equation for sampling the search points, for generation number g.

$$x_k^{g+1} \sim m^{(g)} + \sigma^{(g)} N\left(0, C^{(g)}\right), \; for \; k = 1, ..., \lambda \qquad (3)$$

where

$\sim$ donates the same distribution on both sides.

$N\left(0, C^{(g)}\right)$ is a multivariate normal distribution with zero mean and covariance matrix $C^{(g)}$.

x_k^{g+1}, k-th offspring (individual, search point) from generation $g+1$.

$m^{(g)} \in \mathbb{R}^n$, mean value of the search distribution at generation g.

$\sigma^{(g)} \in \mathbb{R}^+$, "overall" standard deviation, step-size, at generation g.

$C^{(g)} \in \mathbb{R}^{n \times n}$, covariance matrix at generation g.

λ, the population size, sample size, number of offspring.

2) Selection and Recombination: The new mean $m^{(g+1)}$ of the search distribution is a weighted average of μ selected points from the sample $x_1^{(g+1)}, ..., x_\lambda^{(g+1)}$.

$$m^{(g+1)} = \sum_{i=1}^{\mu} w_i x_{i:\lambda}^{(g+1)},$$
$$\sum_{i=1}^{\mu} w_i = 1, w_1 \geq w_2 \geq ... \geq w_\mu > 0 \qquad (4)$$

where

$\mu \leqslant \lambda$, the parent population size, i.e. the number of selected points.

$w_{i=1,\dots,\mu} \in \mathbb{R}^+$, the positive weight coefficients for recombination.

$x_{i:\lambda}^{(g+1)}$, i-th best individual out of $x_1^{g+1}, \dots, x_\lambda^{g+1}$ from (3).

3) Adapting the Covariance Matrix: Rank-μ-updating method.

$$C^{(g+1)} = c_\mu \underbrace{\sum_{i=1}^{\lambda} w_i x_{i:\lambda}^{(g+1)} (y_{i:\lambda}^{(g+1)})^\intercal}_{\text{rank- } \mu \text{ update}} \tag{5}$$

where

$c_\mu \approx min(\mu_{eff}/n^2, 1-c_1) \leqslant 1$, learning rate for updating the covaraince matrix.

$y_{i:\lambda}^{(g+1)} = (x_{i:\lambda}^{(g+1)} - m^{(g)})/\sigma^{(g)}$, the mutation vector expressed in the coordinate system where the sampling is isotropic.

Briefly, we wrote the following pseudo-code to show how we adopted CMA-ES algorithm in our work.

Algorithm 1 CMA-ES

```
Initialize distribution parameters θ
Set population size λ ∈ ℕ
for i = 1, 2, ··· , 130, g ← g + 1 do
    Sample new population of search points P(x | θ → x_1, ..., x_λ ∈ ℝ
    Evaluate x_1, ..., x_λ on objective function f
    Selection and Recombination: Moving the Mean
    Covariance Matrix Adaptation
    Update θ ← F_θ (θ, x_1, ..., x_λ, f (x_1) , ..., f (x_λ))
end for
```

C. Experimental Setting

In our work, BO and CMA-ES algorithms are embedded in a quadruped robot, which is capable of 3-dimensional locomotion. The robot is consists of a light body and four legs of eight joints. Each joint's movement is controlled by a 20kg·cm torque servo, and can rotate 270°. We use the 3-D printing joints to connect the proximal link and the distal link. In addition to the servo, control board, and direct current-direct current (DC-DC) regulator, the components of our robot are all printed by carbon fiber. Carbon fiber materials are known to have high strength and lightweight. Therefore, the total weight of our quadruped robot is only 1.25 kg, as shown in Fig. 1. The arena size of this experiment is 2m × 4 m with marked color lines. A top view of the experiment site can be observed in Fig. 5.

TABLE I

GAIT PARAMETERS CONFIGURATION

Parameter[a]		Oscillation Controls	Range
1	LF/LH/RH/RF_PL_Mid	Angular midpoint	[-45°, 45°]
2	LF/LH/RH/RF_DL_Mid		[-15°, 90°]
3	LF/RF_PL_Amp	Angular amplitude	[0°, 45°]
4	LH/RH_PL_Amp		
5	LF/RF_DL_Amp		
6	LH/RH_DL_Amp		
7	LF/RF_PL_Pha	Phase	[-π/2, π/2]
8	LH/RH_PL_Pha		
9	LF/RH_DL_Pha		
10	LH/RF_DL_Pha		

[a]Left (L), Right (R), Forelimb (F), Hindlimb (H), Proximal Link (PL), Distal Link (DL). For example, LF_PL_Mid means the angular midpoint of the left forelimb proximal link's oscillator.

1) Sinusoidal Variations Controller: In the legged robot locomotion, scientists generally use non-linear oscillators as mathematical models of the natural Central Pattern Generators (CPGs) to model the oscillations of the joint angles produced by the muscular activity [13]. In our work, in order to facilitate numerical computations, we adopt a simplified sinusoidal variation of the robot's joint angle. The form of the oscillator can be expressed in the following equation:

$$x(t) = Asin(2\pi t + \varphi) + x_0 \tag{6}$$

where $x(t)$ is the current servo angle of the robot at time t, A is the angular amplitude of the oscillations, φ is the oscillation phase and x_o is the angular midpoint of the oscillations for each joint.

2) Performance Measurement: In order to test the performance of our quadruped robot, we set every controller's action time to 10 s. Therefore, within 10 s, $SmartAnt$ must move as far as possible away from its initial position. However, performance cannot simply be measured as a straight distance from the origin because our quadruped robot is capable of dynamic motion, unlike a biped robot that only can move forward or backward. In some cases, the robot will go left to right and even makes a circle in place. Therefore, in such cases, the evaluation of the robot becomes poor. For example, the performance evaluation of the same distance of the forward walking and the backward walking from the start point are the same. To overcome this problem, we use the forward displacement between the start and end positions over 10 s as our evaluation criteria.

3) Parameters Chosen: $SmartAnt$ has four legs with eight servos and each servo is controlled by an oscillator. It can be known from (6), that each oscillator has 3 unfixed parameters, so there are 24 parameters for eight servos to be tuned. Even if each parameter sets 4 choices, the search space will become 28 trillion (4^{24}), which is extremely large. Due to the symmetric morphology of the quadruped robot, we decide to do a symmetrical parameter configuration to reduce our problem to optimize 10-dimensional parameters. For more details, it can be referred to our previous work [6]. Table I gives a summary of the corresponding optimization parameters and their ranges. Here, for BO method,

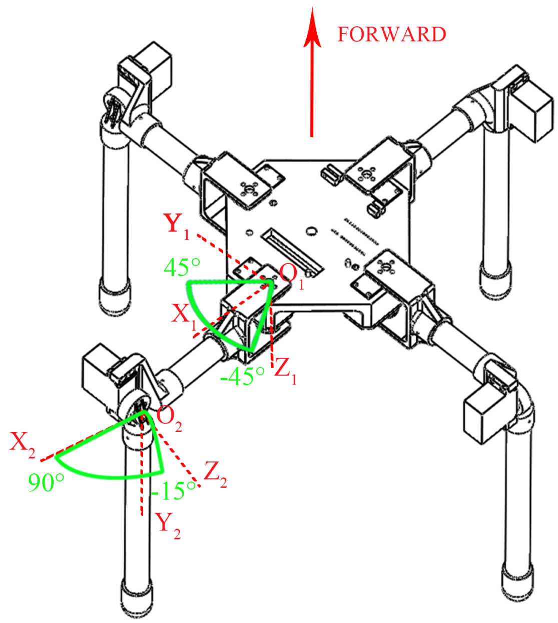


Fig. 3. Reference frames of proximal links and distal links. The proximal link makes the body move in the horizontal face while the distal links control its vertical movement. As a result, $SmartAnt$ could learn variety gaits during 3-dimensional locomotion.

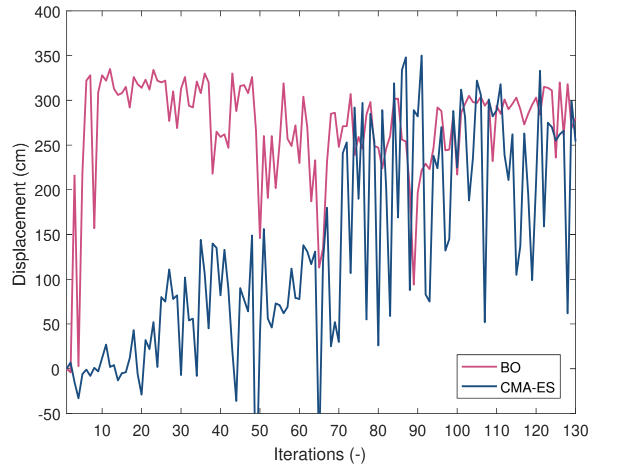


Fig. 4. The figure shows the results of Bayesian Optimization (BO) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Negative values mean that our quadruped robot $SmartAnt$ moves backwards instead of moving forward and we discard the data below -50 cm. It is possible to see that BO was quicker in finding good solutions, but in the long run CMA-ES reached higher results.

because the acquisition function of BO needs to maximize the probability of improvement between each parametric combination to the incumbent maximum objective value and BO is sequential by nature, it is not feasible for BO to take the continuous domain of a high 10-dimensional parametric search problem. While as mentioned before in Section II. B, CMA-ES method's search points are sampled by a multivariate normal distribution with zero mean and covariance matrix for continuous domain optimization. So we take four discrete intervals of the BO's range, while CMA-ES is a continuous parametric search.

In order to better understand the quadruped locomotion of the robot, we drew a reference frame and marked the maximum and minimum range that each link can move, as shown in Fig. 3.

III. Results

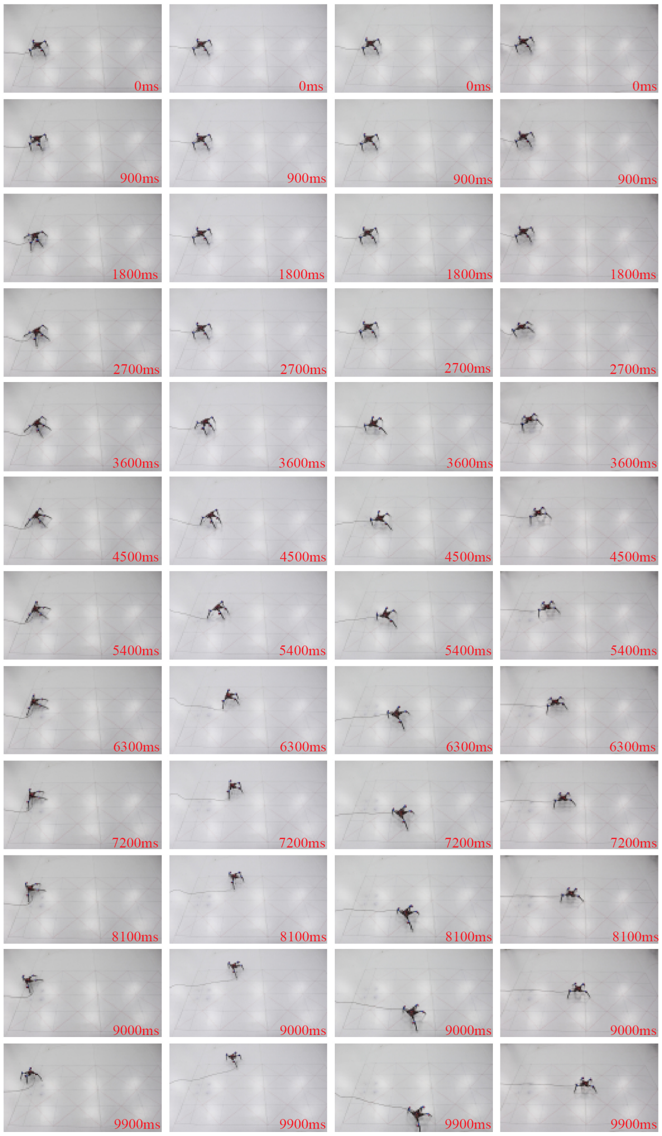
We performed two experiments with BO and CMA-ES on our quadruped robot $SmartAnt$'s gait learning. In order to compare the experimental results of the two methods, we put BO and CMA-ES on the same scale, as shown in Fig. 4.

Each algorithm was done in 130 iterations, where CMA-ES had 5 iterations per generation (total of 26 generations). To ensure a fair comparison, both BO and CMA-ES start from the same initial state of the robot, with parametric combinations are [0°, 0°, 0°, 0°, 0°, 0°, 0, 0, 0, 0]. Upon the completion of the first trial BO's next sample is selected by the acquisition function in the existing parametric combinations, and CMA-ES's new search points are sampled normally distributed by the initial parameters.

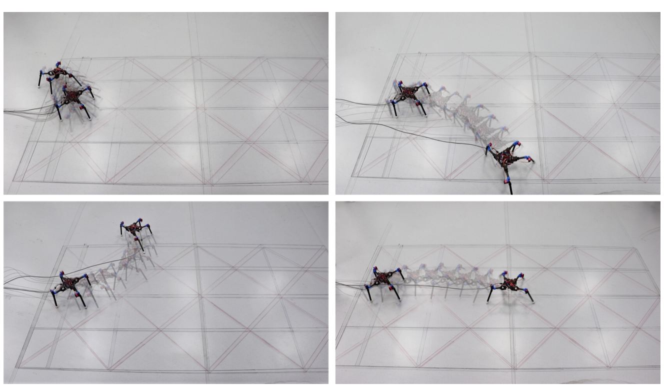
The initial displacements of BO and CMA-ES are both 0, and as the iterative process progresses, they begin to explore and exploit. As it can be seen in Fig. 4, BO quickly reaches a parametric combination that results in forward walking, and reaches a maximum displacement of 335 cm at the 12th iteration. CMA-ES slowly oscillates, and the fitness reaches its maximum at the 19th generation (corresponding to the 81st iteration overall), with a displacement of 350 cm.

During our experiment not all parametric combinations could produce a forward locomotion with $SmartAnt$. Therefore, we selected several representative gaits to display, so that the reader can visually understand the different gaits of the robot during 3-dimensional locomotion. First, we used a normal camera to shoot the four gaits of the robot, and 12 snapshots (900 ms time difference between frames, total walking distance of 4 meters) are arranged in each row as shown in Fig. 5(a). After that, we take these snapshots to form the ant's walking trajectory, as shown in Fig. 5(b). The behavior of the robot is erratic in the beginning of the experiment, and occasionally the robot collapses on the floor.

Due to the computational complexity, in our experiments BO's parameters are sets in a discrete domain, thus causing a large difference between the parameters obtained by CMA-ES, shown in the upper two figures in Fig .6. Because our problem is a 10-dimensional parametric search, and 10 dimensions are not feasible to show in reality. We only selected two pairs of parameters to compare BO and CMA-ES methods.



(a) Snapshot of $SmartAnt$.



(b) Trajectory of $SmartAnt$.

Fig. 5. Our quadruped robot learning how to walk. In order to measure the displacement we set the midpoint of the body as the origin, and the start position of the robot is the left middle side of the marked arena. (a) From left column to right column we show: the robot rotating in place,walking to the left, walking to the right, and walking forward. (b) A demonstration of the trajectory for each one of these trials.

IV. Discussion

A. BO vs CMA-ES

BO and CMA-ES were previously used to solve black-box optimizations [5], [14], [15], [16], [17], and these two methods rely on a probabilistic approach to find an optimal solution. In [5], [14], BO is applied on a bipedal robot and a hexapod robot, respectively, and it is shown that BO could, data-efficiently, find the optimal solution and adapt to the morphological changes of robots. Our results agree with this aspect, as we changed our joint structure between proximal links and distal links in a prior design iteration and BO was still capable of finding the parametric combinations of the forward walking. However, for CMA-ES, it is a state-of-the-art method in derivative-free optimization in robotics, and many researchers still apply it in the physics-based animation of bipedal locomotion and quadruped locomotion [15], [16]. Thus, we apply the CMA-ES method into a real-world quadruped robot and find it is sample-efficient for high dimensional parametric search and large function evaluation, which was also approached by [17], using CMA-ES for hyperparameters optimization of deep neural networks.

We evaluated BO and CMA-ES on our quadruped robot and from Fig. 4, we can see the performance they obtained per iteration. BO quickly reached a peak in a few iterations, while the performance of CMA-ES was slowly rising. This is because CMA-ES was sampled from zero mean and covariance matrix at the beginning, and for the next generation, it updated its mean and covariance matrix. Therefore, from the rising trend it can also be seen that CMA-ES method's covariance adaptation is slowly approaching the direction of the optimal path. As for BO method's quick convergence, the reason is that BO uses Gaussian Processes to build a surrogate model that can use prior knowledge to find the optimal solution. As the iterations reached the maximum displacement the algorithm naturally transits to an exploration behavior (unstable performance). For BO, the acquisition function has a trade-off coefficient that balances these while CMA-ES has weighted recombination and mutation hyperparameters that perform this role. Within our experiments CMA-ES reached a higher forward displacement than BO, and this shows that CMA-ES performed better than BO in the long run, and the fact that CMA-ES was performing a search in a continuous search space (as opposed to BO's discrete search space) might have a strong influence on these results.

B. Angular Midpoint vs Angular Amplitude

As shown in Fig. 6(a) and 6(b), it is easy to see that the heat map generated by CMA-ES is more complicated and irregular, because the sampling methods used by CMA-ES is distributed sampling by the newly updated covariance matrix, which can generate continuous samplings. But for BO, it uses acquisition function to choose the next probable best samples in the observation space. Also, we plot another

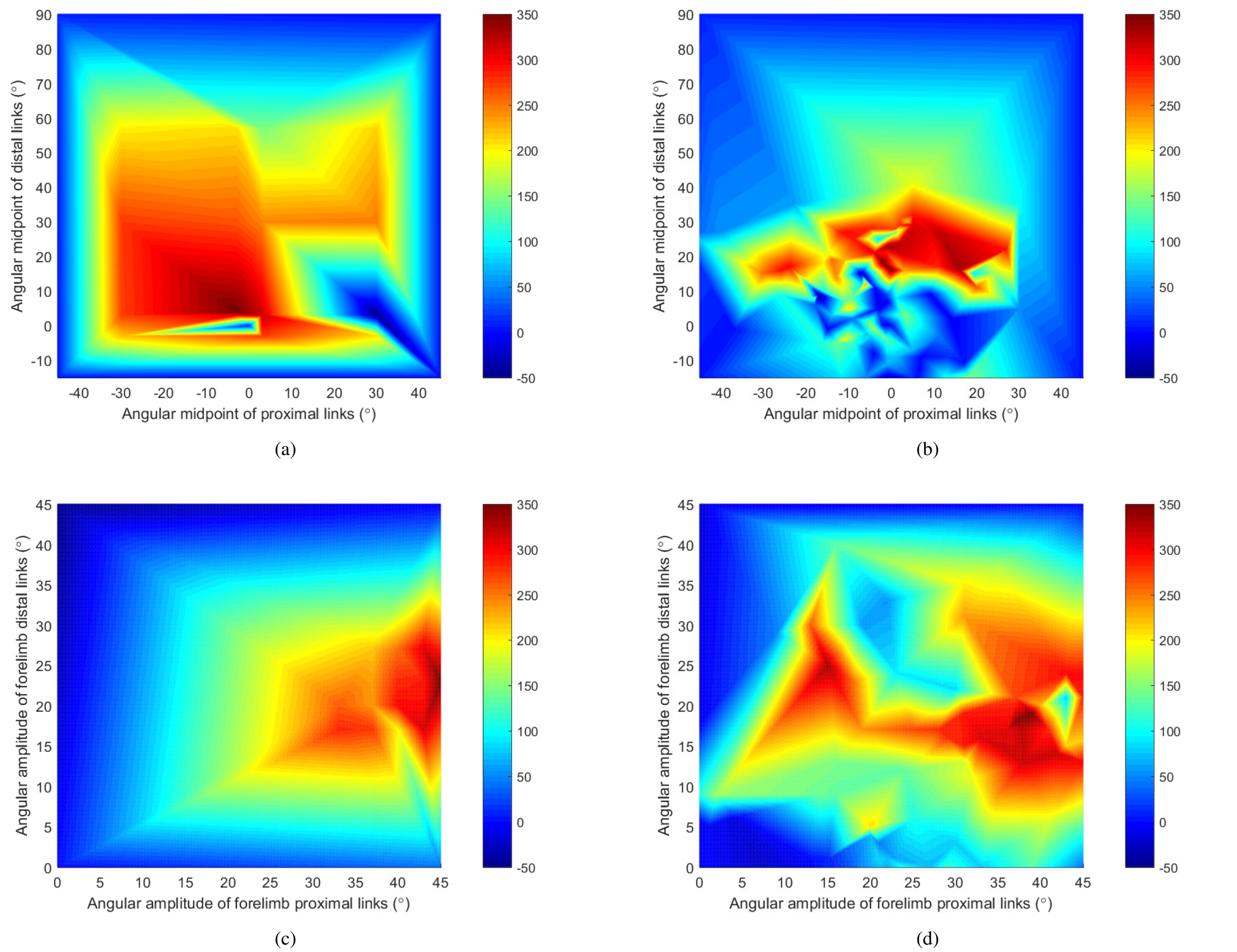


Fig. 6. Heat map of gait parametric combinations under two different methods. (a) are the parameters 1: oscillation angular midpoint of proximal links and 2: oscillation angular midpoint of distal links from BO algorithm. Then (b) are the parameters 1 and 2 from CMA-ES algorithm. Also, (c) are the parameters 3: oscillation angular amplitude of forelimb proximal links and 5: oscillation angular amplitude of forelimb distal links from BO algorithm. As the same, (d) are the parameters 3 and 5 from CMA-ES algorithm. The color bar beside the plot shows the forward displacement values obtained by each parameter pair.

pair parameters, the angular amplitude of forelimb proximal and distal links, and we take a look at the point between [5°, 20°] of proximal links and [15°, 25°] of distal links in Fig. 6(c), 6(d). BO transitions smoothly from yellow to blue, while the CMA-ES has another heat circle in the middle. This is because the CMA-ES continuously samples, there are many sample points in this interval, and BO only selects two segments in the interval, and directly makes the difference between two points, thus forming a gentle color change.

As can be seen from Fig. 6(c) and 6(d), even though the domains of BO and CMA-ES, one is discrete, the other is continuous. However, the heat maps showed still have some similarities. This shows that in this 10-dimensional parameter, controlling the angular amplitude of the oscillator is more important than the angular midpoint. This result is quite meaningful for researchers who are engaged in robotics. When you are not sure which one of the ten-dimensional parameters are prioritized, our work finds that adjusting the amplitude first can significantly improve the robot's walking performance.

V. Conclusion

In this work, we solved a 10-dimensional parametric search problem for a quadruped robot learning 3-dimensional locomotion. As the model of the robot is purposely not provided to the system, in here we solve this black-box problem by making a comparison between BO and CMA-ES. This paper shows both BO and CMA-ES methods are capable of finding the optimal parametric combinations that make $SmartAnt$ walk as far as possible in 10 s. This means that these two methods are data-efficient and friendly for expensive, black-box problem optimization. Especially, CMA-ES method is more sample-efficient than BO method for high dimensional parametric search in a continuous search space, and BO is more suitable for quick convergence with good prior knowledge in discrete ranges. Besides, from controller parameters' aspect, tuning oscillation angular amplitude could profit considerably than other parameters.

REFERENCES

[1] K. B. Ensor and P. W. Glynn, "Stochastic optimization via grid search," *Lectures in Applied Mathematics-American Mathematical Society*, vol. 33, pp. 89–100, 1997.

[2] J. Yosinski, J. Clune, D. Hidalgo, S. Nguyen, J. C. Zagal, and H. Lipson, "Evolving robot gaits in hardware: the hyperneat generative encoding vs. parameter optimization.," in *ECAL*, pp. 890–897, 2011.

[3] S. Chernova and M. Veloso, "An evolutionary approach to gait learning for four-legged robots," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3, pp. 2562–2567, IEEE, 2004.

[4] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, pp. 2951–2959, 2012.

[5] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth, "An experimental comparison of bayesian optimization for bipedal locomotion," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1951–1958, IEEE, 2014.

[6] J. Zhu, S. Li, Z. Wang, and A. Rosendo, "Bayesian optimization of a quadruped robot during 3-dimensional locomotion," in *Conference on Biomimetic and Biohybrid Systems*, pp. 295–306, Springer, 2019.

[7] N. Hansen, "The cma evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[8] K. A. Saar, A. Rosendo, and F. Llda, "Bayesian optimization of gaits on a bipedal slip model," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1812–1817, IEEE, 2017.

[9] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, pp. 63–71, Springer, 2003.

[10] A. Rosendo, M. von Atzigen, and F. Iida, "The trade-off between morphology and control in the co-optimized design of robots," *PloS one*, vol. 12, no. 10, p. e0186107, 2017.

[11] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[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*, vol. 86, no. 1, pp. 97–106, 1964.

[13] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural networks*, vol. 21, no. 4, pp. 642–653, 2008.

[14] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, no. 7553, p. 503, 2015.

[15] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Optimizing walking controllers for uncertain inputs and environments," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, p. 73, 2010.

[16] T. Geijtenbeek, M. Van De Panne, and A. F. Van Der Stappen, "Flexible muscle-based locomotion for bipedal creatures," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 206, 2013.

[17] I. Loshchilov and F. Hutter, "Cma-es for hyperparameter optimization of deep neural networks," *arXiv preprint arXiv:1604.07269*, 2016.